A publication of ISCA*:
International Society for Computers
and Their Applications

INTERNATIONAL JOURNAL OF COMPUTERS AND THEIR APPLICATIONS

TABLE OF CONTENTS

Page

*** "International Journal of Computers and Their Applications is abstracted and indexed in INSPECT and Scopus."**

Volume 22, No. 3, Sept. 2015 **ISSN 1076-5204**

International Journal of Computers and Their Applications

A publication of the International Society for Computers and Their Applications

ISCA Headquarters···•···64 White Oak Court, Winona, MN 55987···•···Phone: (507) 458-4517
E-mail: isca@ipass.net • URL: http://www.isca-hq.org

Guest Editor’s Note

CATA (Computers and their Applications) is the flagship conference for the International Society of Computers and their Applications. The intent of the conference has been to blend theory and practice as a means of stimulating researchers from both research dimensions. The papers for the special issue have been chosen to illustrate the spectrum of the 66 papers presented at the CATA 2015 conference. The authors were asked to change the title of their paper (to avoid confusion with their conference paper) and to extend their paper to make the papers journal appropriate. The CATA special issue contains five papers:

Paper 1: Rahul Sehgal and Hassan Peyravi. End to End Delay Analysis in Delay Tolerant Networks.

Paper 2: Hiroyuki Nishiyama, Akira Yoshizawa, Hirotoshi Iwasaki, and Fumio Mizoguchi. Design and Implementation of a New Tool for Detecting Distracted Car Driving Using Eye Movement and Driving Data on a Tablet PC.

Paper 3: Hiroyuki Chishiro and Nobuyuki Yamasaki. Zero-Jitter Technique for Semi-Fixed-Priority Scheduling with Harmonic Periodic Task Sets.

Paper 4: Keigo Mizotani, Yusuke Hatori, Yusuke Kumura, Masayoshi Takasu, Hiroyuki Chishiro, and Nobuyuki Yamasaki. An Integration of Imprecise Computation Model and Real-Time Voltage and Frequency Scaling on Responsive Multithreaded Processor.

Paper 5: Khaled Buragga, Elnomery Zanaty and Sultan Aljahdali. Improving Fuzzy C-Means Algorithm for Magnetic Resonance Images (MRIs) Segmentation.

Sehgal and Peyravi provide an analytical model of delay tolerant networks that builds in the topology of the underlying networks. Nishiyama et al. have investigated the use of a user interface capable of detecting driver distraction through the driver’s eye movements. Chishiro and Yamasaki propose a new semi-fixed-priority scheduling algorithm with the focus on achieving zero-jitter of tasks with harmonic periodic task sets. Mizotani et al. developed an approach capable of integrating an imprecise computation model with real-time voltage and frequency scaling to improve computation quality and power consumption. Buragga et al. look at improvements for MRI segmentation.

Gordon Lee and Les Miller

End to End Delay Analysis in Delay Tolerant Networks

Rahul Sehgal* and Hassan Peyravi*
Kent State University, Kent, Ohio 44240, USA

Abstract

Delay (Disruption, Disconnection) Tolerant Networks (DTNs) are sparse networks with intermittent connectivity. They play a growing important role in various communication systems including vehicular networks, mule networks, interplanetary networks, sensor networks and mobile ad hoc networks as well as in networks operating under extreme conditions. Delay Tolerant Networks (DTNs) are based on store-carry-forward techniques in order to compensate for intermittent link connectivity. The intermittent connectivity pattern (deterministic or probabilistic) is affected by the network nodes mobility. The End-to-End (E2E) delay performance is mainly affected by the dynamics of the underlying network topology, the routing algorithm, and the traffic load. In this paper, we develop an analytical model that incorporates the link intermittent distribution along with a tandem queuing model to determine the E2E delay performance under various traffic conditions and links disruption distributions.

Key Words: Delay/disruption tolerant networks, delay analysis, tandem queues, mean connection time, Markov chains, link intermittency.

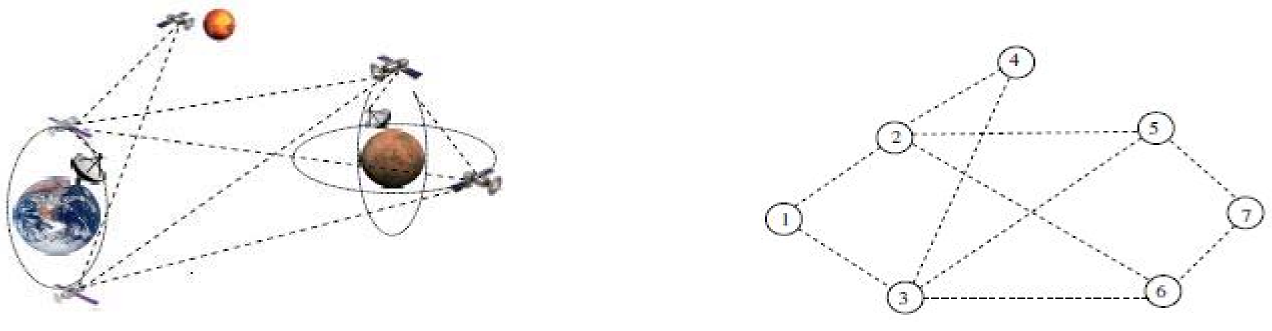
1 Introduction

Since the original work in delay-tolerant networking architecture [2], applications of Delay Tolerant Networks (DTNs) are widespread in various network architectures, including outer space communications such as space and interplanetary network systems, interconnection of mobile wireless devices, such as networks for intelligent highways and remote environmental and animal movement outpost, and mobile ad hoc networks such as battlefield networks interconnecting troops, aircraft, satellites, and sensors. These networks operate under various constraints such as host mobility, network mobility, intermittent connectivity due to network elements failure or transmission reachability, highly integrated low-power, low cost devices, and decentralized control. These constraints result in long and variable propagation delays with no guaranteed End-to-End (E2E) connectivity. The cause of intermittent connectivity could include periodic or predictable mobility such as spacecraft communications, stochastic mobility such as military or tactical networks, and exotic links such as deep space or acoustic links with low capacity and high error rates. Other reasons for link disruption that could make the network disconnected include natural disasters or malicious attacks on some part of the network infrastructure.

Traditionally, networks are modeled as connected graphs, where nodes represent hosts or hubs and edges represent bi-directional links. These networks generally work in a store-forward fashion, where data packets are stored and forwarded at each hop according to a routing mechanism. In this context, packets are not supposed to reside in a node's buffer for a long period of time. Based on this assumption, buffer sizes are relatively small and optimized in a way to keep both delay and throughput under control. The flow and congestion control in Transport Control Protocol (TCP)/Internet Protocol (IP) networks are designed to achieve these objectives. However, the assumption of all-time network connectivity has to be relaxed in DTNs and that would make analytical modeling more challenging. The underlying graph connectivity changes over time by mobility and that results in unusual and repetitive occurrences of network partitioning and topology changes. Thus, it is possible that two nodes that are connected at a given time t will not be connected at time $t + \Delta t$. These challenges and limitations resulted in a new network architecture initially defined by Internet Engineering Task Force (IETF) through RFC 4838 [2] and RFC 5050 [14], and later extended by Internet ResearchTask Force (IRTF) through RFC 6258 [17], and RFC 7242 [3].

DTN can be viewed as overlay network architecture that operates above the protocol stacks of the distinct Intermittently Connected Networks (ICNs). It enables gateway functionality between protocol stacks through the use of storage capacity. Several techniques, including replication and parallel forwarding, and forward error correction for overcoming communication impairment have been developed in the literature [7]. In DTN, a Bundle Protocol (BP) sits on the top of TCP layer and forms a store-and-forward overlay connectivity. Major operations of BP include (i) custody-based retransmission, (ii) coping with intermittent connectivity, (iii) taking advantage of scheduled, predicted, and opportunistic connectivity, and (iv) binding of overlay network endpoint identifiers with Internets addresses [14]. Figure 1 illustrates an interplanetary network with deterministic intermittent connectivity.

* Department of Computer Science. E-mail: (rsehgal, peyravi)@cs.kent.edu.



(a)

(b)

Figure 1: An interplanetary network (a) and its corresponding graph (b)

In this paper, we have developed an analytical model based on an open queuing system that can predict E2E delay in closed form under link intermittency for DTNs. The model incorporates the link disruption distribution, traffic load distribution, and the underlying routing matrix.

1.1 Related Work

In [11], an analytical framework, using a time-homogeneous discrete-time Markov chain, has been developed to predict the performance for utility-based algorithms with general heterogeneous mobility. The model combines mobility properties with actions determined by the DTN algorithm to compute the transition probabilities. This model maps an optimization problem into a Markov chain, where each state (e.g. assignment of content replicas to nodes) is a potential solution. The model incorporates both single copy and multi copy algorithms for unicast routing and content placement. The accuracy of the predictions performance has been measured against simulations over a range of synthetic and mobility traces. The model decouples DTN algorithm's effect from mobility, but allows one to derive performance metrics (convergence delay, delivery probability) using transient analysis of the Markov chain. The model, however, doesn't incorporate multicast routing algorithm and affect of traffic density on E2E delay in DTN under heterogeneous mobility.

The message propagation has been estimated in [12]. It gives a detailed expression of average information dissemination delay based on message size, social dynamics and number of nodes in the network. The model assumes both the contact time and inter-contact period are exponentially distributed. The model considers other social characteristics including distinct inter-contact periods, as various people might exhibit totally different behaviors owning to their working, living place and friends circle. It divides applications into different groups according to their communication intervals with each other. Sometimes people would be unwilling to forward a message to others due to energy and storage constraints or prefer to forward information to people in the same group with them. The two kinds of activities are called individual selfishness and social selfishness, respectively. The model concentrates on social selfishness which has a considerable effect on communication between groups. This model shows that as the number of nodes increases, delay decreases. This approach doesn't relate delay to traffic density or routing scheme.

In [19], the Localized Random Walk (LRAW) is used to model DTNs. In LRAW, each node is assigned a fixed Cell and makes a list of all of its neighbors at each time slot τ. A node moves from its current location with a fixed probability s, and selects one of the neighboring cells with probability p. The model offers an approximation for average message delay. However, the model does not consider the effect of message arrival rate or the traffic load.

The analytical models described in [21] and [20] are based on *Epidemic Routing* [18] algorithm. In [21], the delivery probability is estimated by the bundle size. The model is based on flooding on an edge-Markovian evolving graph in which at each time slot an edge changes its state according to a two-state Markovian process with probabilities p (edge birth-rate) and q (edge death-rate). The model captures the relationship between successive connectivity graphs. It computes the delivery ratio as a function of bundle size, the maximum tolerated delay, and the dynamics of the underlying evolving graph. It assumes a finite number of nodes with a finite link capacity and a finite message size. On the other hand, a random graph approach to gain insight into the temporal nature of the epidemic routing has been used in [20]. It models the spread of a packet in mobile wireless networks. The process is iterative by which the probability of a number of infected nodes is a function of time.

In [1], the authors compare the stochastic properties of different mobility models in Mobile Ad Hoc Networks (MANETs) based on the number of states visited in a fixed time, the time to visit every state in a region, the first passage time, the full coverage time, and the transient behavior. They derive an analytical model to get the transient probability distributions and First Passage Times for 1-dimensional correlated random walks. Then they use experiments to study these stochastic properties for different mobility models.

The contribution of this paper is twofold. First, we develop a traffic distribution model for DTN that incorporates dissemination of packets traversing the network under a routing matrix obtained by a mobility model. For that, we use a time homogeneous discrete time Markov chain. Second, we employ a strong, yet simple, open queuing system based on Jackson's theorem [4] that facilitates the computation of E2E delay performance under various traffic load and routing conditions.

The rest of the paper is organized as follows. The open tandem queuing model that describes packet migration along a path towards a destination along with the effect of aggregate traffic on each hop is described in Section 2. Supporting mobility and closed form solutions are discussed in Section 3. Section 4 incorporates the expected connection time in closed form solutions. Section 5 describes numerical results along with simulation of some network scenarios. Conclusions and remarks are given in Section 6.

2 The Model

The model is based on an interconnection of tandem open queuing systems in which packets move from one queue to the next queue under a routing matrix that is dynamically generated by a routing algorithm due to mobility or disruption. The routing matrix itself represents the state of the network at a given time. Now, consider a partial structure of the network model in Figure 2, where γ_j is the cumulative rate of exogenous (external) traffic flows entering node j, λ_j is the aggregate traffic, both exogenous and indigenous (relay) flows, arriving to node j. r_{ij} is the probability of a packet routed from node i to node j under the routing algorithm.

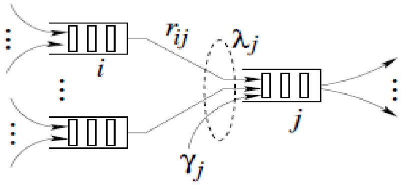


Figure 2: The effect of exogenous and indigenous that form aggregate traffic on node j

The network forms an open tandem queuing model in which each node has a First In First Out (FIFO) buffer. After a packet is transmitted by a node, it may move to another node, or leave the system completely.

In tandem queuing systems, the arrival times to the receiving node are strongly correlated with departure time from the preceding nodes. There exists no analytical results for such networks in which inter-arrival and service times are dependent. However, Kleinrock independence approximation [8] states that merging several packet streams on a transmission line has an effect akin to restoring the independence of inter-arrival times and packet lengths, thus an *M/M/*1 queuing model can be used to analyze the behavior of each communication link.

A Jackson queuing network [4] is a network of an *n M/M/n* state-independent queuing system with the following features. (i) There is only one class of packets arriving to the system. (ii) Exogenous packets arrive at node j according to a Poisson process with rate $\gamma_j \geq 0$. (iii) The service times of the packets at jth queue are exponentially distributed with mean $1/\mu_j$. Upon receiving its service at node i, the packet will proceed to node j with a probability r_{ij} or leave the network at node i with probability $(1-\sum_{j=1}^{n} r_{ij})$. Finally, the queue capacity at each node is assumed to be infinite, so there is no packet dropping.

Let **R** be the $n \times n$ probability matrix describing the routing of packets within a Jackson network [4], $\lambda = (\lambda_1,\lambda_2,\cdots,\lambda_n)$ be the mean arrival rates of the relayed packets, and $\gamma = (\gamma_1,\gamma_2,\cdots,\gamma_n)$ be the mean arrival rates of the exogenous packets. Unlike the state transition used for Markov chains, the rows of **R** matrix need not necessarily sum up to one, i.e., $\sum_j r_{ij} \leq 1$. The routing matrix **R** is simply generated by the underlying DTN routing algorithm during a stationary period. Assuming the network reaches equilibrium, then we can write the following traffic equation using the flow conservation principle, in which the total sum of arrival rates entering the system is equal to the total departure rate under steady-state condition.

$$\lambda_j = \gamma_j + \sum_{i}^{n} \lambda_i r_{ij}, \qquad j = 1, 2, ..., n. \tag{1}$$

In the steady state, assuming the network is stable, the aggregate input rate λ_j into node j is equal to the aggregate output rate from node i, $i = 1, \cdots$, n. Therefore, we have a system of n equations and n unknowns. These equations can be written in matrix form as,

$$\vec{\lambda} = \vec{\gamma} + \vec{\lambda}\,\mathbf{R} \tag{2}$$

and the aggregate arrival rate vector can be solved by,

$$\vec{\lambda} = \vec{\gamma}\ \ (\mathbf{I}-\mathbf{R})^{-1} < \vec{\mu} \tag{3}$$

where, $\gamma = (\gamma_1,\gamma_2,\cdots,\gamma_n)$ and the components of the vector **I** give the arrival rates into the various stations, and $\mu = (\mu_1,\mu_2,\cdots,\mu_n)$ is a vector representing service rates. The service times are assumed to be mutually independent and also independent of the arrival process at that queue, regardless of the previous service times of the same packet in other nodes.

The mean queue size and mean delay for the jth queue are given by,

$$E[L_j] = \frac{\lambda_j}{\mu_j - \lambda_j} = \frac{\rho_j}{1-\rho_j}, \quad E[D_j] = \frac{1}{\mu_j - \lambda_j}, \quad j = 1, 2, ..., n \tag{4}$$

where, we can compute the arrival rate λ_j and the expected queuing delay $E[D_j]$ from Equations (3) and (4), respectively.

Consider a path $x = x_l \rightarrow x_{l-1} \rightarrow \cdots \rightarrow x_1$ from source node x_l to a destination node x_1, then the end-to-end delay for traffic originated at node x_l is:

$$E[D_x] = \sum_{j=1}^{l} \frac{1}{\mu_{x_j} - \lambda_{x_j}} \tag{5}$$

Let χ be the set of paths generated by routing algorithm R, then the average E2E delay $\overline{D}$ can be computed by,

$$\overline{D} = \frac{1}{|\chi|} \sum_{x \in \chi} E[D_x] \tag{6}$$

Consider the network in Figure 1 in which some links may not always be connected. Without loss of generality, assume that during a particular time interval Δt all traffic generated by nodes 2-7 are heading towards node 1, then Equation (3) gives the following solutions.

$$\vec{\lambda} = \begin{bmatrix} \gamma_1 + \gamma_2 + \gamma_3 + \gamma_4 + \gamma_5 + \gamma_6 + \gamma_7 \\ \gamma_2 + 1/2\,\gamma_4 + 1/2\,\gamma_5 + 1/2\,\gamma_6 + 1/2\,\gamma_7 \\ \gamma_3 + 1/2\,\gamma_4 + 1/2\,\gamma_5 + 1/2\,\gamma_6 + 1/2\,\gamma_7 \\ \gamma_4 \\ \gamma_5 + 1/2\,\gamma_7 \\ \gamma_6 + 1/2\,\gamma_7 \\ \gamma_7 \end{bmatrix} \tag{7}$$

If we assume the exogenous generated locally by each node has a rate γ, then the E2E delay for each traffic source can be computed by Equation (6) as,

$$\vec{E}(D_i) = 1 \big/ (\mu_i - \gamma_i), \qquad 1 \leq i \leq n. \tag{8}$$

Note that γ_i is bounded by the condition in Equation (3) and must hold for each node (i.e., $\lambda_i < \mu_i$).

3 Supporting Mobility

Mobility is an important aspect of MANETs and many scenarios in DTNs and MANET. Mobility models can be classified into (i) *deterministic models* (e.g., urban traffic models), (ii) *semi-deterministic models* (e.g., Column model, Pursue models), and (iii) *random models* (e.g., Brownian Motion and Waypoint models) [10]. Mobility can change the state of the routing matrix **R** deterministically or probabilistically at a given time as defined by one of the above mobility models. **R** contains two pieces of information; adjacency (connectivity) matrix that represents that state of the network, and weighted links that represent the proportion of traffic routed through each link. This is particularly useful when alternate paths are used within a DTN. Let $N = \{v_1, v2, \cdots, v_n\}$ be the set of nodes and $L = \{l_{ij};\ 1 \leq i,j \leq n,\ i < j\}$ the set of links. We define the adjacency matric $A = [a_{ij}]$ from the routing matrix $\mathbf{R} = [r_{ij}]$ as,

$$a_{ij} = \begin{cases} 1, & r_{ij} \qquad 1 \leq i,\ j \leq n \\ 0, & otherwise \end{cases} \tag{9}$$

We assume undirected graphs that represent symmetric connectivity in terms of bit-rate between two connected nodes. Therefore, $|L| = n(n-1)/2$, represents the size of transition states. Hence, there are potentially $2^{|L|}$ states the network can go through. Let $S = \{0,1,2,\cdots,2^{|L|}-1\}$ be the set of possible states the network can go through as the result of any mobility pattern, where each state can be represented by a binary number $[s_0 s_1 \cdots s_{|L|-1}]$, $s_i \in \{0,1\}$, $s_i = 1$ corresponds to link i being in ON state, and $s_i = 0$ corresponds to link i being in OFF state, $0 \leq i \leq |L|-1$. Therefore, $[00\cdots0]$ represents an edge-less graph and $[11\cdots1]$ represents a fully-connected graph. The transition from OFF state to ON state on link i occurs with probability β_i and the transition from ON state to OFF state on link i occurs with probability α_i. In DTN terminology, ON and OFF periods are called contact time and inter-contact time, respectively. In discrete time, OFF and ON periods are geometrically distributed with the mean $1/\alpha_i$ and $1/\beta_i$, respectively. In continuous time, OFF and ON periods are exponentially distributed with the mean $1/\alpha_i$ and $1/\beta_i$, respectively. Figure 3 represents the transition probabilities.

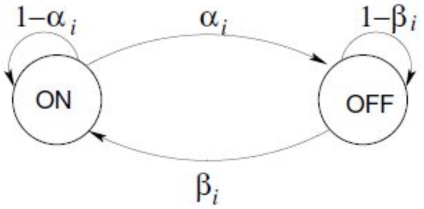
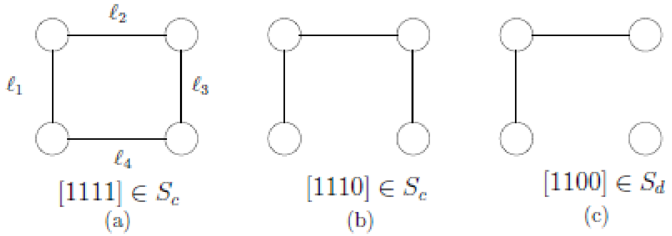


Figure 3: State transition of link i

The transition probability matrix and the transition rate matrix for the state transition in Figure 3 are p_i and q_i, respectively.

$$p_i = \begin{bmatrix} 1-\alpha_i & \alpha_i \\ \beta_i & 1-\beta_i \end{bmatrix}, \qquad q_i = \begin{bmatrix} -\dfrac{1}{\alpha_i} & \dfrac{1}{\alpha_i} \\ \dfrac{1}{\beta_i} & -\dfrac{1}{\beta_i} \end{bmatrix} \tag{10}$$

Now, Consider the network in Figure 4 with 16 possible states of which under mobility 5 states preserve network connectivity and 11 states cause network disconnectivity. The set of possible states are divided into two groups; connected states S_c, in which the underlying graph is connected, and disconnected states, S_d, in which the underlying graph is not connected. The latter cause intermittency in DTNs. When a link disruption does not result in network disconnectivity, such as those in states $S_c = \{7, 11, 13, 14, 15\}$, then the average E2E delay can be computed by Equation (6). However, for the rest of the states (S_d), the E2E delay should be computed differently. This is discussed in Section 4. Figure 5 illustrates the state transitions for the network in Figure 4(a). Yellow nodes correspond to disconnected states and blue nodes correspond to connected states. For the sake of simplicity and without loss of generality, we assume $\alpha_i = \alpha$ and $\beta_i = \beta$ for all links. Let π_k be the probability that the network is in state k, $0 \le k < 2^{|L|} - 1$. Where, $\sum_{k=0}^{2^{|L|-1}} \pi_k = 1$. The state diagram in Figure 5 gives the following balance equations. Let P be the transition probability matrix corresponding to the state transition diagram in Figure 5, where $P_{ij} \in \{0, \alpha, \beta\}$, $0 \le i, j \le 15$. The balance equations in (11) can be re-written in a generic compact form as,



$[1111] \in S_c$ (a) $[1110] \in S_c$ (b) $[1100] \in S_d$ (c)

Figure 4: (a) and (b) are connected states and (c) is a disconnected state.

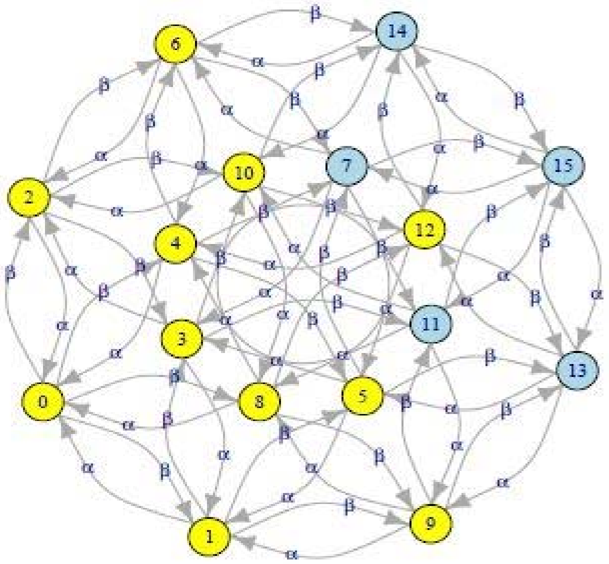


Figure 5: State transitions diagram

$$
\begin{cases}
(4\beta)\pi_0 &= \alpha\pi_1 + \alpha\pi_2 + \alpha\pi_4 + \alpha\pi_8 \\
(3\beta+\alpha)\pi_1 &= \beta\pi_0 + \alpha\pi_3 + \alpha\pi_5 + \alpha\pi_9 \\
(3\beta+\alpha)\pi_2 &= \beta\pi_0 + \alpha\pi_3 + \alpha\pi_6 + \alpha\pi_{10} \\
(2\beta+2\alpha)\pi_3 &= \beta\pi_1 + \beta\pi_2 + \alpha\pi_7 + \alpha\pi_{11} \\
(3\beta+\alpha)\pi_4 &= \beta\pi_0 + \alpha\pi_5 + \alpha\pi_6 + \alpha\pi_{12} \\
(2\beta+2\alpha)\pi_5 &= \beta\pi_1 + \beta\pi_4 + \alpha\pi_7 + \alpha\pi_{13} \\
(2\beta+2\alpha)\pi_6 &= \beta\pi_2 + \beta\pi_4 + \alpha\pi_7 + \alpha\pi_{14} \\
(\beta+3\alpha)\pi_7 &= \beta\pi_3 + \beta\pi_5 + \beta\pi_6 + \alpha\pi_{15} \\
(3\beta+\alpha)\pi_8 &= \beta\pi_0 + \alpha\pi_9 + \alpha\pi_{10} + \alpha\pi_{12} \\
(2\beta+2\alpha)\pi_9 &= \beta\pi_1 + \beta\pi_8 + \alpha\pi_{11} + \alpha\pi_{13} \\
(2\beta+2\alpha)\pi_{10} &= \beta\pi_2 + \beta\pi_8 + \alpha\pi_{11} + \alpha\pi_{14} \\
(\beta+3\alpha)\pi_{11} &= \beta\pi_3 + \beta\pi_9 + \beta\pi_{10} + \alpha\pi_{15} \\
(2\beta+2\alpha)\pi_{12} &= \beta\pi_4 + \beta\pi_8 + \alpha\pi_{13} + \alpha\pi_{14} \\
(\beta+3\alpha)\pi_{13} &= \beta\pi_5 + \beta\pi_9 + \beta\pi_{12} + \alpha\pi_{15} \\
(\beta+3\alpha)\pi_{14} &= \beta\pi_6 + \beta\pi_{10} + \beta\pi_{12} + \alpha\pi_{15} \\
(4\alpha)\pi_{15} &= \beta\pi_7 + \beta\pi_{11} + \beta\pi_{13} + \beta\pi_{14}
\end{cases}
\tag{11}
$$

$$\pi_i \sum_{j=0}^{2^{|L|-1}} P_{ij} - \sum_{j=0}^{2^{|L|-1}} \pi_j P_{ji} = 0, \quad 0 \le i \le 2^{|L|} - 1 \tag{12}$$

In steady state, the set of equations in (11) can be solved symbolically by a linear solver, and the steady state probabilities can be computed as,

$$
\begin{array}{lll}
\pi_0 = \beta^4/d & \pi_1 = \beta^3\alpha/d & \pi_2 = \beta^3\alpha/d \\
\pi_3 = \beta^2\alpha^2/d & \pi_4 = \beta^3\alpha/d & \pi_5 = \beta^2\alpha^2/d \\
\pi_6 = \beta^2\alpha^2/d & \pi_7 = \beta\alpha^3/d & \pi_8 = \beta^3\alpha/d \\
\pi_9 = \beta^2\alpha^2/d & \pi_{10} = \beta^2\alpha^2/d & \pi_{11} = \beta\alpha^3/d \\
\pi_{12} = \beta^2\alpha^2/d & \pi_{13} = \beta\alpha^3/d & \pi_{14} = \beta\alpha^3/d \\
\pi_{15} = \alpha^4/d & &
\end{array}
\tag{13}
$$

where, $d = (\beta + \alpha)^4$.

4 Expected Connection Time

While the expected delay for connected states ($S_c = \{7, 11, 13,$

14,15}) can be directly computed from Equation (6), the expected delay for disconnected states ($S_d = \{0,1,2,3,4,5,6,8,9,10,12\}$) include the expected connection time to reach a connected state and that is what we calculate in this section.

We refer to Markov chain $X_0, X_1, X_2, \cdots$ which has the state space S, and that could be finite or infinite set. Let, as before, P be the transition matrix representing the state transition diagram in Figure 5. The first connection time F_{ij}, and the mean first connection time m_{ij} from state i to state j are defined [9] as follows:

$$F_{ij} = \min\{t \ge 1 : X_t = j - | X_0 = i\}\ i, j \in S$$

$$m_{ij} = E(F_{ij}) = \sum_{k=1}^{\infty} k \Pr[F_{ij} = k] \qquad (14)$$

$M = [m_{ij}]$ is called the mean first connection time matrix of the chain. The mean first connection time gives us information about the short range behavior of the chain, i.e., how long one can expect to get to a state of connectivity, given that the current state is a disconnected state.

The mean connection time, $\overline{m}_i$, for the state transition diagram in Figure 5 has been calculated accordingly in Table 1, where, $i \in S_d = \{0,1,2,3,4,5,6,8,9,10,12\}$ and $j \in S_c = \{7,11,13,14,15\}$. Note that $\overline{m}_i$ is averaged over column j, and that represents the average connection time from state i to any state j.

Table 1: Mean connection time (m_i)

	$S_c\{7,11,13,14,15\}$			
	$\alpha = 0.5$	$\alpha = 0.7$	$\alpha = 0.6$	$\alpha = 0.8$
S_d	$\beta = 0.5$	$\beta = 0.4$	$\beta = 0.3$	$\beta = 0.7$
Mean	2.26	3.95	5.83	1.71

Now, the average E2E delay can be computed by,

$$\overline{\overline{D}} = \sum_{i \in S_c} \pi_i \overline{D}_i + \sum_{i \in S_d} \pi_i (\overline{D}_i + \overline{m}_i) \qquad (15)$$

where, $\overline{D}_i$ is obtained from Equation (6) and $\overline{m}_i$ from Equation (14). In order to tackle the composite delay we decouple packet transmission delay from disruption delay as shown in Equation (15).

5 Simulation Results

We have tested the model for various network topologies using both the analytical model described in this paper as well simulation models. Specifically, we used DTN simulator *ONE* [6] to compare theoretical results with simulation results. In these experiments, we have maintained an unsaturated state for each node (queue) in the system and maintained the stability condition for each node under composite traffic aggregation by conforming with Equation (3). We run simulations long enough to meet the steady state requirement of the analytical model. We used *First Contact* routing algorithm [5] which forwards a packet along an edge chosen randomly among all the current contacts. Anti-cycling and packet duplication elimination have been provisioned. The routing protocol is a single copy protocol in which at any given time t, there exists only one copy of the message in the network. We set up the simulation parameters with channel-speed (250 KB/s), packet size (250 KB), service time (μ = 1 s), traffic load ($0.1 \le \rho \le 0.9$), and ON/OFF probabilities ($\alpha = \beta = 0.5$).

The results obtained from the analytical model in Sections 3 and 4 have been compared with the simulation results obtained under various traffic load and disruption conditions. The E2E delay was calculated under two scenarios; for connected states (S_c) in which E2E connectivity is always available, and for all states ($S_c \cup S_d$) to observe the impact of link disruption on E2E delay. These are illustrated in Figure 6a and 6b, respectively. While we used a 4-node network to illustrate the delay analysis, we have also tested the model for more realistic topologies. In particular, we used the two network topologies used by NASA for space communications. These are shown in Figure 7. The analytical and simulation results are shown in Figure 8. The reason the analytical model gives a slightly better performance than the simulation experiments in Figures 6 and 8 is the result of finer timing used in analytical model ($\Delta t \to 0$) and exponentially generated pseudorandom numbers generated by the simulation. The gap becomes slightly larger when the number of nodes or the network load is increased, and the analytical results show better performance than simulation. Still, E2E delay approximation is very close to simulation results.

6 Conclusions

In this paper, we have developed a new model that can be used to approximate the average E2E delay in DTNs. The model is based on an open queuing system with Markovian process using Jackson's theorem [4]. We have augmented the model to incorporate various types of mobility models. The model efficiently calculates the transition time between various graph disconnectivity states and those which are in the states of connected graphs. The model is quite general in the sense that it can accommodate various routing algorithms as long as the algorithm that generates a routing matrix is irreducible on each time interval. The limitation of the model relates to the scenarios when mobility pattern is highly random and causes infinite disconnectivity. This model can be further extended to explore various routing algorithms such as Spray and Wait routing [16] in which multiple copies of the same message are transmitted via connected links through a controlled flooding mechanism to increase the probability of destination reachability while increasing the queuing delay.

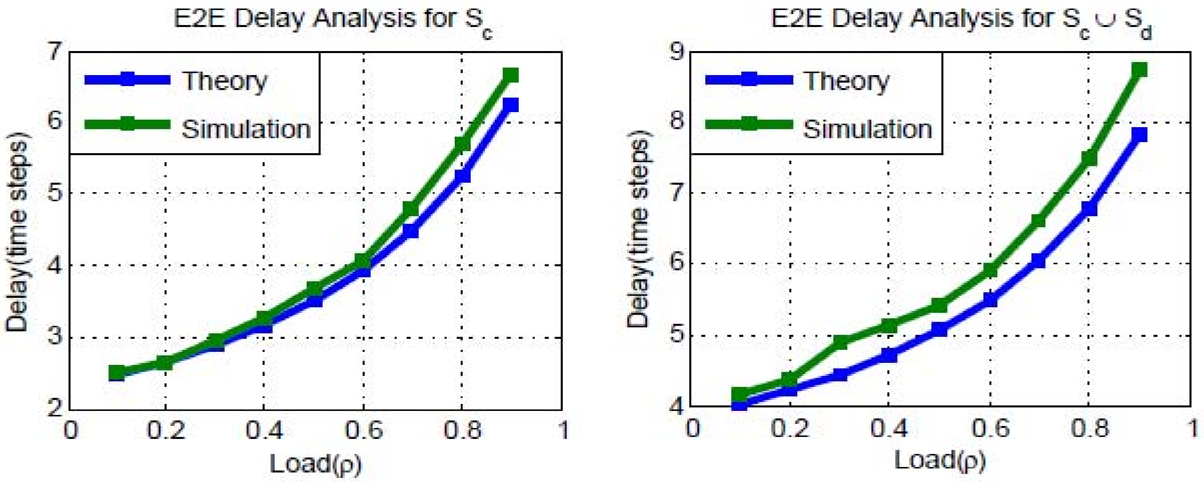
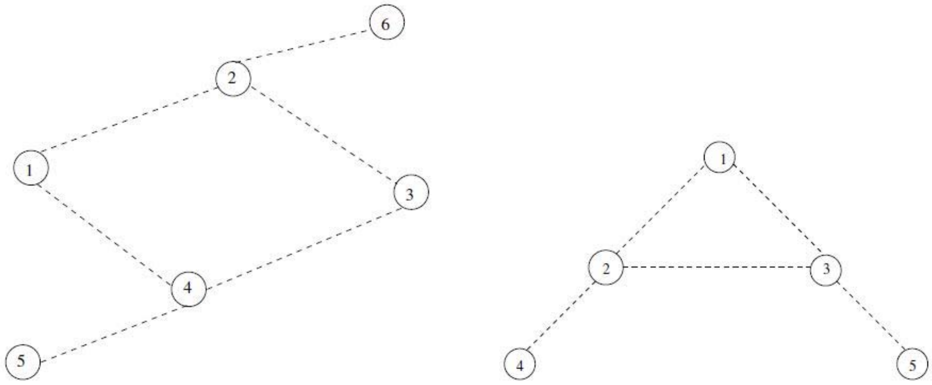


Figure 6: E2E delay analysis of 4-node network



(a) Multiple Spacecraft as in [13]

(b) Spacecraft Topology similar to [15]

Figure 7: NASA Spacecraft Topologies

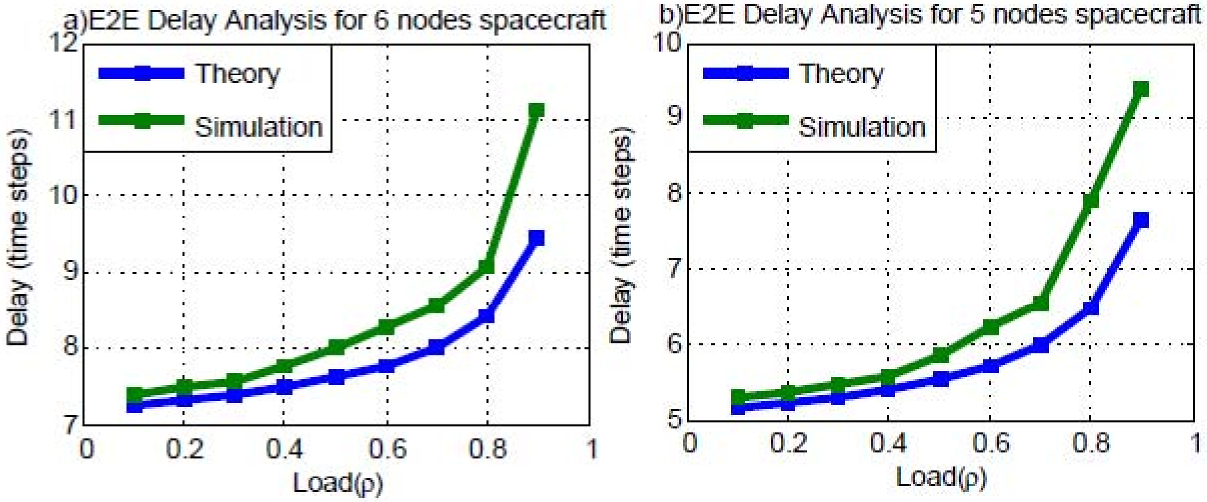


Figure 8: E2E delay analysis of NASA spacecraft topologies

References

[1] S. Bandyopadhyay, E. J. Coyle, and T. Falck, “Stochastic Properties of Mobility Models in Mobile Ad Hoc Networks,” *IEEE Transactions on Mobile Computing*, 6(11):1218-1229, 2007.

[2] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, *Delay-Tolerant Networking Architecture*, Network Working Group, RFC:4838, April 2007.

[3] M. Demmer, J. Ott, and S. Perreault, *Delay-Tolerant Networking TCP Convergence-Layer Protocol*, Network Working Group, RFC:7242, June 2014.

[4] J. R. Jackson, “Networks of Waiting Lines,” *Operations Research*, 5(4):518-521, 1957.

[5] S. Jain, K. Fall, and R. Patra, “Routing in a Delay Tolerant Network,” ser. SIGCOMM ’04. New York, NY, USA, ACM, pp. 145-158, 2004.

[6] A. Keränen, J. Ott, and T. Kärkkäinen, “The ONE Simulator for DTN Protocol Evaluation,” *SimuTools 2009*, Rome, Italy, ICST, p. 55, March 2-6, 200*9*.

[7] M. Khabbaz, C. M. Assi, and W. Fawaz, “Disruption-Tolerant Networking: A Comprehensive Survey on Recent Developments and Persisting Challenges,” *IEEE Communications Surveys and Tutorials*, 14(2):607-640, 2012.

[8] L. Kleinrock, *“Queueing Systems”*. Wiley Interscience, Vol. I: Theory, 1975.

[9] G. F. Lawler, “Introduction to Stochastic Processes, Second Edition (Chapman & Hall/CRC Probability Series)”. Chapman and Hall/CRC, 2006.

[10] M. Natkaniec, “Ad Hoc Mobile Wireless Networks: Principles, Protocols, and Applications,” *IEEE Communications Magazine*, 47(5):12-14, May 2009.

[11] A. Picu and T. Spyropoulos, “Forecasting DTN Performance Under Heterogeneous Mobility: The Case of Limited Replication,” *SECON*. IEEE, 9th Annual IEEE Communications Society Conference, pp. 569-577, 2012.

[12] L. Qiu, Y. Li, P. Hui, D. Jin, L. Su, and L. Zeng, “Edge-Markovian Dynamic Graph Based Performance Evaluation for Delay Tolerant Networks,” *IEEE WCNC*, pp. 2129-2133, April 2012.

[13] A. Rogers, K. Anderson, A. Mracek, R. Zenick, J. Grau, and C. Gramling, “An Integrated Vision-Based System for Spacecraft Attitude and Topology Determination for Formation Flight Missions,” 18th Annual AIAA/USU Conference on Small Satellites, 2004.

[14] K. Scott and S. Burleigh, *Bundle Protocol Specification*, Network Working Group, RFC:5050, Nov. 2007.

[15] R. Smith and F. Hadaegh, “Control Topologies for Deep Space Formation Flying Spacecraft,”,. *Proceedings of the 2002 American Control Conference,* 4:2836-2841, 2002.

[16] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, “Spray and Wait: an Efficient Routing Scheme for Intermittently Connected Mobile Networks,” *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, ser. WDTN ’05. New York, NY, USA, ACM, pp. 252-259, 2005.

[17] S. Symington, *Delay-Tolerant Networking Metadata Extension Block*, Network Working Group, RFC:6258, May 2011.

[18] A. Vahdat and D. Becker, *Epidemic Routing for Partially-Connected Ad Hoc Networks*, Department of Computer Science, Duke University, Tech. Rep., 2000.

[19] B. Walker, T. Clancy, and J. Glenn, “Using Localized Random Walks to Model Delay-Tolerant Networks,” *IEEE MILCOM 2008*, pp. 1-7, Nov 2008.

[20] Q. Wang and Z. J. Haas, “Analytical Model of Epidemic Routing for Delay-Tolerant Networks,” ser. HP-MOSys ’12, New York, NY, USA, ACM, pp. 1-8, 2012.

[21] J. Whitbeck, V. Conan, and M. Dias de Amorim, “Performance of Opportunistic Epidemic Routing on Edge-Markovian Dynamic Graphs,” *IEEE Transactions on Communications*, 59(5):1259-1263, May 2011.

Rahul Sehgal is currently a PhD candidate in Department of Computer Science at Kent State University, Ohio USA. He received his MS in Computer Science from Kent State University and B.Tech from Bharati Vidyapeeth, Pune, India. His current research area focuses on system modeling and performance engineering of challenged networks (Delay Tolerant Networks and Mobile Ad-Hoc networks (MANETs)) with applications. He also has valuable industrial experience as a Software Engineer.

Hassan Peyravi is Professor of Computer Science at Kent State University. He received his M.S. and Ph.D. degrees in Computer Science from School of EECS at the University of Oklahoma. He was a Member of Technical Staff with the Chief Architecture Group at AT&T Bell Laboratories in New Jersey. His research focuses on theoretical foundation and implementation aspects of various communication systems. His research has been supported by NASA, NSF, CAIDA, and Cisco.

Design and Implementation of a New Tool for Detecting Distracted Car Driving Using Eye Movement and Driving Data on a Tablet PC

Hiroyuki Nishiyama*[†]
Tokyo University of Science, CHIBA, 278-8510, JAPAN

Akira Yoshizawa[‡], Hirotoshi Iwasaki[‡]
Denso IT Laboratory, Shibuya-ku Tokyo, 150-0002 JAPAN

Fumio Mizoguchi*[†]
Tokyo University of Science, CHIBA, 278-8510, JAPAN
WisdomTex Co. Ltd., Tokyo 153-0063, JAPAN

Abstract

In this study, we design a new graphic user interface (GUI) to detect distracted car driving using the driver's eye movement and driving data. Our system uses an eye movement sensor to collect data about the driver's eye movements and a driving data sensor to gather data on speed, acceleration, and steering, as well as GPS data for location information. Our system measures the driver's eye movements and driving conditions based on information obtained from each sensor, and detects distracted car driving using the driver's mental model. We implement our system on a Windows PC and a tablet PC to detect distracted driving. We expect that this GUI will reduce car driving risks by providing advice or urging caution by voice utterance from the tablet PC when our system detects distracted driving.

Key Words: Detecting distracted car driving, eye movement, driving data, tablet PC, cognitive distraction

1 Introduction

Near the end of the 20th century, car driving safety was enhanced with improvements in the transportation infrastructure and in the laws and regulations in Japan, Europe, and the United States. In addition, various service systems (e.g., car navigation systems) for car drivers have been developed and sold. However, such devices as televisions and DVD players that are viewable in the car have also become popular. Such products can cause distracted driving and have caused traffic accidents in recent years. In addition, mobile phone and smart phone penetration is now resulting in distracted driving due to calls or operation of these phones while driving, causing traffic accidents and presenting new problems.

The National Highway Traffic Safety Administration (NHTSA) has identified three types of distracted driving based on distraction factors [7]: (1) visual distraction, (2) cognitive distraction, and (3) manual distraction. Visual distraction occurs when the driver is viewing an unrelated object (i.e., look-away driving). Viewing and operating a smart phone, viewing the car's TV, or operating and viewing the car navigation system while driving are considered visual distractions. In addition, the visibility of outside material (beyond safety checks) during driving also comprises a visual distraction. The second type, cognitive distraction, is due to the internal state of the driver who is thinking about unrelated things while driving. Examples include driving while talking on a cell phone and concentrating on one's thoughts. The third type, manual distraction, is caused by an intentionally careless driver. Among these distractions, we may consider the similarity between visual distraction and the concept of in attentional blindness described by Arien Mack [4]: "when you are looking at something, you fail to perceive a suddenly appearing stimulus." In order to detect distracted driving, we measure the driver's eye movements while driving. However, cognitive distraction is the result of the driver's internal state. It is difficult to detect cognitive distraction using just eye movement and driving data [8]. To solve this problem, we focus on Harbluk's finding that saccade frequency is reduced when driving with cognitive distraction [3]. Thus, it is possible to detect decreased attention to the driving environment by measuring saccade frequency. We have demonstrated that it is possible to detect cognitive distraction by applying eye movement and driving data to the driver's mental model [8] and measuring saccade frequency [6].

In the present study, we design a system to detect distracted car driving using the driver's eye movement and driving data. Our system uses an eye movement measurement sensor to collect data on the driver's eye movements, and we use a driving data measurement sensor to gather data on speed, acceleration, and steering, as well as GPS data for location. The system measures the driver's eye movements and driving situation based on information obtained from each sensor and detects distracted car driving using the driver's mental model

[*] Faculty of Sci. and Tech., Yamazaki 2641, Noda-shi, Email: hiroyuki@rs.noda.tus.ac.jp.
[‡] Shibuya CROSS TOWER, 28th Floor, 2-15-1 Shibuya.
[†] 1-17-3 Meguro-ku Meguro. Email: mizo@wisdomtex.com.

[8]. Our study implements a system on a tablet PC to detect distracted driving. We expect that this system will reduce the car driving risk by providing advice or urging caution by voice from the tablet PC when it detects distracted driving.

2 Eye Movement Driving Data

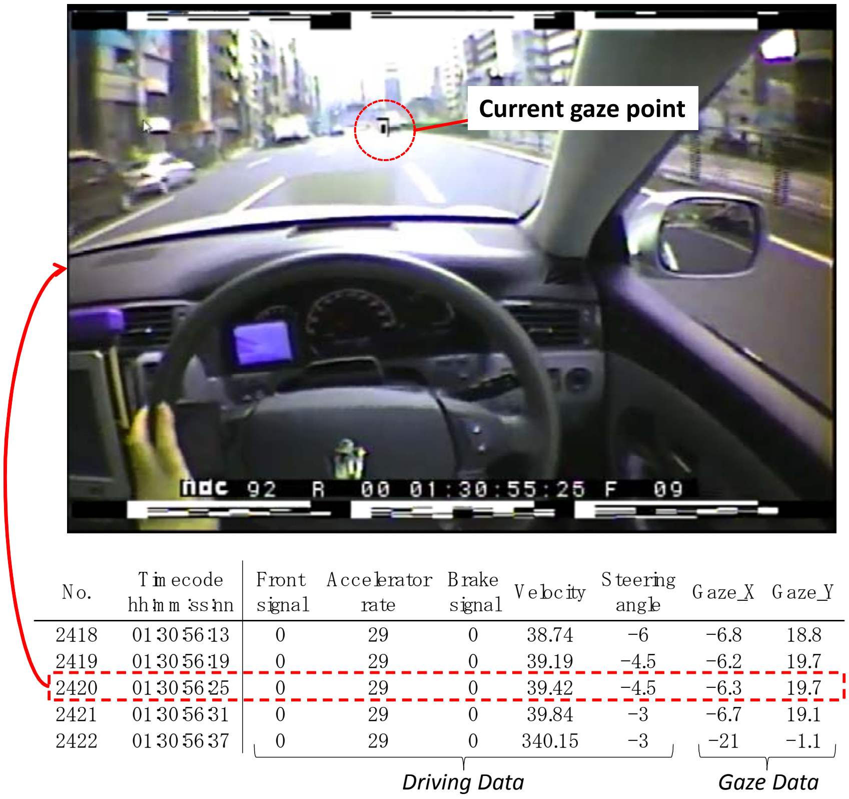
2.1 Raw Data

We use the EMR-8 (NAC Image Technology, Inc. (www.eyemark.jp)) to collect data on the driver's eye movements. This device measures horizontal and vertical viewing angles in degrees. For this study, we consider 60 data points per second [5-6, 10].

The Controller Area Network (CAN), an in-vehicle LAN used to gather driving data, is a standard used for data transfer between in-vehicle equipment. The International Organization for Standardization has standardized it as ISO 11898 and ISO 11519. We use CAN to obtain the accelerator depression rate (0% to 100%), braking signal (0 or 1), steering signal (-450 to 450 degrees), a signal representing the gear (0 to 4), and the front separation (in meters). For these measurements, we modified a Toyota Crown and obtained 10 data points per second.

We measured eye movement and driving data of a novice driver for 20 min. while driving on two types of road: an urban road with much traffic (Shibuya area) and a road with moderate traffic (Noda area). It should be noted that this subject usually stays in the Shibuya area; this driving experience in the Noda area was the first time. In this experiment, the results of driving for 1767 seconds provided data for 100,464 measurements in the Shibuya area, and the results of driving for 1266 seconds provided data for 72,905 measurements in the Noda area. First in our study, we generate a learning rule that enables us to detect visual distractions from the eye movement and driving data of this novice driver. We extract the object of attention in the timing of saccade from the video record made during driving, and judge whether attention paid to the subject is relevant to visual distraction. We execute SVM learning using saccade data judged as visual distractions as positive examples. It should be noted that we also have driving data for a senior driver; however, because the saccade data judging the visual distraction of the senior driver was less extreme, we use only the novice driver's data for learning.

The eye movement data and the car driving data must be synchronized (Figure 1). The third-row data (No. 2420) indicates that the car's velocity is 39.42km/h with an accelerator depression rate of 29%, and the vehicle is running almost straight (steering angle -4.5). The driver is gazing at the center (coordinates (-6.3, 19.7)). All the data can be used to produce training data, as demonstrated in the next section.



No.	Timecode hh:mm:ss:nn	Front signal	Accelerator rate	Brake signal	Velocity	Steering angle	Gaze_X	Gaze_Y
2418	01:30:56:13	0	29	0	38.74	−6	−6.8	18.8
2419	01:30:56:19	0	29	0	39.19	−4.5	−6.2	19.7
2420	01:30:56:25	0	29	0	39.42	−4.5	−6.3	19.7
2421	01:30:56:31	0	29	0	39.84	−3	−6.7	19.1
2422	01:30:56:37	0	29	0	340.15	−3	−21	−1.1

Figure 1: Eye movement and driving data

In our study, we define saccade and fixation as follows [8].

- Saccade

  Saccade is caused by a change in the road situation or the appearance of pedestrians or cars. It is considered a perception factor in the driving model.

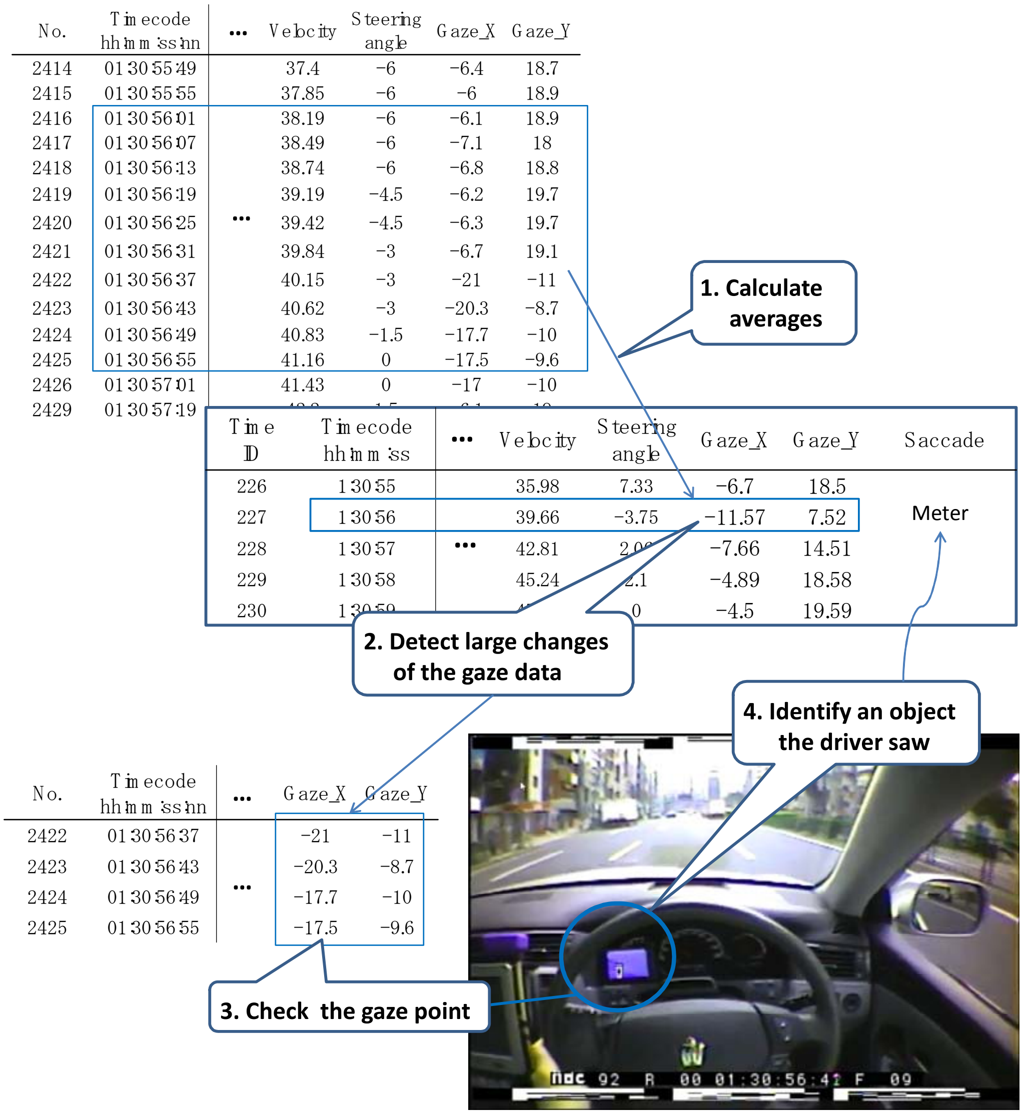
- Fixation

  Fixation is regarded as a cognitive factor in which the driver determines the next action by recognizing changes in the environment and objects. It is also related to the perception of temporal changes, such as signal changes and road signs.

2.2 Data Arrangement

We identify a driver’s cognitive load induced by a saccade in one second, and average the raw data for each second. Tracking saccade events, we detect such targets as the speedometer, rear-view mirror, car navigation device, road signs, and pedestrians. In addition, we define the target as unknown when we cannot detect the target of a saccade event, such as looking away. This raw data arrangement consists of the following steps [5-6].

- Step 1. Collect a set of raw data measured over a second and average each attribute of driving data (Figure 2, Step 1).
- Step 2. Detect large changes of eye movement that indicate a saccade event and obtain a set of time codes (Figure 2, Step 2).
- Step 3. Calculate the gaze point corresponding to a saccade from the raw data (Figure 2, Step 3).



No.	Timecode hh:mm:ss:nn	…	Velocity	Steering angle	Gaze_X	Gaze_Y
2414	01:30:55:49		37.4	−6	−6.4	18.7
2415	01:30:55:55		37.85	−6	−6	18.9
2416	01:30:56:01		38.19	−6	−6.1	18.9
2417	01:30:56:07		38.49	−6	−7.1	18
2418	01:30:56:13		38.74	−6	−6.8	18.8
2419	01:30:56:19		39.19	−4.5	−6.2	19.7
2420	01:30:56:25	…	39.42	−4.5	−6.3	19.7
2421	01:30:56:31		39.84	−3	−6.7	19.1
2422	01:30:56:37		40.15	−3	−21	−11
2423	01:30:56:43		40.62	−3	−20.3	−8.7
2424	01:30:56:49		40.83	−1.5	−17.7	−10
2425	01:30:56:55		41.16	0	−17.5	−9.6
2426	01:30:57:01		41.43	0	−17	−10
2429	01:30:57:19		[illegible]	[illegible]	[illegible]	[illegible]

Time ID	Timecode hh:mm:ss	…	Velocity	Steering angle	Gaze_X	Gaze_Y	Saccade
226	1:30:55		35.98	7.33	−6.7	18.5	
227	1:30:56		39.66	−3.75	−11.57	7.52	Meter
228	1:30:57	…	42.81	[illegible]	−7.66	14.51	
229	1:30:58		45.24	2.1	−4.89	18.58	
230	1:30:59		[illegible]	0	−4.5	19.59	

2. Detect large changes of the gaze data
4. Identify an object the driver saw

No.	Timecode hh:mm:ss:nn	…	Gaze_X	Gaze_Y
2422	01:30:56:37		−21	−11
2423	01:30:56:43	…	−20.3	−8.7
2424	01:30:56:49		−17.7	−10
2425	01:30:56:55		−17.5	−9.6

Figure 2: Data arrangement for each second. Driving data is averaged, and saccade events are detected

- Step 4. Identify an object (e.g., a meter) within the gaze point through video analysis (Figure 2, Step 4). If the object is identified, it is used as the value of the saccade attribute.

Step 4 is currently performed manually, rather than automatically. In Figure 2, the rear-view mirror is identified as the saccade target.

2.3 Data Transformation

A driver's cognitive state can be characterized by a sequence of saccades; however, we employ only adjacent saccades within a short time period (5 to 10 sec.) to predict the cognitive state for recognition and decision making. This includes driving data after 5 sec. because some operations are intended by the driver and are considered predictable changes for the driver. Thus, driving data for a small number of saccade events are regarded as training data for learning.

- Step 1. Collect an ordered set of saccade event data in which "saccadeID" is inserted for each datum.
- Step 2. Add new attributes indicating differences within a short time period (5 sec. before and after), where each difference is represented by ⊿-second. This data is used for SVM learning.

In addition, we split the data at a constant interval and averaged the data for each interval from the raw driving data. Next, we divided each split data into several sections using the threshold data, and converted the qualitative data. For example, our Noda area data is divided into 252 data sections. This data is used to determine whether the data could be adapted to the mental model [8] in our system.

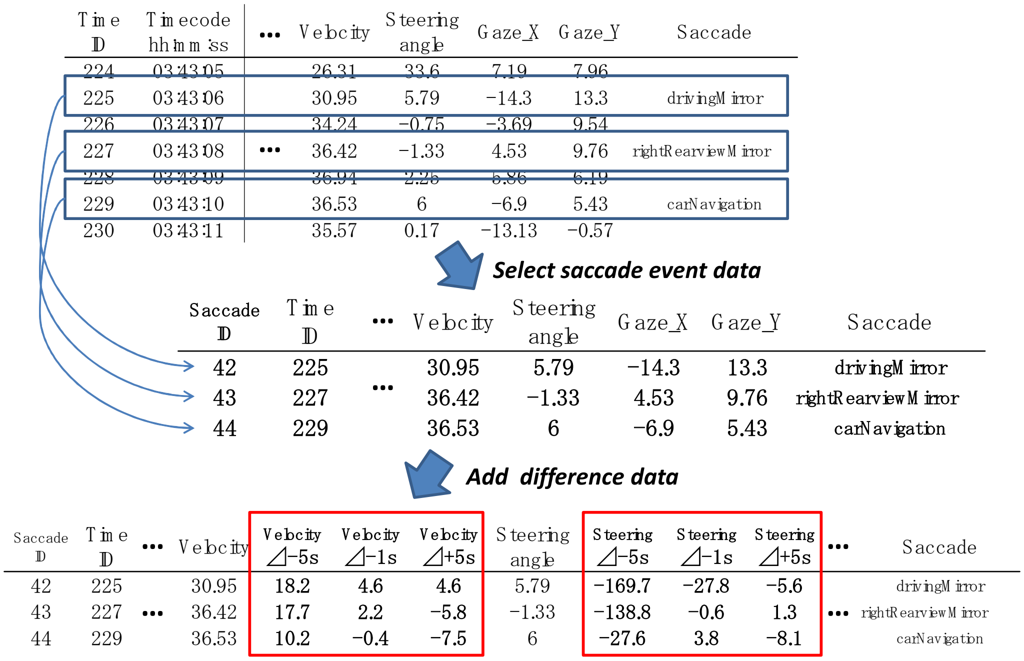
3 Graphic User Interface to Detect Distraction: Cog-Tracker

In this study, we develop the graphic user interface (GUI) Cog-Tracker (Figure 4) to extract eye movement features and detect distracted driving. This GUI system consists of video driving data, raw driving data, and transformed data based on the raw data. This is indicated by drawing a viewpoint circle at a given time during the driving video. This system is implemented in Java programming language. We incorporated the VideoLAN Client (VLC: http://www.videolan.org/vlc/) and its library on the video display into a device using the Windows OS. In a tablet PC using the Android OS, we used the Android OS's own library [1] on the video display. In addition, we incorporated a Google Map display function in the tablet PC to determine specific distracted car driving in the resulting location.

Figure 3 illustrates the process of data transformation consisting of the following steps.

3.1 Flow of the Cog-Tracker

Cog-Tracker uses the playback time of the driving video data as a trigger. Figure 5 illustrates the flow of the

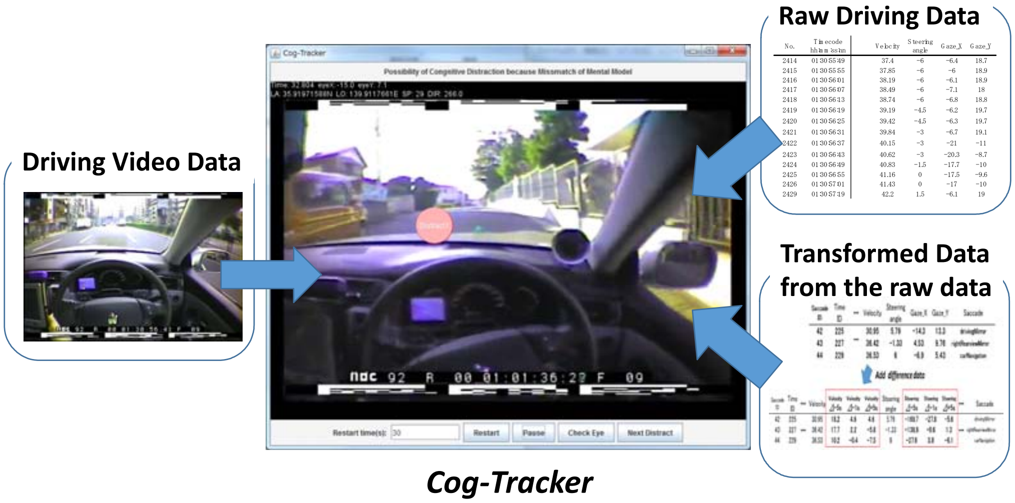


Time ID	Timecode hh:mm:ss	…	Velocity	Steering angle	Gaze_X	Gaze_Y	Saccade
~~224~~	~~03:43:05~~		~~26.31~~	~~33.6~~	~~7.19~~	~~7.96~~	
225	03:43:06		30.95	5.79	−14.3	13.3	drivingMirror
~~226~~	~~03:43:07~~		~~34.24~~	~~−0.75~~	~~−3.69~~	~~9.54~~	
227	03:43:08	…	36.42	−1.33	4.53	9.76	rightRearviewMirror
~~228~~	~~03:43:09~~		~~36.94~~	~~2.26~~	~~5.86~~	~~6.19~~	
229	03:43:10		36.53	6	−6.9	5.43	carNavigation
230	03:43:11		35.57	0.17	−13.13	−0.57	

Saccade ID	Time ID	…	Velocity	Steering angle	Gaze_X	Gaze_Y	Saccade
42	225		30.95	5.79	−14.3	13.3	drivingMirror
43	227	…	36.42	−1.33	4.53	9.76	rightRearviewMirror
44	229		36.53	6	−6.9	5.43	carNavigation

Saccade ID	Time ID	…	Velocity	Velocity ⊿-5s	Velocity ⊿-1s	Velocity ⊿+5s	Steering angle	Steering ⊿-5s	Steering ⊿-1s	Steering ⊿+5s	…	Saccade
42	225		30.95	18.2	4.6	4.6	5.79	−169.7	−27.8	−5.6		drivingMirror
43	227	…	36.42	17.7	2.2	−5.8	−1.33	−138.8	−0.6	1.3	…	rightRearviewMirror
44	229		36.53	10.2	−0.4	−7.5	6	−27.6	3.8	−8.1		carNavigation

Figure 3: Data transformation


Driving Video Data
Raw Driving Data
Transformed Data from the raw data
Add difference data
Cog-Tracker
Cog-Tracker
Possibility of Cognitive Distraction because Missmatch of Mental Model
Time: 32.804 eyeX:-15.0 eyeY: 7.1
LA: 35.91971588N LO: 139.9117081E SP: 29 DIR: 266.0
Restart time(s): 30
Restart
Pause
Check Eye
Next Distract

Raw Driving Data

No.	Timecode hh:mm:ss:nn	Velocity	Steering angle	Gaze_X	Gaze_Y
2414	01 30 55:49	37.4	−6	−6.4	18.7
2415	01 30 55:55	37.85	−6	−6	18.9
2416	01 30 56:01	38.19	−6	−6.1	18.9
2417	01 30 56:07	38.49	−6	−7.1	18
2418	01 30 56:13	38.74	−6	−6.8	18.8
2419	01 30 56:19	39.19	−4.5	−6.2	19.7
2420	01 30 56:25	39.42	−4.5	−6.3	19.7
2421	01 30 56:31	39.84	−3	−6.7	19.1
2422	01 30 56:37	40.15	−3	−21	−11
2423	01 30 56:43	40.62	−3	−20.3	−8.7
2424	01 30 56:49	40.83	−1.5	−17.7	−10
2425	01 30 56:55	41.16	0	−17.5	−9.6
2426	01 30 57:01	41.43	0	−17	−10
2429	01 30 57:19	42.2	1.5	−6.1	19

Transformed Data from the raw data

Saccade ID	Time ID	…	Velocity	Steering angle	Gaze_X	Gaze_Y	Saccade
42	225		30.95	5.76	−14.3	13.3	[illegible]
43	227	…	38.42	−1.30	4.53	9.76	[illegible]
44	229		38.53	0	−6.9	5.43	[illegible]

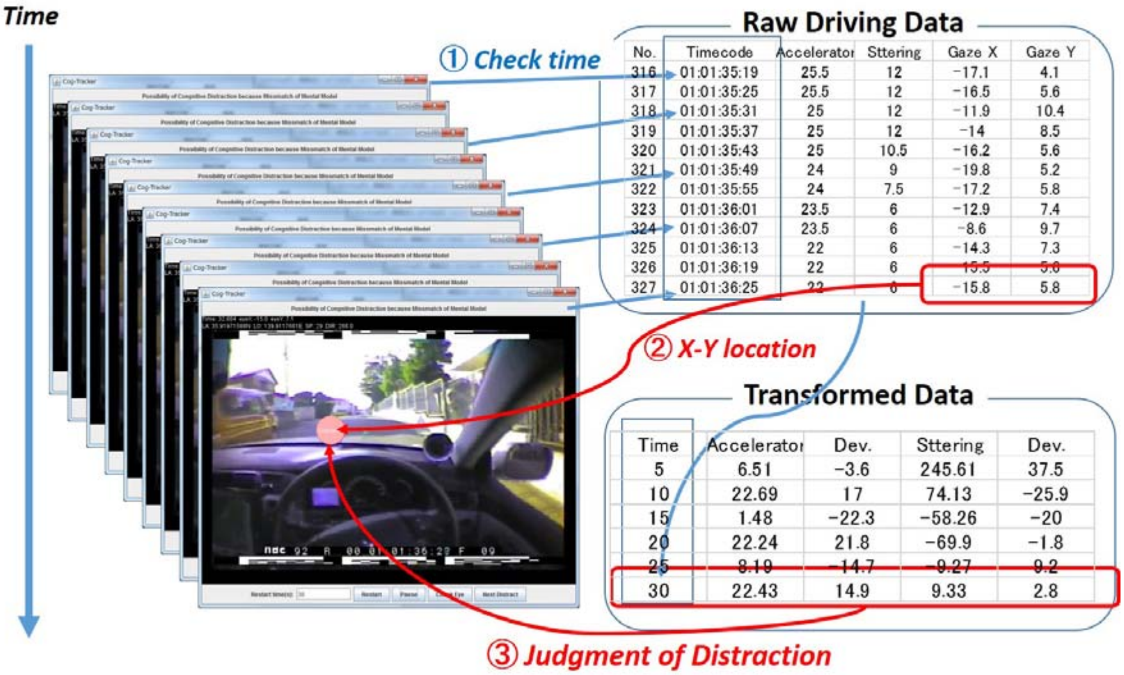
Add difference data

Saccade ID	Time ID	…	Velocity	Velocity Δ-5s	Velocity Δ-1s	Velocity Δ+5s	Steering angle	Steering Δ-5s	Steering Δ-1s	Steering Δ+5s	…	Saccade
42	225		30.95	[illegible]	[illegible]	[illegible]	5.76	[illegible]	[illegible]	[illegible]		[illegible]
43	227	…	38.42	[illegible]	[illegible]	[illegible]	−1.33	[illegible]	[illegible]	[illegible]	…	[illegible]
44	229		38.53	[illegible]	[illegible]	[illegible]	0	[illegible]	[illegible]	[illegible]		[illegible]

Figure 4: Cog-Tracker: This system extracts video driving data, raw data, and transformed data from the raw data. This is indicated by drawing a viewpoint circle at the current time during the driving video

Cog-Tracker. The Cog-Tracker flow is as follows:

- Step 0. Cog-Tracker plays the driving video data using VideoLAN Client (VLC) via the API library. VLC displays images at 30 frames per second.
- Step 1. Cog-Tracker checks the playback time of the driving video from VLC.
- Step 2. Cog-Tracker confirms eye location from the raw driving data at the playback time checked in step 1 and displays eye location in the video that is playing.
- Step 3. Cog-Tracker judges whether or not the timing of the playback time is distracted car driving using the transformed data. If the timing is that of distracted car driving, Cog-Tracker displays the situation information.


Time
① Check time
② X-Y location
③ Judgment of Distraction
Cog-Tracker
Possibility of Cognitive Distraction because Missmatch of Mental Model
Restart time(s): 30
Restart
Pause
Check Eye
Next Distract

Raw Driving Data

No.	Timecode	Accelerator	Sttering	Gaze X	Gaze Y
316	01:01:35:19	25.5	12	−17.1	4.1
317	01:01:35:25	25.5	12	−16.5	5.6
318	01:01:35:31	25	12	−11.9	10.4
319	01:01:35:37	25	12	−14	8.5
320	01:01:35:43	25	10.5	−16.2	5.6
321	01:01:35:49	24	9	−19.8	5.2
322	01:01:35:55	24	7.5	−17.2	5.8
323	01:01:36:01	23.5	6	−12.9	7.4
324	01:01:36:07	23.5	6	−8.6	9.7
325	01:01:36:13	22	6	−14.3	7.3
326	01:01:36:19	22	6	−15.5	5.6
327	01:01:36:25	22	6	−15.8	5.8

Transformed Data

Time	Accelerator	Dev.	Sttering	Dev.
5	6.51	−3.6	245.61	37.5
10	22.69	17	74.13	−25.9
15	1.48	−22.3	−58.26	−20
20	22.24	21.8	−69.9	−1.8
25	8.19	−14.7	−9.27	9.2
30	22.43	14.9	9.33	2.8

Figure 5: Flow of cog-tracker

- Step 4. Return to step 1 at an interval (usually 1000/30 sec. because the video rate is 30 frames per second).

Cog-Tracker uses two methods to detect two types of distracted car driving.

- Detection of visual distraction using a support vector machine (SVM)

  In our previous study [6], we produced learning rules for visual distraction detection using an SVM [9]. Cog-Tracker can detect visual distraction by SVM determination using such rules on eye movement and driving data.

- Detection of cognitive distraction based on the mental model

  In our previous study [6], we demonstrated that it is possible to detect cognitive distraction using the driver's mental model and measuring saccade frequency. We used the saccade frequency information from eye movement data and a qualitative model [8] of the cognitive mental load to detect cognitive distractions that change the driver's internal state. Using this qualitative model, 80% of the obtained eye movement and driving data can be interpreted. In our previous study, we focused on the remaining portion of the eye movement and driving data that is inconsistent with the model and discovered that cognitive distraction generally occurs in a state with inconsistent data. Cog-Tracker can detect cognitive distraction by detecting a conflict between the qualitative model relating mental load to eye movement and the corresponding driving data. In addition, we focus on Harbluk's finding that saccade frequency is reduced during driving with cognitive distraction [3]. Cog-Tracker can confirm cognitive distraction using eye-movement data when saccade frequency is reduced and car speed is high (40km/h).

3.2 Display of Eye Movement State

The GUI combines a video recording of the driving, the eye movement data, and the driving data to confirm the video driving situation (Figure 4). It also records the location of eye movement in real time. In addition, it is possible to check the status of eye movement over a predetermined period of time (Figure 6).

3.3 Display Based on Visual Distraction Detection Using SVM

Using the rule learned by SVM for the training data (gathered data), Cog-Tracker can detect visual distraction in new data. Figure 7 presents a situation in which Cog-Tracker detects visual distraction. In this situation, Cog-Tracker also detects cognitive distraction at the same time.

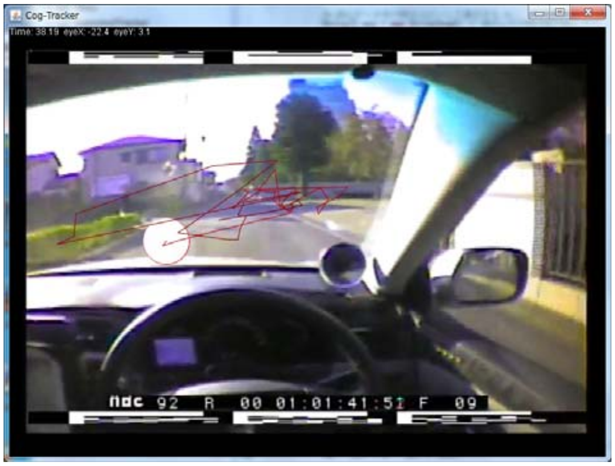
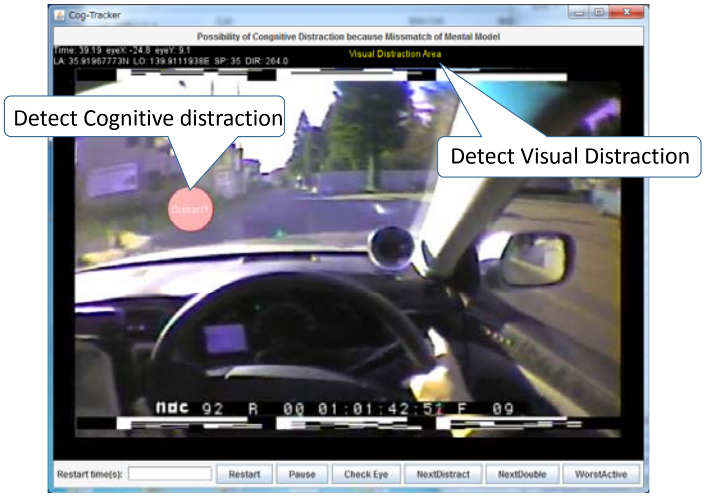

Cog-Tracker
Time: 38.19 eyeX: -22.4 eyeY: 3.1
ndc 92 R 00 01:01:41:51 F 09

Figure 6: Visualization of eye movement over a certain period of time (5 sec. in the figure) presented by Cog-Tracker. The color becomes clearer as the visualization approaches the present time in the driving video



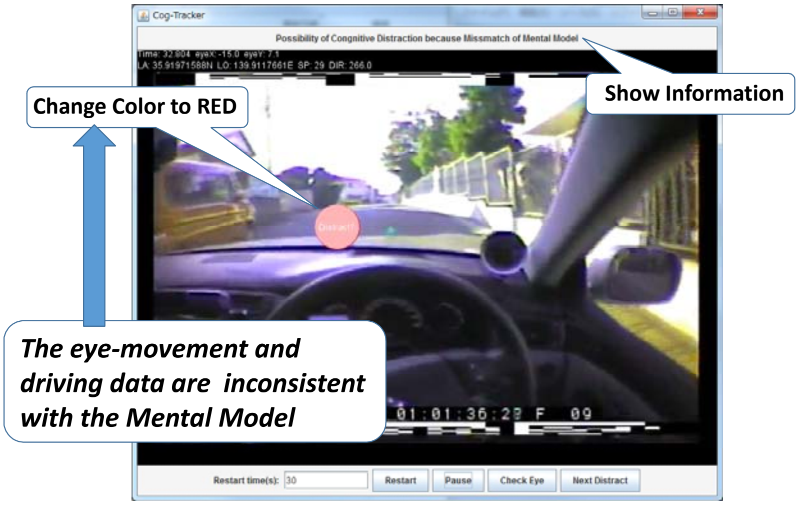
Possibility of Congnitive Distraction because Missmatch of Mental Model

Time: 39.19 eyeX: -24.8 eyeY: 9.1
LA: 35.91967773N LO: 139.9111938E SP: 35 DIR: 264.0

Figure 7: Detecting visual and cognitive distraction using the Cog-Tracker. Each detector detects each distraction at the same time

3.4 Display Based on Cognitive Distraction Detection Using a Mental Model

Cog-Tracker detects cognitive distraction by detecting when the eye movement and driving data are inconsistent with the mental model. Figure 8 depicts a situation in which Cog-Tracker detects cognitive distraction. In addition, Cog-Tracker confirms cognitive distraction based on eye movement data when the saccade frequency is reduced (no eye movement) and car speed is high (Figure 9).



Possibility of Congnitive Distraction because Missmatch of Mental Model

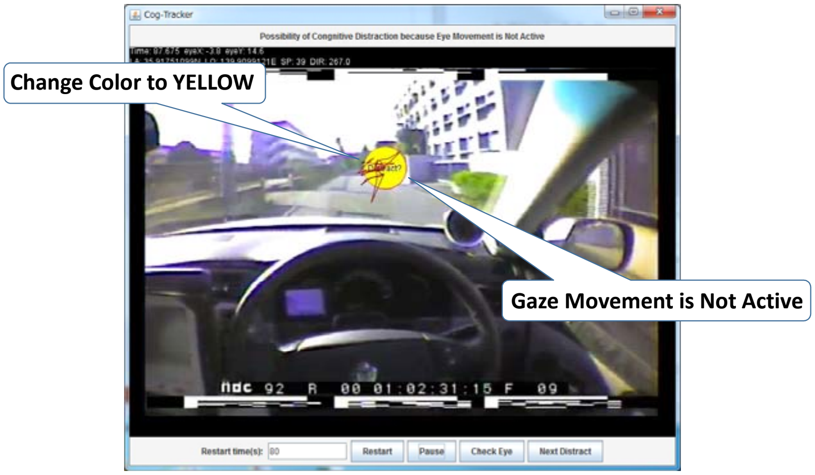
Time: 32.804 eyeX: -15.0 eyeY: 7.1
LA: 35.91971588N LO: 139.9117661E SP: 29 DIR: 266.0

Change Color to RED

Show Information

The eye-movement and driving data are inconsistent with the Mental Model

Figure 8: Detecting cognitive distraction using Cog-Tracker (1). The relationship between the eye movement and driving data was inconsistent with the mental model



Possibility of Congnitive Distraction because Eye Movement is Not Active

Time: 87.675 eyeX: -3.0 eyeY: 14.6
LA: 35.91751099N LO: 139.9099121E SP: 39 DIR: 267.0

Change Color to YELLOW

Gaze Movement is Not Active

Figure 9: Detecting cognitive distraction using Cog-Tracker (2). Saccade frequency is reduced (no eye movement)

3.5 Display on the Tablet PC

Our Tablet PC (Android) is a Google Nexus 10. Developing an Android tablet application is comparatively easy because the development language basically conforms to Java (Android SDK [1]). Figure 10 depicts its implementation on a tablet PC. The system on the tablet PC is able to check the current location using map information. Using this system, we can specify distracted driving at accident prone locations.

4 Discussion

Using Cog-Tracker on a Windows device or an Android tablet PC, we can detect visual and cognitive distraction during vehicle operation. In detecting cognitive distractions, many situations (25%) are detected as visual distractions at the same time. Figure 11 presents an example of cognitive and visual distraction. In this scene, the driver increased the ratio of saccade and fixation (in the driver's mental model [8], the parameter of 'env' is up). This means the driver was visually gathering information at this timing. Usually in these situations, the driver’s resource consumption due to environmental change must increase (in the model, the parameter is 'used_p'). However, in this timing, 'used_p' decreased. This scene is thus inconsistent with the driver’s mental model. We checked the reason and found that this driver increased speed at this time (then 'perform+' parameter is increased in the model). As a result, we were finally able to confirm that the 'used_p' parameter increased. In addition, this driver drove while looking at other objects (here, a vending machine). Using Cog-Tracker, we were able to discover dangerous driving as a result of detecting cognitive and visual distraction. In a future study, we will consider

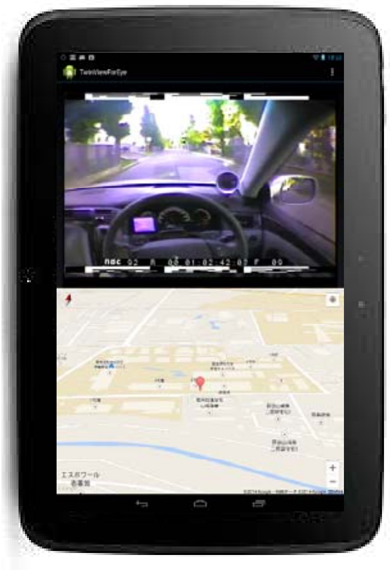


Figure 10: Cog-Tracker implementation on a tablet PC. This system can display video and map information to drivers

the relationship between visual distraction and cognitive distraction.

We found that in 90% of detected cognitive distraction situations in which the driver drives at high speed (40km/h), saccade frequency is reduced (i.e., no eye movement). Thus, the driver's eye movement is not active during driving, which is inconsistent with the qualitative model. However, this is consistent with the fact that saccade frequency during driving with cognitive distraction is reduced, as Harbluk proposed [3]. However, some situations in which the saccade frequency is reduced are expected to present less risk because the driver is driving stably on a wide road. Announcements concerning distractions made to the driver, the subject of our study, are meaningless in such situations. In future work, distractions detected with consideration of risk based on location information will become important.

5 Conclusion

In this study, we designed a new GUI to detect distracted car driving using the driver’s eye movement and driving data. Our system employs an eye movement measurement sensor to collect data on the driver’s eye movements, and a driving data measurement sensor to gather driving data on speed, acceleration, and steering, as well as GPS data for location information. The system measures the driver’s eye movements and the driving situation based on the information obtained from each sensor and detects distracted car driving using the driver’s mental model. In our study, we implemented our system on a Windows PC and a tablet PC to detect distracted driving. We expect that our system will

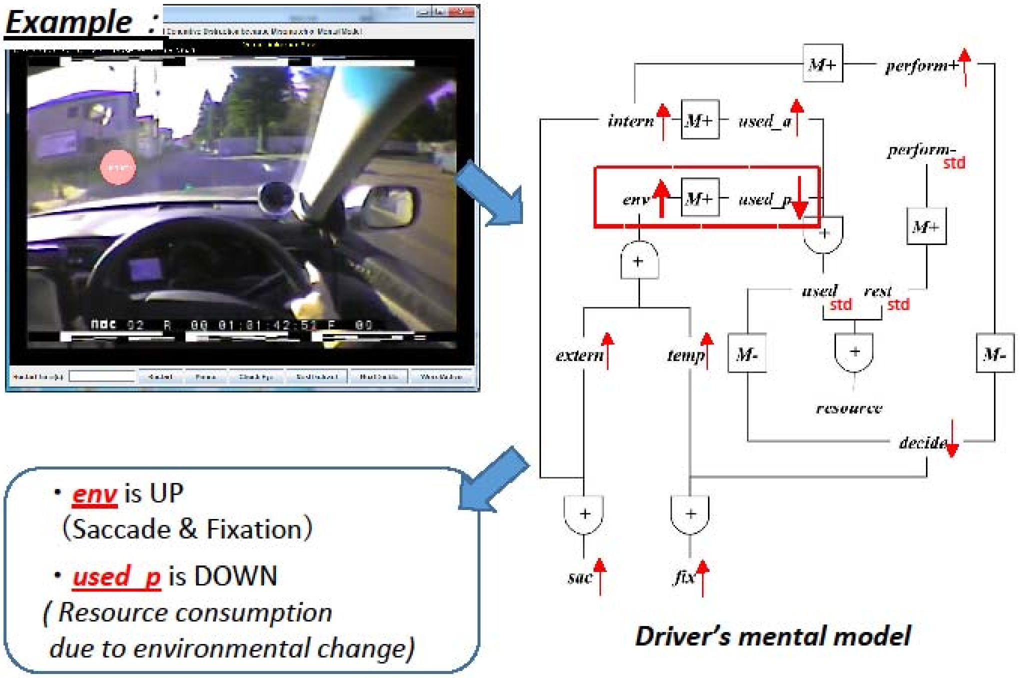


Figure 11: Typical scene of cognitive and visual distraction. The left image depicts cognitive and visual distractions. The figure on the right illustrates the driver's mental model [8] of the scene and detection inconsistent with the model

reduce driving risk by providing advice and urging caution by voice utterance from the tablet PC when the system detects distracted driving.

References

[1] Android SDK Guide, http://www.android.com/.

[2] Taku Harada, Hirotoshi Iwasaki, Kazuaki Mori, Akira Yoshizawa and Fumio Mizoguchi, “Evaluation Model of Cognitive Distraction State Based on Eye Tracking Data Using Neural Networks,” *International Journal of Software Science and Computational Intelligence (IJSSCI),* 6(1):1-16, 2014.

[3] Joanne L. Harbluk, Y. Ian Noy, Patricia L. Trbovich and Moshe Eizenman, “An On-Road Assessment of Cognitive Distraction: Impacts on Drivers’ Visual Behavior and Braking Performance,” *Accident Analysis & Prevention*, 39(2):372-379, 2007.

[4] Arien Mack and Irvin Rock, *Inattentional Blindness*, MIT Press, 1998.

[5] Fumio Mizoguchi, Hayato Ohwada, Hiroyuki Nishiyama and Hirotoshi Iwasaki, “Identifying Driver’s Cognitive Load Using Inductive Logic Programming,” Inductive Logic Programming, Lecture Notes in Artificial Intelligence (LNAI), LNAI7842, pp. 166-177, 2013.

[6] Fumio Mizoguchi, Hiroyuki Nishiyama and Hirotoshi Iwasaki, “A New Approach to Detecting Distracted Car Drivers Using Eye-Movement Data,” The 13th IEEE International Conference on Cognitive Informatics & Cognitive Computing, ICCI*CC 2014, pp. 266-272, 2014.

[7] NHTSA, TSA Distracted Driving Research Plan, [PDF] DOT-HS-811-299, http://www.nhtsa.gov/Research/Human+Factors/Distraction, April 2010.

[8] Shinichiro Sega, Hirotoshi Iwasaki, Hironori Hiraishi and Fumio Mizoguchi, “Qualitative Reasoning Approach to a Driver’s Cognitive Mental Load,” *International Journal of Software Science and Computational Intelligence,* 3(4):18-32, 2011.

[9] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer-Verlag, 1995.

[10] Yutaka Yoshida, Hayato Ohwada and Fumio Mizoguchi, “Temporal Discretization Method and Naive Bayes Classifier for Classifying Car Driver’s Cognitive Load,” 29th International Conference on Computers and Their Applications, pp. 9-14, 2014.

Hiroyuki Nishiyama is an Associate Professor at Tokyo University of Science. He obtained his Ph.D. from Tokyo University of Science in 2000. He is interested in the study of designing human-machine cooperative systems, user interface, multi-agent systems and network security based on Artificial Intelligence. He is a member of IEEE and ACM. His e-mail address is hiroyuki@rs.noda.tus.ac.jp.

Akira Yoshizawa is a project manager of Human Machine Interface (HMI) Team in Research and Development Group of Denso IT Laboratory Inc. His main research interests focus on human factors aspects of road safety, intelligent automotive HMI systems, and advanced driver assistant systems. He is a member of Japan Ergonomics Society, Vision Society of Japan and Human Interface Society.

Hirotoshi Iwasaki (Photo not available) received the B.S. and M.S. degrees in Electrical Engineering from Nagoya University in 1988 and 1990, Ph.D. degree from Tokyo University of Science in 2008, and Master of Management of Technology from Tokyo University of Science in 2015, respectively. Ph.D. From 1990 to 2000, he worked for Denso Cooporation. Since 2000, he has been on loan to Denso IT Laboratory, Inc. He received the best paper award on W2GIS2011, ICCI*CC11. His research interests include in-vehicle information systems and intelligent interface, and recommender systems.

Fumio Mizoguchi is an Emeritus Professor and a Researcher of Next Generation of Data Mining Division at Tokyo University of Science. He obtained his Ph.D. from Tokyo University in 1978. He is a senior research associate at the Center for the Study on Language and Information, and a member of the editorial boards of Intelligent Systems. He was also a member of the editorial boards of Artificial Intelligence Journal, New Generation Computing Journal, and the Journal of Logic Programming. He is a member of the judge committee of the Ministry of Education, Culture, Sports, Science and Technology in Japan. He has published more than 150 papers and thirty books in computer science. He started up his venture company in 2001, and he is the chief executive officer of WisdomTex, Inc. His e-mail address is mizo@wisdomtex.com.

Zero-Jitter Semi-Fixed-Priority Scheduling with Harmonic Periodic Task Sets

Hiroyuki Chishiro* and Nobuyuki Yamasaki*
Keio University, Yokohama, JAPAN

Abstract

Real-time systems such as humanoid robots require low jitter and high Quality of Service (QoS). An imprecise computation is one of the solutions to improve QoS but dynamic-priority imprecise real-time scheduling has high jitter. Semi-fixed-priority scheduling was presented to achieve low jitter and high QoS for imprecise computation. Unfortunately, a semi-fixed-priority scheduling algorithm, called Rate Monotonic with Wind-up Part (RMWP), has high jitter if the actual case execution time (ACET) of each task is shorter than its worst case execution time (WCET). We propose a new semi-fixed-priority scheduling algorithm, called Rate Monotonic with Wind-up Part++ (RMWP++), to achieve the zero-jitter of each task with harmonic periodic task sets. The zero-jitter technique adds the previous and post optional parts to the extended imprecise computation model that has a second mandatory (wind-up) part. We prove that the jitter of each task in RMWP++ is always zero and the least upper bound of RMWP++ is one with harmonic periodic task sets on uniprocessors. Simulation results show that RMWP++ achieves the zero-jitter and has a smaller number of context switches than RMWP, if the ACET of each task is shorter than its WCET.

Key Words: Semi-fixed-priority scheduling, schedulability, imprecise computation, jitter, harmonic periodic task sets.

1 Introduction

Real-time systems such as humanoid robots [16] are composed of real-time scheduling algorithms that have used worst case execution time (WCET) to schedule real-time tasks. However, the analysis of the WCET is difficult on current real-time systems due to both hardware and software complexities. Moreover, actual case execution time (ACET) in humanoid robots tends to change from time to time, because their behaviors depend on their environments. In addition, these robots usually have periodic real-time tasks with harmonic relationships (harmonic periodic task sets) where task periods are integer multiples of each other. Harmonic periodic task sets improve the utilization bound of real-time scheduling [12] and the precision of schedulability test [2], compared with general task sets that do not have a relationship among the task periods. These robots must control motors with low quality of service (QoS) under overloaded conditions. However, the traditional real-time scheduling model, called Liu and Layland's model [15], does not support the overloaded conditions, and hence the imprecise computation model [14] was presented.

The imprecise computation model is one of the techniques used to cope with such uncertainty. The crucial point is that the computation is split into two parts: mandatory part and optional part. A mandatory part affects the correctness of the result and an optional part only affects QoS. By restricting the execution of the optional part to only after the completion of the mandatory part, real-time applications based on the imprecise computation model can provide the correct output with lower QoS, by terminating the optional part. However, the imprecise tasks in humanoid robots require outputting the results to their actuators. When the imprecise tasks terminate or complete their optional parts, the imprecise computation model cannot guarantee completing them by their deadlines. In order to overcome the weakness of the imprecise computation model, we use the extended imprecise computation model [13] with a second mandatory (wind-up) part.

Extended imprecise tasks for humanoid robots such as MPEG decoder, object detection, and path search can be adapted to the extended imprecise computation model because they must guarantee completing their wind-up parts by their deadlines. In real-time scheduling of extended imprecise tasks, Mandatory-First with Wind-up Part (M-FWP) [13] has high jitter of the shortest period task due to Earliest Deadline First (EDF) [15] based dynamic-priority scheduling [4]. Hence, M-FWP is difficult to adapt to humanoid robots because the jitter-sensitive task such as the motor control task with the shortest period in humanoid robots requires the minimized jitter to achieve the precise motions. Unfortunately, fixed-priority scheduling such as Rate Monotonic (RM) [15] with low jitter of the shortest period task is also difficult to adapt to the extended imprecise computation model because one task may miss its deadline due to the overrun of the optional part.

In our previous research, we presented semi-fixed-priority scheduling [5] and a semi-fixed-priority scheduling algorithm based on RM, called Rate Monotonic with Wind-up Part (RMWP) [5], to achieve both low jitter and high schedula-

* Department of Information and Computer Science. E-mail: {chishiro,yamasaki}@ny.ics.keio.ac.jp.

bility in the extended imprecise computation model. Fortunately, the jitter of each task is zero in RMWP with harmonic periodic task sets when the ACET of each task is always equal to its WCET. Unfortunately, the jitter of each task is not zero when the ACET of each task is shorter than its WCET. Especially, high jitter causes robots to fall down, and hence to reduce jitter is very important to achieve the precise motion.

We propose a new semi-fixed-priority scheduling algorithm, called Rate Monotonic with Wind-up Part++ (RMWP++), to achieve the zero-jitter of each task with harmonic periodic task sets. The zero-jitter technique adds the previous and post optional parts to the extended imprecise computation model. We prove that the jitter of each task in RMWP++ is always zero and the least upper bound of RMWP++ is one with harmonic periodic task sets on uniprocessors. Simulation results show that RMWP++ achieves the zero-jitter and has a smaller number of context switches than RMWP, if the ACET of each task is shorter than its WCET.

The remainder of this paper is organized as follows: Section 2 describes the system model. Section 3 explains semi-fixed-priority scheduling and RMWP. Section 4 presents the RMWP++ algorithm and the zero-jitter technique. The effectiveness of RMWP++ is evaluated in Section 5. Section 6 discusses the compatibility of the zero-jitter technique for other real-time scheduling algorithms, and Section 7 concludes this paper.

2 System Model

Figure 1 shows the extended imprecise computation model [13]. The extended imprecise computation model adds the wind-up part to the imprecise computation model [14]. The imprecise computation model assumes that the processing to terminate or complete the optional part is not required. However, motor control tasks in humanoid robots require outputting the results to their actuators. They must guarantee schedulability of them, and hence the extended imprecise computation model has the wind-up part.

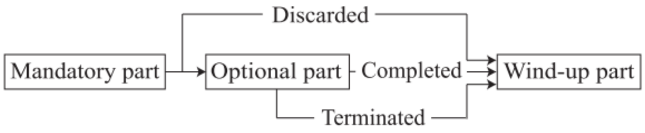


Figure 1: Extended imprecise computation model

We assume that the system has one processor and a task set Γ consisting of n tasks with harmonic period relations (harmonic periodic task sets). Task τ_i is represented as the following tuple (T_i, D_i, OD_i, am_i, m_i, o_i, aw_i, w_i): where T_i is the period, D_i is the deadline, OD_i is the optional deadline, am_i is the ACET of the mandatory part, m_i is the WCET of the mandatory part, o_i is the Required Execution Time (RET) of the optional part, aw_i is the ACET of the wind-up part, and w_i is the WCET of the wind-up part. The RET of each optional part tends to be underestimated or overestimated from time to time because humanoid robots run in uncertain environments. The relative deadline D_i of each task τ_i is equal to its period T_i. The j^{th} instance of τ_i is called job $\tau_{i,j}$. The utilization of each periodic task is defined as $U_i = (m_i + w_i)/T_i$. The reason why U_i does not include o_i is because the optional part of τ_i is a non-real-time part, and hence completing it is not relevant to scheduling the task set successfully. Hence, the utilization of the system within n tasks can be defined as $U = \Sigma_i\ U_i$. All tasks are ordered by decreasing priority ($T_1 \leq T_2 \leq ... \leq T_n$) and τ_1 has the highest priority. We also assume that both am_i and aw_i of each task can be analyzed by structural code analysis techniques [10-11], when they are ready to be executed.

In addition, we define the following symbols as follows.

- $o_{i,j}$: actual case RET of $\tau_{i,j}$
- $r_{i,j}$: release time of $\tau_{i,j}$
- $f_{i,j}$: finishing time of $\tau_{i,j}$
- $R_i(t)$: remaining execution time of τ_i at time t
- H_k: hyperperiod of the k^{th} task set

We define jitter as Relative Finishing Jitter (RFJ) [3]. RFJ is the maximum deviation of the finishing time of two consecutive jobs: $RFJ_i = \max_j |(f_{i,j+1} - r_{i,j+1}) - (f_{i,j} - r_{i,j})|$. We describe RFJ in Figure 2. In this case, the RFJ of τ_1 is the maximum of $|(f_{1,2} - r_{1,2}) - (f_{1,1} - r_{1,1})|$ and $|(f_{1,3} - r_{1,3}) - (f_{1,2} - r_{1,2})|$. In addition, the RFJ of the shortest period task τ_1 is defined as Shortest Period Jitter (SPJ).

An optional deadline is a time when an optional part is terminated and a wind-up part is released. Each wind-up part is ready to be executed after each optional deadline and can be completed if each mandatory part is completed by each optional deadline. Figure 3 shows the optional deadline of each task. Solid up arrow, solid down arrow, and dotted down arrow represent release time, deadline, and optional deadline, respectively. Task τ_1 completes its mandatory part by OD_1 and executes its optional part until OD_1. After OD_1, then τ_1 executes its wind-up part. In contrast, task τ_2 does not complete its mandatory part by OD_2. When τ_2 completes its mandatory part, τ_2 starts to execute its wind-up part and does not execute its optional part.

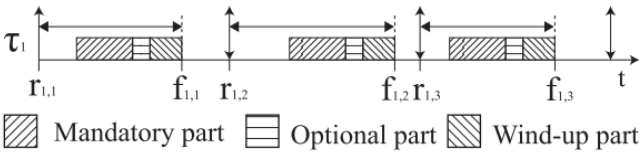


Figure 2: Relative finishing jitter

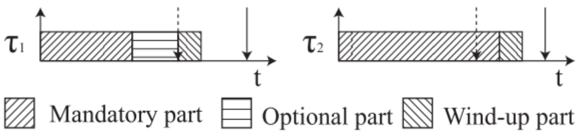


Figure 3: Optional deadline

3 Semi-Fixed-Priority Scheduling

Semi-fixed-priority scheduling [5] fixes the priority of each part in the extended imprecise task and changes the priority of each extended imprecise task only in two cases: (i) when the extended imprecise task completes its mandatory part and executes its optional part and (ii) when the extended imprecise task terminates or completes its optional part and executes its wind-up part. When there is no task which is ready to execute its mandatory or wind-up part, the optional part of each task is executed.

Figure 4 shows the difference between general scheduling such as RM and EDF in Liu and Layland's model [15] and semi-fixed-priority scheduling in the extended imprecise computation model. In general scheduling, when τ_i is released at 0, then $R_i(t)$ is set to $m_i + w_i$ and monotonically decreasing until $R_i(t)$ becomes 0 at $m_i + w_i$. In semi-fixed-priority scheduling, when τ_i is released at 0, then $R_i(t)$ is set to m_i and monotonically decreasing until $R_i(t)$ becomes 0 at m_i. When $R_i(t)$ is 0 at m_i, then τ_i sleeps until OD_i. When τ_i is released at OD_i, then $R_i(t)$ is set to w_i and monotonically decreasing until $R_i(t)$ becomes 0 at $OD_i + w_i$. If τ_i does not complete its mandatory part by OD_i, then $R_i(t)$ is set to w_i at the time when τ_i completes its mandatory part. The advantage of semi-fixed-priority scheduling against general scheduling is that semi-fixed-priority scheduling can execute optional parts between the time when completing mandatory parts and starting to execute wind-up parts. In general scheduling as well as semi-fixed-priority scheduling, τ_i completes its wind-up part by D_i.

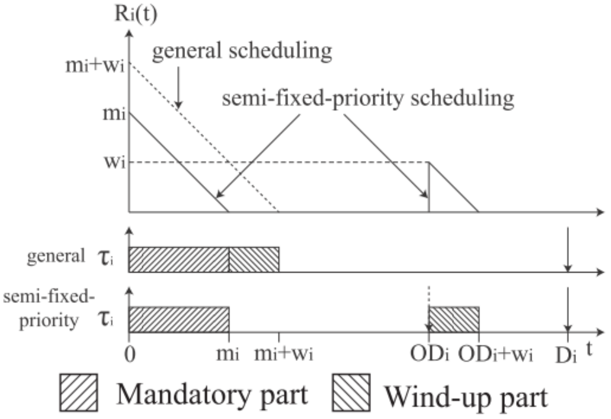


Figure 4: General scheduling and semi-fixed-priority scheduling

RMWP [5] is one of the semi-fixed-priority scheduling algorithms with the extended imprecise computation model on uniprocessors. As shown in Figure 5, RMWP manages three task queues: Real-Time Queue (RTQ), Non-Real-Time Queue (NRTQ), and Sleep Queue (SQ). RTQ manages tasks that are ready to execute their mandatory or wind-up parts in RM order. One task is not ready to execute its mandatory and wind-up parts simultaneously. NRTQ manages tasks that are ready to execute their optional parts in RM order. Every task in RTQ has higher priority than that in NRTQ. SQ manages tasks that have completed their optional parts by their optional deadlines or wind-up parts by their deadlines. The optional deadline of each task with harmonic periodic task sets is calculated by Response Time Analysis for Optimal Optional Deadline with Harmonic periodic task sets (RTA-OODH) [5]. The optimal optional deadline OD_k of task τ_k is defined as the time when the assignable time of τ_k in $[OD_k, D_k)$ is equal to w_k if the ACET of each task is always equal to its WCET. In order to describe RTA-OODH, we define the following terms in [5].

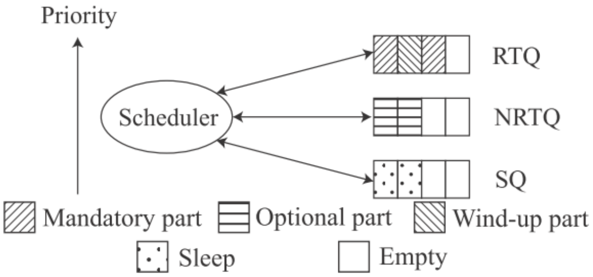


Figure 5: Task queue

Theorem 1 (From Theorem 1 in [5]). The worst case interference time $I_k^i(i < k)$ that is the upper bound time when τ_k is interfered with by τ_i is

$$I_k^i = \left\lceil \frac{T_k}{T_i} \right\rceil (m_i + w_i).$$

Theorem 2 (From Theorem 5 in [5]). The assignable time A_k of task τ_k except w_k is

$$A_k = D_k - w_k - \sum\nolimits_{i=1}^{k-1} I_k^i.$$

Theorem 3 (From Theorem 6 in [5]). The worst case interference time I_k of τ_k in $[0, OD_k)$ is

$$I_k = \sum\nolimits_{i=1}^{k-1} I_k^i (\left\lceil \frac{OD_k}{T_i} \right\rceil m_i + \left\lceil \frac{OD_k - OD_i}{T_i} \right\rceil w_i).$$

By Theorems 1, 2, and 3, we introduce RTA-OODH as follows.

Theorem 4 (From Theorem 7 in [5]). The optimal optional deadline OD_k of task τ_k with harmonic periodic task sets is

$$OD_k = A_k + I_k.$$

Using RTA-OODH, simulation results show that the reward ratio of the optional part in RMWP is higher than that in M-FWP with harmonic periodic task sets [5] though the optional deadline of each task in RMWP is calculated offline.

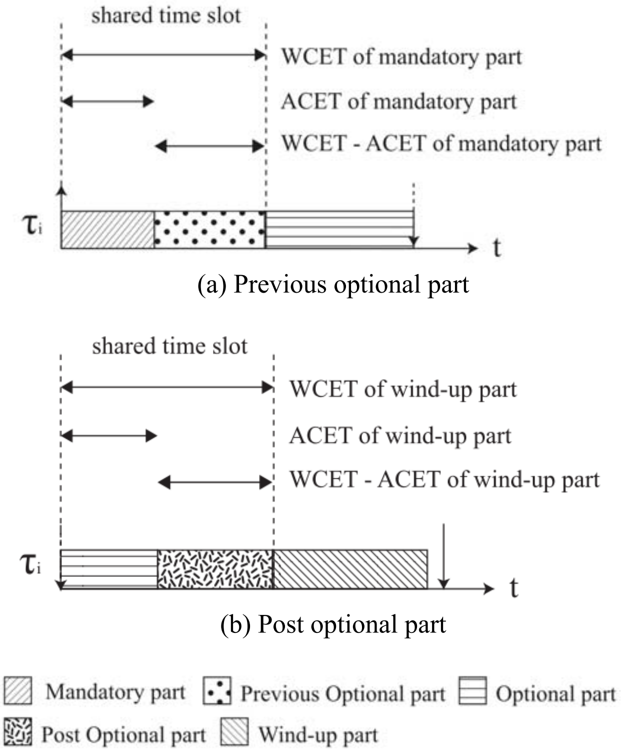
4 The RMWP++ Algorithm

In this section, we present the RMWP++ algorithm that has the technique to achieve the zero-jitter. We first describe previous and post optional parts in the extended imprecise computation model. We next introduce the overview of the RMWP++ algorithm.

The zero-jitter technique adds the previous and post optional parts to the extended imprecise computation model. Using the zero-jitter technique, the jitter of each task is always zero. The previous and post optional parts execute the optional part, as shown in Figure 1.

Figure 6 shows the previous and post optional parts. The previous optional part is executed after its mandatory part if am_i of each task is shorter than its m_i in the shared time slot. On the other hand, the post optional part is executed before its wind-up part if aw_i of each task is shorter than its w_i in the shared time slot. As previously mentioned, both am_i and aw_i of each task can be analyzed when they are ready to be executed, and hence the wind-up part delays its execution in the interval of w_i - aw_i. Moreover, the priorities of the previous and post optional parts of each task are equal to its mandatory and wind-up parts. That is to say, RTQ manages tasks that are ready to execute their previous and post optional parts. When the previous or post optional part of each task completes its optional part, the task executes its idle part (i.e., the task becomes idle but does not switch its context) that does nothing and the priority of which is also equal to its mandatory and wind-up parts. The reason why each task executes its idle part is to achieve the zero-jitter of each task, regardless of the ACET of each task. Therefore, the previous and post optional parts do not degrade schedulability, compared with if mandatory and wind-up parts are executed in their WCETs.

Figure 7 shows the overview of the RMWP++ algorithm. RMWP++ executes nine scheduling events when their conditions are met. The difference between RMWP++ and RMWP is the behavior, if the ACET of each task is shorter than its WCET, in scheduling events (2.a), (3), (4.a), (5.b.i),



(a) Previous optional part

(b) Post optional part

Figure 6: Previous and post optional parts

(1) When τ_i becomes ready, set $R_i(t)$ to m_i, dequeue τ_i from SQ and enqueue τ_i to RTQ. If τ_i has the highest priority in RTQ, preempt the current task.

(2) When τ_i completes its mandatory part, set $R_i(t)$ to o_i .

 (a) If am_i is shorter than m_i, execute its previous optional part until m_i.

 (b) Otherwise, dequeue τ_i from RTQ and enqueue τ_i to NRTQ. If there are one or multiple tasks in RTQ or NRTQ which have higher priority than τ_i, preempt τ_i.

(3) When τ_i does not complete its previous optional part at m_i, dequeue τ_i from RTQ and enqueue τ_i to NRTQ. If there are one or multiple tasks in RTQ or NRTQ which have higher priority than τ_i, preempt τ_i.

(4) When τ_i completes its optional part:

 (a) If τ_i executed its previous or post optional part in the last minute, execute its idle part.

 (b) Otherwise, dequeue τ_i from NRTQ and enqueue τ_i to SQ.

(5) When OD_i expires:

 (a) If τ_i is in RTQ and does not complete its mandatory part, do nothing.

 (b) If τ_i is in NRTQ or SQ, terminate and dequeue τ_i from NRTQ or SQ, set $R_i(t)$ to w_i and enqueue τ_i to RTQ. If τ_i has the highest priority in RTQ, preempt the current task.

  (i) If aw_i is shorter than its w_i, execute its post optional part until $OD_i + w_i - aw_i$.

  (ii) Otherwise, execute its wind-up part.

(6) When τ_i does not complete its post optional part at $OD_i + w_i - aw_i$, terminate τ_i and execute its wind-up part.

(7) When τ_i completes its wind-up part, dequeue τ_i from RTQ and enqueue τ_i to SQ.

(8) When there are one or multiple tasks in RTQ, perform RM in RTQ.

(9) When there is no task in RTQ and there are one or multiples tasks in NRTQ, perform RM in NRTQ.

Figure 7: The RMWP++ algorithm

and (6). If am_i of each task is shorter than its m_i, the previous optional part is ready to be executed at the time when τ_i completes its mandatory part. If aw_i of each task is shorter than its w_i, the post optional part is ready to be executed at its OD_i. RMWP++ as well as RMWP calculates the optional deadline by Theorem 4. We first analyze the jitter of RMWP++ with harmonic periodic task sets on uniprocessors, regardless of the ACET of each task.

Theorem 5 (Jitter of RMWP++ with Harmonic Periodic Task Sets). RMWP++ achieves the zero-jitter with harmonic periodic task sets on uniprocessors, regardless of the ACET of each task.

Proof. Regardless of the ACET of each task, the relative finishing time of each task is always the time when each task completes its wind-up part, as shown Figure 6. In addition, the worst case interference time of each job by higher priority jobs is always constant with harmonic periodic task sets on uniprocessors. It is clear that the RFJ of each task is always zero. Hence, this theorem holds.

We next analyze the least upper bound of RMWP++ with harmonic periodic task sets on uniprocessors.

Theorem 6 (Least Upper Bound of RMWP++ with Harmonic Periodic Task Sets). The least upper bound of RMWP++ with harmonic periodic task sets on uniprocessors is $U_{lub} = 1$.

Proof. RMWP++ generates the same schedule as RMWP if the ACET of each task is always equal to its WCET. Moreover, the relative finishing time of each task is always constant regardless of the ACET of each task by Theorem 5. Hence, the least upper bound of RMWP++ as well as RMWP with harmonic periodic task sets on uniprocessors is $U_{lub} = 1$ [5].

By Theorem 6, we prove that the least upper bound of RMWP++ is equal to that of RMWP on uniprocessors though if the previous and post optional parts are executed in RTQ.

5 Simulation Studies

5.1 Simulation Environments

This section studies the effectiveness of RMWP++ using four performance metrics. The simulation uses 1,000 task sets and compares RMWP++ with both RMWP and RM. In humanoid robots, there are tasks that have various periods. Therefore, the period T_i of each task τ_i is selected within [100, 200, 400, 800, 1600, 3200]. Each U_i is selected within [0.02, 0.03, 0.04, ..., 0.25] and splits U_i into two utilizations that are assigned to m_i and w_i, respectively. The CPU utilization U is selected within [0.3, 0.35, 0.4, ..., 1.0]. The simulation length of the k^{th} task set is 1, 024 times of the hyperperiod H_k.

The CPU utilization of $o_{i,j}$ is within the range of $[o_i - 0.05, o_i + 0.05]$, where o_i is selected within [0.1, 0.2, 0.3], represented such as RMWP++-10, RMWP++-20, and RMWP++-30, computed at every task release, because humanoid robots run in uncertain environments, and hence each $o_{i,j}$ is fluctuated. If the CPU utilization of $o_{i,j}$ is always equal to zero, the result is represented as RMWP++. Also, we consider the tasks, the ACETs of which tend to fluctuate from time to time in humanoid robots, and hence we evaluate three cases where ACET/WCET uniformly varies in the range of [0.5, 1.0], [0.75, 1.0], and 1.0.

We use four performance metrics to evaluate the effectiveness of RMWP++ from various perspectives. The performance metrics are defined as the following equations.

$$\mathit{Re\ ward\ Ratio} = \frac{\sum_k \sum_i \frac{T_i}{H_k} \sum_j \frac{o_{i,j}}{o_i}}{\#\ \mathit{of\ \ tasks\ \ in\ \ scheduled\ \ task\ \ sets}}$$

$$\mathit{Switch\ Ratio} = \frac{\sum_k \frac{\#\,\mathit{of\ context\ switches}}{H_k}}{\#\ \mathit{of\ \ tasks\ \ in\ \ scheduled\ \ task\ \ sets}}$$

$$\mathit{RFJ\ Ratio} = \frac{\sum_k \sum_i \frac{RFJ_i}{T_i}}{\#\ \mathit{of\ \ tasks\ \ in\ \ scheduled\ \ task\ \ sets}}$$

$$\mathit{SPJ\ Ratio} = \frac{\sum_k \frac{RFJ_1}{T_1}}{\#\ \mathit{of\ \ scheduled\ \ task\ \ sets}}$$

We do not show the result with respect to schedulability because it is clear that the least upper bounds of RMWP++, RMWP, and RM with harmonic periodic task sets on uniprocessors are equal to one by Theorem 6 (RMWP++) , results from [5] (RMWP), and results from [15] (RM), respectively.

In this paper, QoS is defined as the reward ratio. The reward ratio of each task is higher and higher, its QoS is higher and higher. The reason why the SPJ ratio is evaluated in this simulation is that the shortest period task τ_1 is usually the motor control task in humanoid robots, which strongly requires the low jitter.

5.2 Simulation Results

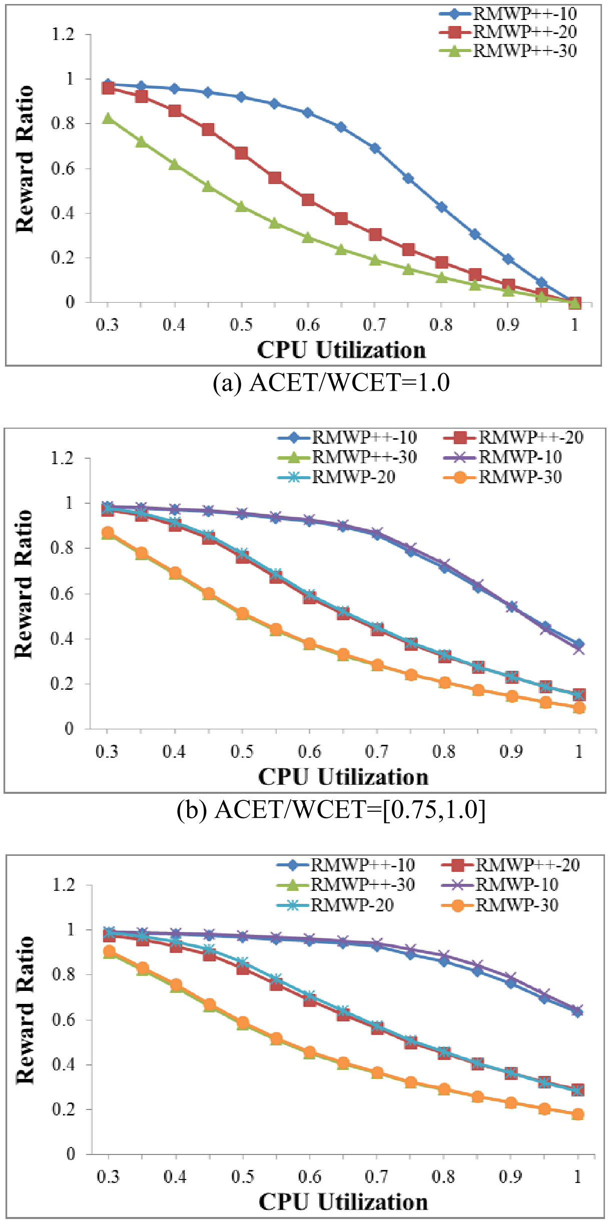
Figures 8, 9, 10, and 11 show the simulation results of Reward ratio, Switch ratio, RFJ ratio, and SPJ ratio, respectively. In Figures 8(a), 9(a), 10(a), and 11(a), the results of RMWP++ are the same as those of RMWP, and hence those of RMWP are omitted. In Figures 10(a), 10(b), 10(c), 11(a), 11(b), and 11(c), the RFJ ratios and SPJ ratios of RMWP and RMWP++ do not depend on the ACET of each optional part, and hence those of RMWP-10, RMWP-20, RMWP-30, RMWP++-10, RMWP++-20, and RMWP++-30 are omitted.

In Figure 8, the reward ratios of RMWP++-10, RMWP++-20, and RMWP++-30 are slightly lower than those in RMWP-

10, RMWP-20, and RMWP-30, respectively, because each task executes its idle part if there is remaining time of previous or post optional parts, as shown in Figure 6. However, the reward ratios of RMWP++-10, RMWP++-20, and RMWP++-30 are slightly lower than those of RMWP-10, RMWP-20, and RMWP-30, respectively, and hence the disadvantage of RMWP++ against RMWP is trivial.

In Figure 9, the switch ratios of RMWP++ and RMWP are higher than the switch ratio of RM because RMWP++ and RMWP support the extended imprecise computation model, which increases the number of context switches. In RMWP++-10, RMWP++-20, and RMWP++-30 as well as RMWP-10, RMWP- 20, and RMWP-30, the RET of each optional part is longer and longer, the switch ratio is lower and lower because of executing each part continuously more frequently. That is to say, the high utilization of optional part contributes the small number of context switches. In addition, if the ACET of each task is shorter than or equal to its WCET, the switch ratios of RMWP++-10, RMWP++-20, and RMWP++-30 are dramatically lower than those of RMWP-10, RMWP-20, and RMWP-30, respectively, which is one advantage of RMWP++ against RMWP.

In Figure 10(a), the RFJ ratios of RMWP++ and RM are always zero. In Figures 10(b) and 10(c), the RFJ ratios of RMWP and RM are higher than zero and the RFJ ratio of RMWP is higher than that of RM. When the CPU utilization is higher and higher or the ratio of ACET/WCET is wider and wider, the RFJ ratios of RMWP and RM are higher and higher.

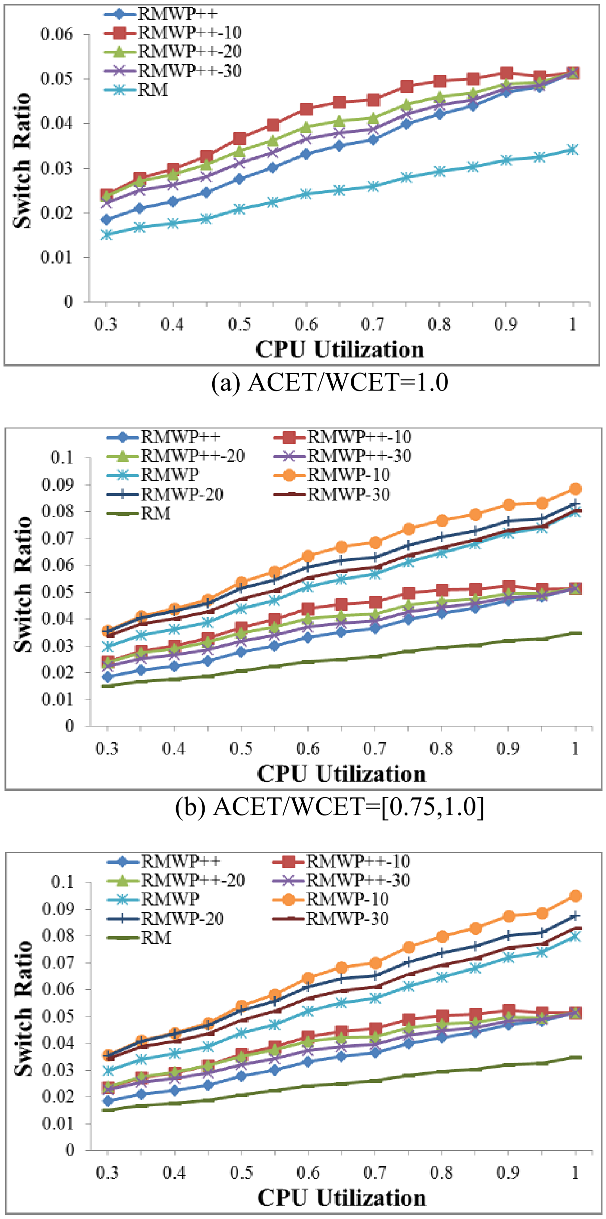


(a) ACET/WCET=1.0

(b) ACET/WCET=[0.75,1.0]

(c) ACET/WCET=[0.5,1.0]

Figure 8: Reward ratio



(a) ACET/WCET=1.0

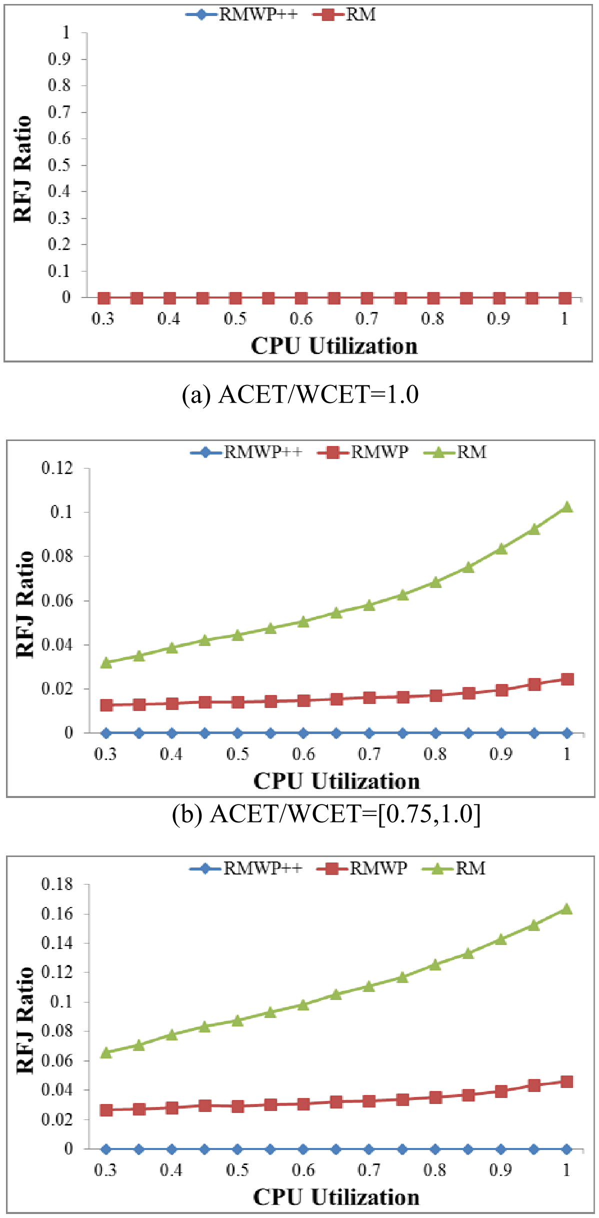
(b) ACET/WCET=[0.75,1.0]

(c) ACET/WCET=[0.5,1.0]

Figure 9: Switch ratio

In contrast, the RFJ ratio of RMWP++ is always zero by Theorem 5, and hence RMWP++ achieves the zero-jitter, regardless of the ACET of each task.

In Figure 11(a), the SPJ ratios of RMWP++ and RM are always zero as well as Figure 10(a). In Figures 11(b) and 11(c), RMWP++ is always zero but RMWP and RM are high because the ratio of ACET/WCET is fluctuated. Like Figures 10(b) and 10(c), when the ratio of ACET/WCET is wider and wider, the SPJ ratios of RMWP and RM are higher and higher. In contrast, unlike Figures 10(b) and 10(c), the SPJ ratios of RMWP and RM in Figures 11(b) and 11(c) are approximately constant, regardless of the CPU utilization. This is because the ratio of ACET/WCET is fluctuated using the uniform distribution. That is to say, the result of SPJ ratio depends on the distribution of the ratio of ACET/WCET. If the ratio of ACET/WCET is fluctuated using other distributions such as binomial distribution and exponential distribution, the result of SPJ ratio is different.

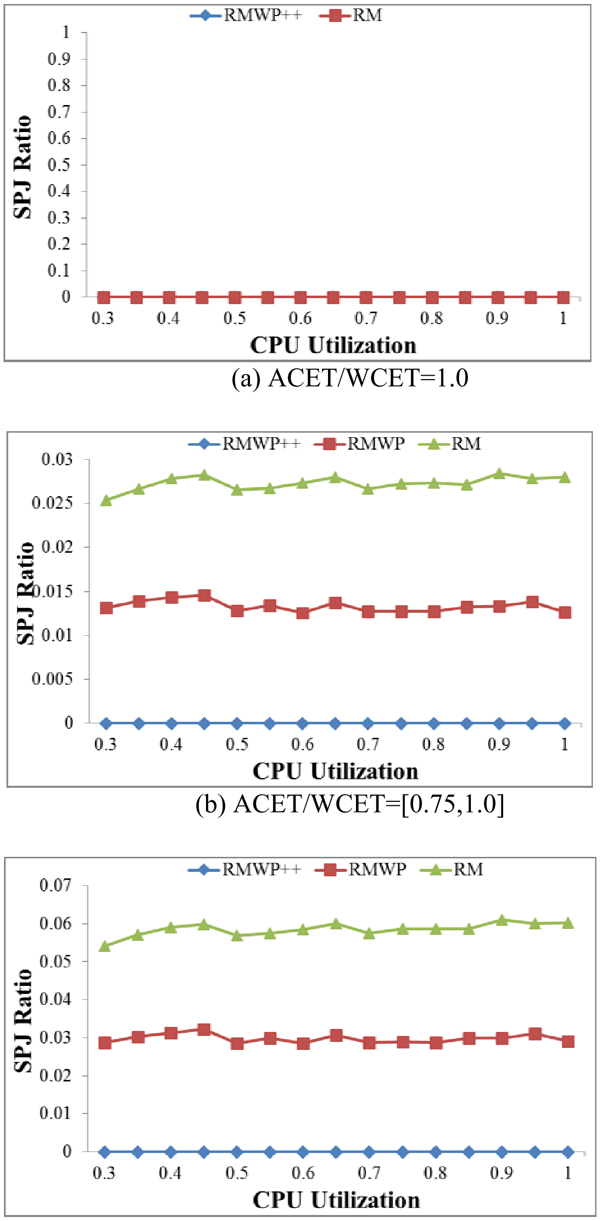


(a) ACET/WCET=1.0

(b) ACET/WCET=[0.75,1.0]

(c) ACET/WCET=[0.5,1.0]

Figure 10: RFJ ratio



(a) ACET/WCET=1.0

(b) ACET/WCET=[0.75,1.0]

(c) ACET/WCET=[0.5,1.0]

Figure 11: SPJ ratio

In the end, the zero-jitter of each task, regardless of the ratio of ACET/WCET, is very important to achieve the precise motion in humanoid robots [16], and hence RMWP++ is well suited to such robots.

6 Related Work

We discuss the compatibility of the zero-jitter technique for other real-time scheduling algorithms. General scheduling with Liu and Layland's model such as RM and EDF [15] can also achieve the zero-jitter of each task and reduce the number of context switches assuming that each task only has its wind-up part and sets its optional deadline to zero. Unfortunately, each task always executes its idle part if the ACET of each task is shorter than its WCET because each task does not have its optional part. The zero-jitter technique in semi-fixed-priority scheduling is more effective than that in general scheduling to execute optional parts. In contrast, M-FWP [13] is difficult to achieve the zero-jitter because M-FWP calculates the assignable execution time of each optional part online. Moreover, it is difficult to calculate the optional deadline as well as RMWP due to dynamic-priority scheduling [5]. Therefore, we conclude that RMWP is the highest compatibility of the zero-jitter technique in these algorithms.

We introduce other approaches to reduce jitter. Bini and Buttazzo address the problem of determining the region of the feasible task deadlines, called D-space, when tasks are scheduled by EDF [1]. Yomsi et al. improve the sensitivity of the deadline with harmonic periodic task sets for Deadline Monotonic scheduling [17]. They use Mesoid approach [18] to compute the worst case response time of the task and the minimum deadline reduction factor. This approach may be integrated to our approach, because this approach can be adapted to harmonic periodic task sets.

7 Conclusion

We proposed the RMWP++ algorithm that has the technique to achieve the zero-jitter of each task for semi-fixed-priority scheduling with harmonic periodic task sets on uniprocessors, called zero-jitter technique. The zero-jitter technique adds the previous and post optional parts to the extended imprecise computation model, in order to achieve the zero-jitter. We proved that the jitter of each task in RMWP++ is always zero if the ACET of each task is shorter than or equal to its WCET, and the least upper bound of RMWP++ is one (i.e., $U_{lub} = 1$) and equal to the least upper bounds of RMWP and RM. Therefore, there is no disadvantage of RMWP++ against RMWP with respect to schedulability. Simulation results show that the number of context switches in RMWP++ is smaller than that in RMWP. The disadvantage of RMWP++ against RMWP is to drop the reward ratio slightly. However, the degradation of reward ratio of RMWP++ against RMWP is not so many, and hence this disadvantage is trivial. We conclude that RMWP++ is more effective than RMWP to achieve the precise motion in humanoid robots.

In future work, we will integrate Yomsi's approach [17, 18] to our approach with harmonic periodic task sets that have tasks with shorter relative deadlines than periods. Also, we will implement RMWP++ in the RT-Est real-time operating system [7] that supports semi-fixed-priority scheduling in the extended imprecise computation model. In addition, we will adapt the zero-jitter technique to multiprocessor semi-fixed-priority scheduling [6, 8] and the practical imprecise computation model [9] that has multiple mandatory and optional parts to support many real-time applications.

Acknowledgement

This research was supported in part by Keio Gijuku Academic Development Funds, Keio Kougakukai, and CREST, JST.

References

[1] E. Bini and G. Buttazzo, "The Space of EDF Deadlines: The Exact Region and a Convex Approximation," *Real-Time Systems*, 41(1):27-51, January 2009.

[2] V. Bonifaci, A. Marchetti-Spaccamela, N. Megow, and A. Wiese, "Polynomial-Time Exact Schedulability Tests for Harmonic Real-Time Tasks," *Proceedings of the 34th IEEE Real-Time Systems Symposium*, pp. 236-245, December 2013.

[3] G. C. Buttazzo, *Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications*, *Springer*, 3rd Edition, September 2011.

[4] G. C. Buttazzo, "Rate Monotonic vs. EDF: Judgment Day," *Real-Time Systems*, 29(1):5-26, January 2005.

[5] H. Chishiro, A. Takeda, K. Funaoka, and N. Yamasaki, "Semi-Fixed-Priority Scheduling: New Priority Assignment Policy for Practical Imprecise Computation," *Proceedings of the 16th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, pp. 339-348, August 2010.

[6] H. Chishiro and N. Yamasaki, "Global Semi-Fixed-Priority Scheduling on Multiprocessors," *Proceedings of the 17th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, pp. 218-223, August 2011.

[7] H. Chishiro and N. Yamasaki, "RT-Est: Real-Time Operating System for Semi-Fixed-Priority Scheduling Algorithms," *Proceedings of the 2011 International Symposium on Embedded and Pervasive Systems*, pp. 358-365, October 2011.

[8] H. Chishiro and N. Yamasaki, "Experimental Evaluation of Global and Partitioned Semi-Fixed-Priority Scheduling Algorithms on Multicore Systems," *Proceedings of the 15th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing*, pp. 127-134, April 2012.

[9] H. Chishiro and N. Yamasaki, "Semi-Fixed-Priority Scheduling with Multiple Mandatory Parts," *Proceedings of the 16th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing*, pp. 1-8, June 2013.

[10] A. Colin and S. M. Petters, "Experimental Evaluation of Code Properties for WCET Analysis," *Proceedings of the 24th IEEE Real-Time Systems Symposium*, pp. 190-199, December 2003.

[11] A. Ermedahl, *A Modular Tool Architecture for Worst-Case Execution Time Analysis*, PhD Dissertation, Uppsala University, June 2003.

[12] M. Fan and G. Quan, "Harmonic Semi-Partitioned

Scheduling For Fixed-Priority Real-Time Tasks On Multi-Core Platform," *Proceedings of the 2012 Design, Automation & Test in Europe*, pp. 503-508, March 2012.

[13] H. Kobayashi and N. Yamasaki, "An Integrated Approach for Implementing Imprecise Computations," *IEICE Transactions on Information and Systems*, 86(10):2040-2048, October 2003.

[14] K. Lin, S. Natarajan, and J. Liu, "Imprecise Results: Utilizing Partial Computations in Real-Time Systems," *Proceedings of the 8th IEEE Real-Time Systems Symposium*, pp. 210-217, December 1987.

[15] C. L. Liu and J. W. Layland, "Scheduling Algorithms for Multiprogramming in a Hard Real-Time Environment," *Journal of the ACM*, 20(1):46-61, January 1973.

[16] I. Mizuuchi, Y. Nakanishi, Y. Sodeyama, Y. Namiki, T. Nishino, N. Muramatsu, J. Urata, K. Hongo, T. Yoshikai, and M. Inaba, "Advanced Musculoskeletal Humanoid Kojiro," *Proceedings of the 2007 IEEE-RAS International Conference on Humanoid Robots*, pp. 294-299, November 2007.

[17] P. M. Yomsi, L. George, Y. Sorel, and D. de Rauglaudre, "Improving the Sensitivity of Deadlines with a Specific Asynchronous Scenario for Harmonic Periodic Tasks scheduled by FP," *Proceedings of the 4th International Conference on Systems*, pp. 29-37, March 2009.

[18] P. M. Yomsi and Y. Sorel, "Extending Rate Monotonic Analysis with Exact Cost of Preemptions for Hard Real-Time Systems," *Proceedings of 19th Euromicro Conference on Real-Time Systems*, pp. 280-290, July 2007.

Hiroyuki Chishiro is a Research Associate in the Department of Information and Computer Science at Keio University. He received his B.S., M.S., and Ph.D. degrees from Keio University in 2008, 2010, and 2012, respectively. His research interests include real-time systems, operating systems, middleware, and trading systems.

Nobuyuki Yamasaki is a Professor in the Department of Information and Computer Science at Keio University. He is also a guest researcher in the Digital Human Research Center at Advanced Industrial Science and Technology. He received his B.S., M.S., and Ph.D. degrees from Keio University in 1991, 1993, and 1996, respectively. His research interests include parallel and distributed systems, real-time systems, system LSI, and robotics.

An Integration of Imprecise Computation Model and Real-Time Voltage and Frequency Scaling on Responsive Multithreaded Processor

Keigo Mizotani[1*], Yusuke Hatori[*], Yusuke Kumura[2*],
Masayoshi Takasu[3*], Hiroyuki Chishiro[*], and Nobuyuki Yamasaki[*]
Keio University, Yokohama, JAPAN

Abstract

As microprocessor performance grows, high throughput and power management have been important on embedded real-time systems. Real-Time Voltage and Frequency Scaling (RT-VFS) has been proposed to reduce power consumption and ensure real-time constraints. An imprecise computation model adds an optional part to Liu and Layland's model to improve the quality of computations. However, the trade-off between power consumption and quality of computations has not been well investigated on actual systems. This paper proposes the scheme to integrate an imprecise computation model and RT-VFS to improve the quality of computations and reduce the power consumption within real-time constraints. Moreover, we implement this scheme on Dependable Responsive Multithreaded Processor (D-RMTP), which is a prioritized simultaneous multithreaded processor for embedded real-time systems. We implement the proposed scheme by use of D-RMTP original features. Through experimental evaluation, we show that the proposed scheme satisfies both the lower energy consumption and higher quality of computations on actual systems. In particular, the proposed scheme achieves a maximum of 135% improvement of the quality of computations per energy consumption.

Key Words: Embedded real-time systems, power consumption, RT-VFS, imprecise computation model, responsive multithreaded processor.

1 Introduction

With the improvement of microprocessor performance, embedded real-time systems have required both low power consumption and high throughput. Voltage and Frequency Scaling (VFS) is known as a power management technique [4]. In embedded real-time systems, the VFS scheme must ensure real-time constraints even if tasks run at a lower frequency. Thus, Real-Time VFS (RT-VFS) is important in embedded real-time systems that require the low power consumption. An imprecise computation model [5] adds an optional part to Liu and Layland's model [6] to improve the quality of computations. In the imprecise computation model, a mandatory part is a real-time task and an optional part is no real-time task. When a task is ready to execute or executing its optional part and its deadline expires, its optional part is terminated to ensure its real-time constraint. However, the trade-off between power consumption and quality of computations has not been well investigated on actual systems.

This paper proposes an integration of an imprecise computation model and RT-VFS as the scheme to satisfy both the low power consumption and high quality of computations. The proposed scheme executes the optional part after scaling the supply voltage and processor frequency by RT-VFS. In actual systems, the configurable frequency is discrete, and hence the slack (idle) time remains even if RT-VFS scales the processor frequency. Our proposed scheme reduces the power consumption and utilizes this slack time to improve the quality of computations.

In this paper, we implement the proposed scheme on Dependable Responsive Multithreaded Processor (D-RMTP) [9]. D-RMTP System on a Chip (SoC) integrates many functions for embedded real-time systems. D-RMTP SoC supports the frequency scaling and the original features for embedded real-time systems. Moreover, D-RMTP System in a Package (SiP) supports the voltage scaling. In this way, D-RMTP SoC and SiP support RT-VFS [8]. In experimental evaluation, this paper evaluates the proposed scheme on D-RMTP.

The remainder of this paper is organized as follows. Section 2 describes the background of this paper and Section 3 shows the proposed scheme of this paper. Section 4 describes the detail of D-RMTP and an implementation of the proposed scheme for D-RMTP. In Section 5, we evaluate the proposed scheme on D-RMTP. Finally, we conclude with a summary in Section 6.

2 Background

2.1 Imprecise Computation Model

The imprecise computation model [5] decomposes each task into a mandatory part and an optional part. A mandatory part is required to complete by the deadline of each task. An optional part is executed after the mandatory part and improves the quality of computations. When a task is ready to

[1] Currently, Nintendo.
[2] Currently, Accenture.
[3] Currently, NTT Communications Corporation.
[*] Department of Information and Computer Science. E-mail: {mizotani,hatori,yusuke,takasu,chishiro,yamasaki}@ny.ics.keio.ac.jp.

execute or executing its optional part and its deadline expires, its optional part is terminated to ensure its real-time constraint. Figure 1 shows how tasks are executed on an imprecise computation model. In Figure 1, the upward arrow means the task release, the downward arrow means the deadline of the task, and an instance of the task is called job. In the second job and the third job, the optional part is terminated because its deadline expires.

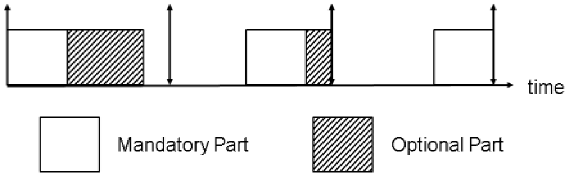


Figure 1: Example of imprecise computation model

Mandatory-First with Earliest Deadline (MFED) [3] and OPT-LU [1] are representative imprecise computation real-time scheduling algorithms. MFED is based on Earliest Deadline First (EDF) [6]. Each task is scheduled by EDF, in which a task with earliest deadline is executed preferentially. However, the priority of each task during executing an optional part is lower than that during executing a mandatory part. Therefore, MFED is an optimal real-time scheduling algorithm on a single processor as with EDF. On the other hand, OPT-LU maximizes the reward function associated with the execution of the optional part. OPT-LU is based on an optimal real-time scheduling algorithm (e.g., EDF) and prioritizes the execution of the optional part within real-time constraints of the mandatory part. However, the execution time of an optional part needs to be known for the maximization of the reward function in OPT-LU. In this paper, the execution time of an optional part is varied by RT-VFS. In addition, MFED always prioritizes the execution of a mandatory part and does not complicate the schedulability test of RT-VFS, and hence we employ MFED as the imprecise computation scheduling algorithm.

2.2 Real-Time Voltage and Frequency Scaling

In recent years, an integrated circuit such as a processor is composed of CMOS. This power consumption P_{TOTAL} is expressed in the following Equation (1):

$$P_{TOTAL} = P_{SW} + P_{LEAK}, \quad (1)$$

where P_{SW} is the switching power and P_{LEAK} is the leakage power [4]. The power consumption of the processor is composed of these powers. The switching power occurs by the switching of transistors on CMOS. This power is expressed in the following Equation (2):

$$P_{SW} = \alpha \times C \times V_{DD}^2 \times f, \quad (2)$$

where α is the activity ratio, C is the load capacity, V_{DD} is the supply voltage, and f is the processor frequency. The switching power is proportional to the square of the supply voltage. Moreover, it is proportional to the processor frequency. On the other hand, the leakage power occurs by the power supply. The leakage power is expressed in the following Equation (3):

$$P_{LEAK} = I_{LEAK} \times V_{DD}, \quad (3)$$

where I_{LEAK} is the leakage current. The leakage power is proportional to the supply voltage. Therefore, the power consumption of the processor can be reduced by scaling down the processor frequency and supply voltage on the RT-VFS scheme.

RT-VFS is classified into RT-Static VFS (RT-SVFS) and RT-Dynamic VFS (RT-DVFS), as shown in Figure 2. RT-SVFS sets the schedulable lowest processor frequency and supply voltage at a system initialization (off-line). On the other hand, RT-DVFS scales the schedulable processor frequency and supply voltage while a system run (on-line). RT-DVFS has a potential to reduce the power consumption more than RT-SVFS but RT-DVFS raises execution overhead. This is because the voltage and frequency scaling is executed at each scheduling event by utilizing the slack on RT-DVFS.

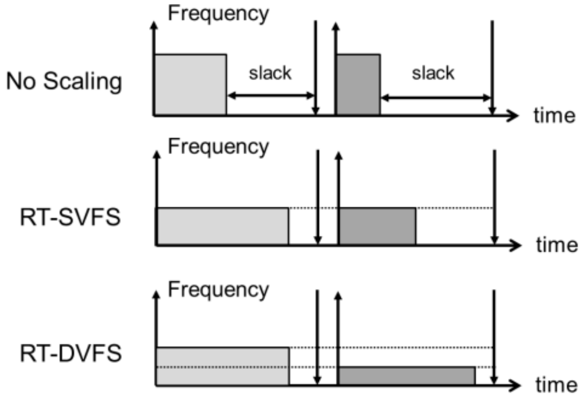


Figure 2: RT-VFS

Cycle-Conserving RT-DVFS and Look-Ahead RT-DVFS [7] are the representative RT-DVFS schemes. If the Actual Execution Time (AET) of each task is less than its Worst-Case Execution Time (WCET), these algorithms utilize the dynamic slack (i.e., WCET - AET) to scale the processor frequency. Cycle-Conserving RT-DVFS calculates the dynamic slack by recalculating the utilization using the AET consumed by tasks at each scheduling event. On the other hand, Look-Ahead RT-DVFS minimizes the current processor frequency by deferring task execution within its real-time constraint. In this paper, an optional part is executed by use of remaining slack. Therefore, Look-Ahead RT-DVFS is unsuited for the integration with an imprecise computation model. In addition, Cycle-Conserving RT-DVFS is a simple and practical algorithm, and hence we employ Cycle-Conserving RT-DVFS as the RT-DVFS scheme.

3 Integration of Imprecise Computation Model and RT-VFS

This paper proposes an integration of an imprecise computation model and RT-VFS. The proposed schemes support RT-SVFS and RT-DVFS to investigate the trade-off between quality of computations and energy consumption. In addition, the proposed model improves the quality of computations to execute optional parts in the slack time.

3.1 System Model

This paper considers a preemptive real-time system on multiprocessors or simultaneous multithreaded (SMT) processors [10]. The processor frequency $f = \{f_1, f_2, ..., f_l \mid f_1 < \cdots < f_l\}$ is discretely scaled at l frequency levels. The supply voltage V_k is scaled to the required value by f_k. In this paper, we implement the proposed scheme on the SMT processor. Each executing thread on the SMT processor is called logical processor. m logical processors $P = \{p_1, p_2, ..., p_m\}$ are supplied with the common voltage and frequency because all executing threads on the SMT processor share hardware resources such as ALU and FPU. Note that our proposed scheme can be extended easily to the independent voltage and frequency model.

Task set $\Gamma = \{\tau_1, \tau_2, ..., \tau_n\}$ is composed of n independent periodic tasks. Each task τ_i has the period T_i, the WCET of the mandatory part M_i, and the WCET of the optional part O_i. These WCETs are analyzed at the highest frequency f_l. The relative deadline D_i is equal to T_i. In addition, this paper considers partitioned scheduling that assigns tasks to processors off-line. At a system initialization, tasks are assigned to each processor by a partitioned algorithm. Each processor utilization U_j is defined as $U_j = \Sigma_{\tau_i \in p_j} M_i/T_i$. The system utilization U is defined as $U = \Sigma_j U_j$. Since an optional part is no real-time task and can be terminated, the processor utilization does not consider O_i.

3.2 Integration of MFED and RT-SVFS

First, we propose an integration of MFED and RT-SVFS, called Static MFED (S-MFED). On S-MFED, the processor frequency is scaled by RT-SVFS at a system initialization and tasks are scheduled by MFED. At a system initialization, all tasks are assigned to each processor and the supply voltage and the processor frequencies are scaled to the lowest schedulable value by EDF on the processor with highest utilization. This processor frequency f_{min} is expressed in the following Equation (4).

$$f_{min} = \min\{f_k \in f \mid \frac{f_l}{f_k} U_{max} \le 1\}, \qquad (4)$$

where $U_{max} = \max_{1 \le j \le m} U_j$.

Figure 3 shows an example of S-MFED dispatch on two processors. The example uses the task set in Table 1 which indicates the WCETs of both mandatory part and optional part, the period of each task. τ_i^M is defined as the mandatory part of task τ_i, and τ_i^O is defined as the optional part of task τ_i. The example assumes that three normalized discrete frequencies are available from (0.25, 0.50, and 1.00). It uses Worst-Fit algorithm for partitioning the example task set. In the example task set, τ_1 is assigned to p_1 and τ_2 and τ_3 are assigned to p_2 by Worst-Fit. Thus, U_1 and U_2 are 0.375, and hence the processor frequency is set to 0.50 by Equation (4). As a result, mandatory parts of all tasks are schedulable on the processor frequency 0.50. Moreover, the optional parts of τ_1 and τ_2 are executed by use of remaining idle time in Figure 3 in order to improve the quality of computations.

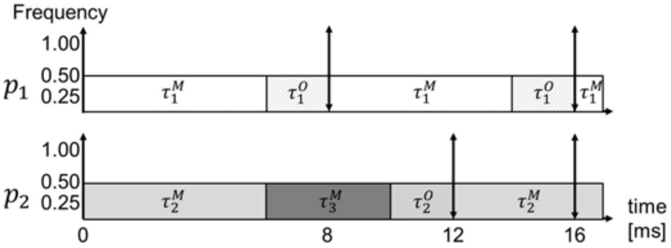


Figure 3: Example of S-MFED dispatch

Table 1: Example task set

Task	Mandatory WCET	Optional WCET	Period
1	3*ms*	3*ms*	8*ms*
2	3*ms*	2*ms*	12*ms*
3	2*ms*	2*ms*	16*ms*

The reason why we use Worst-Fit to assign tasks to logical processors is that Worst-Fit minimizes the power consumption, compared with other heuristic algorithms such as First-Fit, Next-Fit, and Best-Fit [2]. In addition, the slack time is distributed uniformly by Worst-Fit to allocate the execution time of the optional part.

Figure 4 shows an example of task execution scheduled by S-MFED on the same condition as shown in Figure 3, but using the AET from Table 2. Comparing Figure 4 with Figure 3, the execution time of each optional part is increased because the AET of each optional part is shorter than its

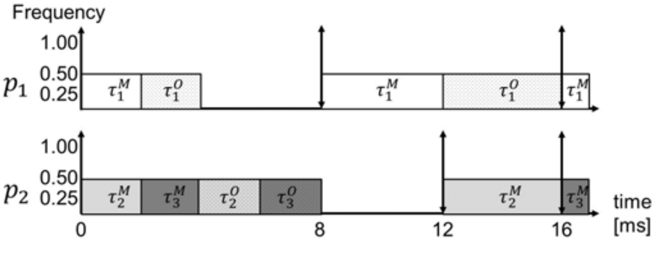


Figure 4: Example of task execution scheduled by S-MFED

WCET. In this way, there are many opportunities to execute optional parts even if the processor frequency is scaled down by RT-SVFS.

Table 2: AET of example task set

	First Job		Second Job	
Task	Mandatory	Optional	Mandatory	Optional
1	1*ms*	1*ms*	2*ms*	2*ms*
2	1*ms*	1*ms*	2*ms*	2*ms*
3	1*ms*	1*ms*	2*ms*	1*ms*

3.3 Integration of MFED and Cycle-Conserving RT-DVFS

Secondly, we propose an integration of MFED and Cycle-Conserving RT-DVFS, called Cycle-Conserving MFED (CC-MFED). On CC-MFED, tasks are scheduled by MFED and the supply voltage and processor frequency is scaled by Cycle-Conserving RT-DVFS for EDF scheduling [7]. Whenever each scheduling event occurs, it recalculates the utilization of each processor and scales the lowest schedulable frequency based on the highest utilization of all processors.

Figure 5 shows the task execution example scheduled by CC-MFED on the same condition as shown in Figure 4. The example uses the task set in Table 1 and the AET in Table 2. The remaining execution time of each task is set to WCET at each task release and measured at each task completion. The utilization of each processor is recalculated by use of this execution time. CC-MFED scales the processor frequency based on the highest processor utilization at each task release and completion. In this example, U_1 and U_2 are 0.375, and hence the selected frequency by Equation (4) is 0.50 at the system initialization. At time 2*ms*, U_1 is 0.125 and U_2 is 0.208. The highest processor utilization is U_2, and hence CC-MFED scales down the processor frequency to 0.25. At time 8*ms*, U_1 is 0.375 and U_2 is 0.146. The highest processor utilization is U_1, and hence CC-MFED scales up the processor frequency to 0.50.

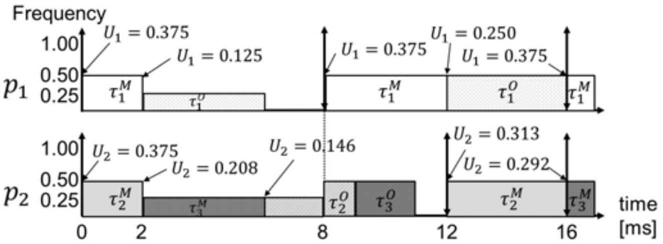


Figure 5: Example of task execution scheduled by CC-MFED

CC-MFED usually achieves lower power consumption than S-MFED but S-MFED executes the optional part of each task more than CC-MFED. This relation is the trade-off between energy consumption and quality of computations.

4 Dependable Responsive Multithreaded Processor

In this paper, we implement the proposed schemes on D-RMTP [9]. D-RMTP is one version of Responsive Multithreaded Processor [11]. This section introduces the detail of D-RMTP and presents how to use the D-RMTP original features in our implementation.

4.1 Overview of D-RMTP

D-RMTP is a prioritized SMT processor for embedded real-time systems and can execute up to 8 threads simultaneously with prioritizing each thread. Table 3 shows the outline of D-RMTP. It has 32*KBytes* of instruction and data cache (i.e., Harvard architecture) and 64*KBytes* of SRAM. All programs are executed on SRAM.

Table 3: Outline of D-RMTP

Active Threads	8
Fetch Width	8
Issue Width	4
Integer Register	32*bit* x 32*entry* x 8*set*
Floating Point Register	64*bit* x 8*entry* x 8*set*
ALU	4 + 1 (Divider)
FPU	4 + 1 (Divider)
Branch Unit	2
Memory Access Unit	1

We implement the partitioned MFED scheduler by use of the D-RMTP original feature. Threads with the same priorities are executed simultaneously on D-RMTP, and hence resource contention among these threads delays the execution of each thread.

Since a mandatory part is a real-time task and an optional part is no real-time task, a thread executing an optional part should not delay a thread executing a mandatory part simultaneously by resource contention. Thus, on our implementation, a thread executing the mandatory part is set to a high priority and a thread executing the optional part is set to a low priority. In this way, a thread executing the mandatory part can use hardware resources preferentially.

Figure 6 shows the clock tree of D-RMTP SoC. The D-RMTP SoC integrates D-RMTP, SRAM, and various I/Os.

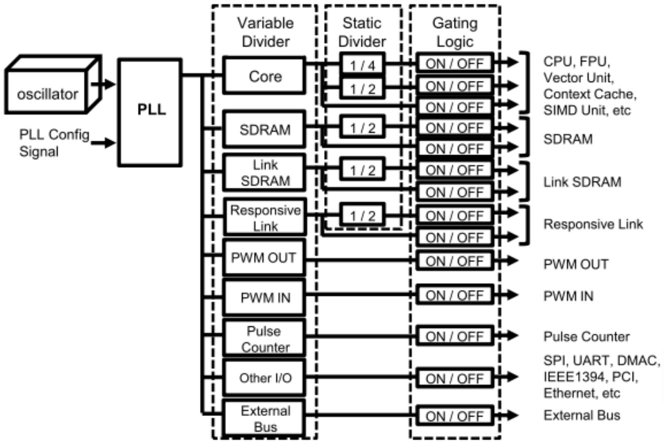


Figure 6: Clock tree of D-RMTP SoC

These modules are supplied with the clock of oscillator by the clock tree. This clock tree supports the frequency scaling by use of a variable frequency divider on each hardware module. Moreover, this clock tree supports clock gating by use of a gating logic on each hardware module.

4.2 Context Cache

In general, a context switch requires saving/restoring thread contexts to/from main memory by software instructions, which consumes a large amount of clock cycles. In order to reduce the overhead of context switch, D-RMTP has the context cache, which is an on-chip memory for thread contexts [11]. This context cache can save a maximum of 32 thread contexts.

Figure 7 illustrates the context cache implemented on D-RMTP. Bus connection between hardware contexts and the context cache is exclusive. It can transfer context data in 4 clock cycles, and hence the context switch by the context cache can reduce the overhead, compared with that by software instructions.

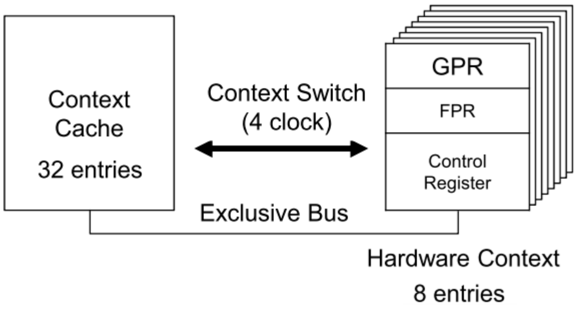


Figure 7: Context cache

When the execution of the mandatory part is completed, MFED scheduler must save the thread context for the termination of executing the optional part at any point. We use this context cache for the general scheduling context switch and save the context of the mandatory part in MFED scheduler, and hence we can reduce the scheduler overhead.

4.3 D-RMTP SiP

The D-RMTP SiP integrates the D-RMTP SoC, DDR-SDRAM modules, a power supply module, and a voltage sensor, etc. on a 30*mm* square package as shown in Figure 8.

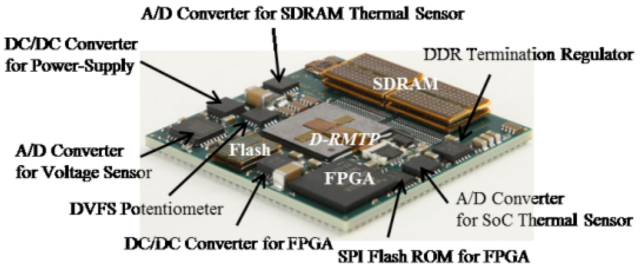

A/D Converter for SDRAM Thermal Sensor
DC/DC Converter for Power-Supply
DDR Termination Regulator
SDRAM
A/D Converter for Voltage Sensor
Flash
D-RMTP
FPGA
DVFS Potentiometer
A/D Converter for SoC Thermal Sensor
DC/DC Converter for FPGA
SPI Flash ROM for FPGA

Figure 8: D-RMTP SiP

The D-RMTP SiP supports the voltage scaling by a DC-DC converter and a potentiometer. These devices can scale the processor supply voltage within the range of [0.8, 1.1]*V* by a serial peripheral interface [8]. We implement RT-VFS by use of the frequency scaling in Subsection 4.1 and this voltage scaling.

5 Experimental Evaluation

5.1 Evaluation Environment

In experimental evaluation, we use a D-RMTP evaluation kit which has the D-RMTP SiP, an oscillator, I/O pins, and FPGA, etc. as shown in Figure 9. Also, Figure 10 shows measurement environment. The energy consumption is precisely measured by the synchronization of an oscilloscope and a logic analyzer. The sampling rate is 100*K* samples per second and the measurement time is 1,024*ms*.

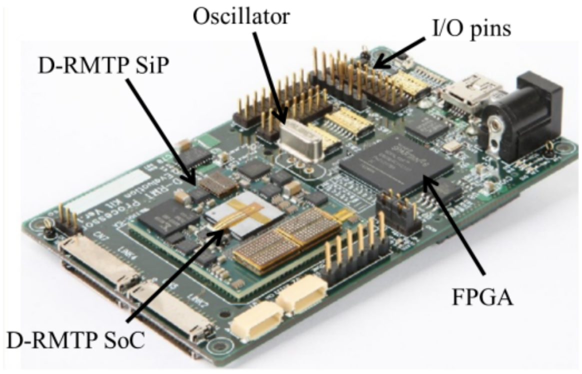


Figure 9: D-RMTP evaluation kit

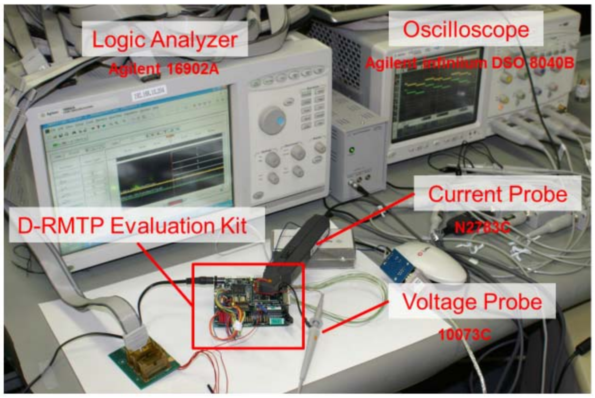


Figure 10: Measurement environment

We evaluate S-MFED, CC-MFED, and only-MFED on a single logical processor and two logical processors. Moreover, we measure the energy consumption of EDF on RT-SVFS (S-EDF) and Cycle-Conserving RT-DVFS (CC-EDF). By comparing MFED with EDF on the same RT-VFS scheme, we analyze the influence of executing optional parts in the energy consumption. Table 4 shows the combination of the scaled processor frequency and supply voltage by RT-VFS on D-RMTP. In addition, this experimental evaluation applies clock gating to unused modules in Figure 6, specifically, DMA Controller, PCI, IEEE1394, Vector Unit, PWM OUT, PWM IN, Responsive Link, Link SDRAM I/F, and Ethernet. The energy consumption is additionally reduced by the clock gating to unused modules.

We generate random task sets on uniformed distribution for evaluation. Each task set is composed of the periodic task to multiply square matrices. The range of the system utilization U is [0.1, 0.9] at intervals of 0.1 on a single logical processor and [0.2, 1.8] at intervals of 0.2 on two logical processors. The range of the number of tasks n is [4, 10]. All task sets are harmonic; periods of tasks are integer multiples of each other. The period of each task is within the range of [4, 64]ms.

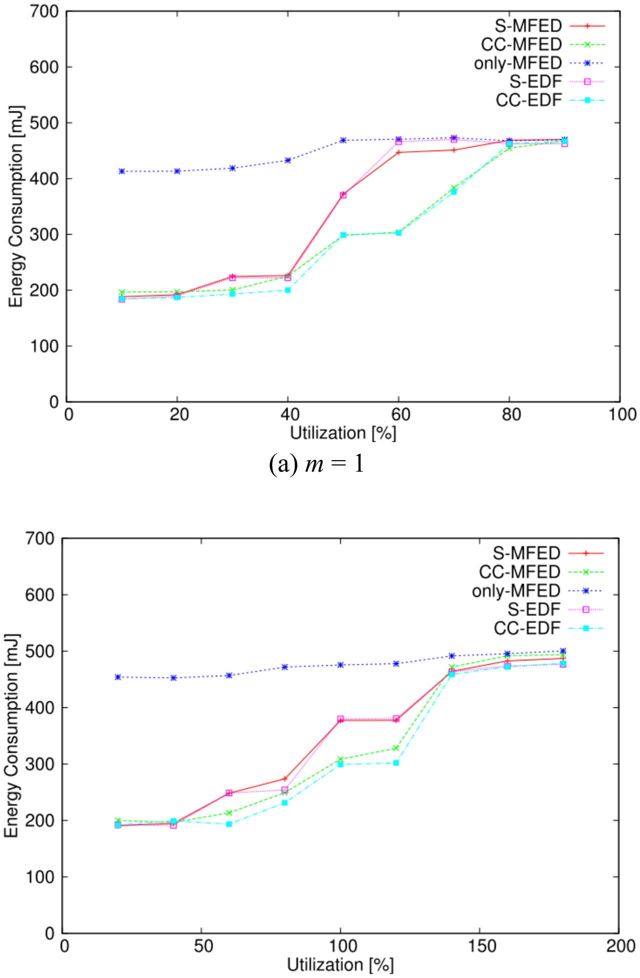
Both the WCETs of the mandatory and optional parts in all tasks are $1ms$. Each evaluation result is the arithmetic mean of five task sets. Also, the processor frequency is scaled uniformly, and hence it is desirable that the partitioned algorithm assigns tasks to each logical processor as uniformly as possible for reducing the energy consumption. Therefore, we employ Worst-Fit Decreasing Utilization (WFDU) algorithm for assigning tasks to logical processors. WFDU assigns tasks to logical processors by Worst-Fit after tasks are sorted in decreasing order of the processor utilization. In addition, the slack time is distributed uniformly to execute optional parts.

5.2 Energy Consumption

Figure 11 shows the energy consumption of the processor on a single logical processor and two logical processors. The difference between the trends on a single logical processor and on two logical processors is small. S-MFED achieves an average of 26% and a maximum of 57% reduction in the energy consumption against only-MFED. On the other hand, CC-MFED achieves an average of 29% and a maximum of 58% reduction in the energy consumption against only-MFED. CC-MFED achieves lower energy consumption than S-MFED within the system utilization 30% to 70% on a single logical processor and 60% to 120% on two logical processors. This reason is that CC-MFED gains many opportunities to scale down the processor frequency by measuring the AET of each task. However, CC-MFED scales down the processor frequency and supply voltage on-line but the result of this scaling is approximately the same as that of S-MFED in the other range of system utilization. This is because CC-MFED has higher scheduling overhead than S-MFED due to RT-DVFS. On the other hand, the difference between S-MFED and S-EDF is a maximum of $19mJ$. Also, the difference between CC-MFED and CC-EDF is a maximum of $27mJ$. Thus, the execution of optional parts has a small influence on the energy consumption, and hence executing optional parts is energy effective on D-RMTP.

5.3 Quality of Computations

Figure 12 shows the quality of computations on a single logical processor and two logical processors. This quality is defined as the ratio of the executed optional part, called *optional part execution ratio*. Each evaluation result is the arithmetic mean quality of each job in a task set. The difference between the trends on a single logical processor and on two logical processors is also small as that in energy consumption. This shows that the execution on two logical processors does not often cause the resource contention. S-MFED and CC-MFED execute less optional parts than only-MFED because of scaling down the processor frequency. Especially, the difference between CC-MFED and only-MFED is an average of 38% and a maximum of 95%. CC-MFED prioritizes scaling down the processor frequency, and hence the optional part execution ratio is reduced to a greater degree.

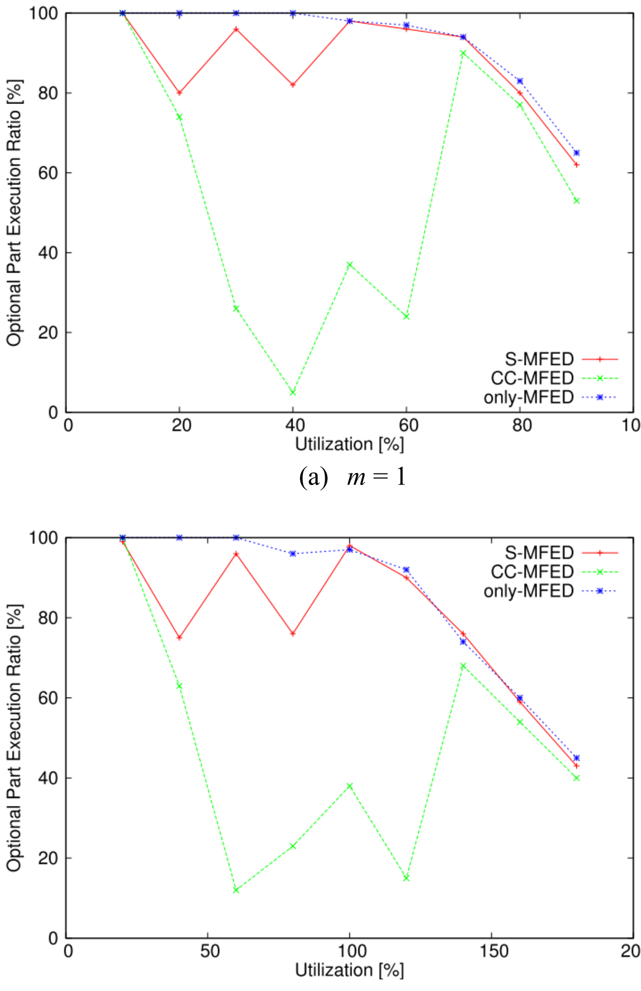


(a) $m = 1$

(b) $m = 2$

Figure 11: Energy consumption when $m = 1$ and $m = 2$

On the other hand, the difference between S-MFED and only-MFED is an average of 6% and a maximum of 25%. The optional part execution ratio on S-MFED is not reduced much. This reason is that the AET of each task was shorter than its WCET, and hence the optional part of each task could be executed for more time. In addition, S-MFED and CC-MFED are not monotonically decreased when the system utilization becomes high because of scaling the processor frequency discretely.



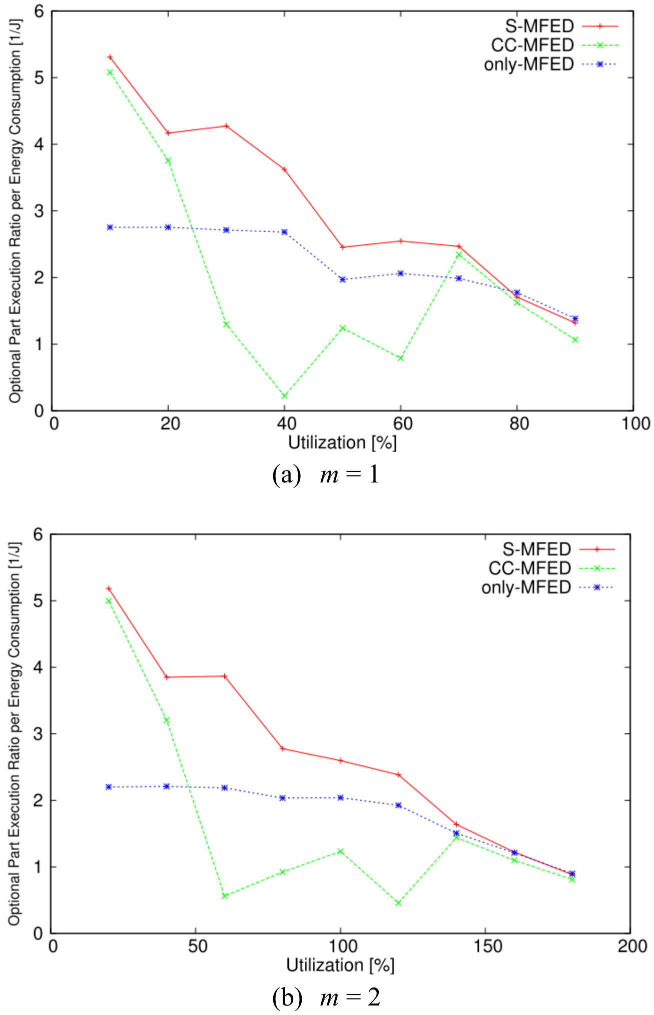
(a) $m = 1$

(b) $m = 2$

Figure 12: Quality of computations when m = 1 and m = 2

5.4 Trade-off between Energy Consumption and Quality of Computations

As discussed in Section 3, there is the trade-off between the energy consumption and the quality of computations. We calculate the optional part execution ratio per energy consumption by use of evaluation results in Subsection 5.2 and Subsection 5.3 for analyzing this trade-off. Figure 13 shows the optional part execution ratio per energy consumption on a single logical processor and two logical processors. S-MFED and CC-MFED achieve maximums of 135% and 127% improvement against only-MFED, respectively. In particular, the proposed schemes achieve higher improvement than only-MFED on the low system utilization. The reason is that there is sufficient slack to reduce the energy consumption by RT-VFS and improve the quality of computations by executing an optional part. S-MFED always achieves higher optional part execution ratio per energy consumption than only-MFED. Therefore, S-MFED can achieve both lower energy consumption and higher performance. In particular, S-MFED can achieve an average of 38 % improvement of the optional part execution ratio per energy consumption against only-MFED.



(a) $m = 1$

(b) $m = 2$

Figure 13: Relation between energy consumption and quality of computations when m = 1 and m = 2

On the other hand, CC-MFED has lower optional part execution ratio per energy consumption than only-MFED when the system utilization is higher than or equal to 30% on a single logical processor and 60% on two logical processors. Therefore, CC-MFED results in an average of 13 % reduction of the optional part execution ratio per energy consumption against only-MFED. However, as discussed in Subsection 5.2, CC-MFED achieves the lower energy consumption within the system utilization 30% to 70% on a single logical processor and 60% to 120% on two logical processors than S-MFED. When the energy consumption is regarded as important in embedded real-time systems, CC-MFED can satisfy the requirement. Moreover, when WCET is very longer than

AET, CC-MFED can utilize enough dynamic slack time and achieve more improvement of the optional part execution ratio per energy consumption. In this experimental evaluation, generated tasks execute very simple calculations and WCET is close to AET. However, tasks execute more complex processing and WCET is much longer than AET in the actual systems. In such actual systems, CC-MFED may be the effective scheme.

6 Conclusion

This paper proposed S-MFED and CC-MFED as an integration of an imprecise computation model and RT-VFS. These proposed schemes satisfy both requirements: reduction of the energy consumption and improvement of the quality of computations within real-time constraints. Moreover, we implemented these schemes on D-RMTP by use of D-RMTP original features to improve the quality of computations and reduce the energy consumption. Through experimental evaluation, we showed the proposed schemes satisfying both the lower energy consumption and higher quality of computations on actual systems. In our evaluation, the proposed schemes achieved a maximum of 135% improvement of the optional part execution ratio per energy consumption.

In future work, we plan to evaluate the proposed schemes over two logical processors. Moreover, we plan to extend CC-MFED to be aware of the optional part execution ratio of each task.

Acknowledgement

This research was supported in part by Keio Gijuku Academic Development Funds, Keio Kougakukai, and CREST, JST.

References

[1] H. Aydin, R. Melhem, D. Mosse, and P. Mejfa-Alvarewz. “Optimal Reward-Based Scheduling for Periodic Real-Time Tasks,” *IEEE Transactions on Computers*, 50(2):111-130, February 2001.

[2] H. Aydin and Q. Yang, “Energy-Aware Partitioning for Multiprocessor Real-Time Systems,” *Proceedings of the 17th International Parallel and Distributed Processing Symposium*, pp. 113-121, April 2003.

[3] S. K. Baruah and M. E. Hickey, “Competitive On-line Scheduling of Imprecise Computations,” *IEEE Transactions on Computers*, 47(9):1027-1032, September 1998.

[4] T. D. Burd and R. W. Brodersen, “Energy Efficient CMOS Microprocessor Design,” *Proceedings of the 28th Hawaii International Conference on System Sciences*, pp. 288-297, January 1995.

[5] K. J. Lin, S. Natarajan, and J. W. Liu, “Imprecise Results: Utilizing Partial Computations in Real-Time Systems,” *Proceedings of the 8th IEEE Real-Time Systems Symposium*, pp. 210-217, December 1987.

[6] C. L. Liu and J. W. Layland, “Scheduling Algorithms for Multiprogramming in a Hard Real-Time Environment,” *Journal of the ACM*, 20(1):46-61, January 1973.

[7] P. Pillai and K. G. Shin, “Real-Time Dynamic Voltage Scaling for Low-Power Embedded Operating Systems,” *Proceedings of the 18th ACM Symposium on Operating Systems Principles*, pp. 89-102, October 2001.

[8] K. Suito, M. Takasu, R. Ueda, K. Fujii, H. Matsutani, and N. Yamasaki, “Experimental Evaluation of Low Power Techniques on Dependable Responsive Multithreaded Processor,” *Proceedings of the 28th ISCA International Conference on Computers and Their Applications*, pp. 281-288, March 2013.

[9] K. Suito, R. Ueda, K. Fujii, T. Kogo, H. Matsutani, and N. Yamasaki, “The Dependable Responsive Multithreaded Processor for Distributed Real-Time Systems,” *IEEE Micro*, 32(6):52-61, December 2012.

[10] D. M. Tullsen, S. J. Eggers, and H. M. Levy, “Simultaneous Multithreading: Maximizing On-Chip Parallelism,” *Proceedings of the 22nd Annual International Symposium on Computer Architecture*, pp. 392-403, May 1995.

[11] N. Yamasaki, “Responsive Multithreaded Processor for Distributed Real-Time Systems,” *Journal of Robotics and Mechatronics*, 17(2):130-141, April 2005.

Keigo Mizotani works at Nintendo. He received his B.S. and M.S. degrees from Keio University in 2013 and 2015, respectively. His research interests include embedded real-time systems, power management, and imprecise computation models.

Yusuke Hatori is a master course student in Graduate School of Science and Technology at Keio University. He received his B.S. degree from Keio University in 2014. His research interests include real-time systems and operating systems.

Yusuke Kumura works at Accenture. He received his B.S. and M.S. degrees from Keio University in 2013 and 2015, respectively. His research interests include real-time systems and operating systems.

Masayoshi Takasu works at NTT Communications Corporation. He received his B.S. and M.S. degrees from Keio University in 2013 and 2015, respectively. His research interests include real-time systems and low energy consumption.

Hiroyuki Chishiro is a research associate in the Department of Information and Computer Science at Keio University. He received his B.S., M.S., and Ph.D. degrees from Keio University in 2008, 2010, and 2012, respectively. His research interests include real-time systems, operating systems, middleware, and trading systems.

Nobuyuki Yamasaki is a professor in the Department of Information and Computer Science at Keio University. He is also a guest researcher in the Digital Human Research Center at Advanced Industrial Science and Technology. He received his B.S., M.S., and Ph.D. degrees from Keio University in 1991, 1993, and 1996, respectively. His research interests include parallel and distributed systems, real-time systems, system LSI, and robotics.

Improving Fuzzy C-Means Algorithm for Magnetic Resonance Images (MRIs) Segmentation

Khalid A. Buragga*[†]
Northern Border University, SAUDI ARABIA

Sultan Aljahdali*[‡]
Taif University, SAUDI ARABIA

E. A. Zanaty*[§]
Taif University, SAUDI ARABIA

Abstract

Image segmentation is one of the most important parts of clinical diagnostic tools. Medical images mostly contain noise and inhomogeneity. Especially, accurate segmentation of magnetic resonance imaging (MRI) is a very difficult task. However, the process of accurate segmentation of these images is very important and crucial for a correct diagnosis by clinical tools. In this paper, we present fuzzy c-means (FCM) method based on Gaussian function for improving MRI segmentation. The proposed algorithm is realized by modifying the objective function of the conventional FCM algorithm with a Gaussian function and to allow the labeling of a pixel to be influenced by its neighbors in the image. The proposed algorithm is fed by the initial centers for the objective function as a prior knowledge to avoid the coincident clusters. Then, the process of finding the best clusters are continue to update the centers and the membership and only stop when the factor between two successive centers is smaller than a prescribed value. The proposed algorithm is applied to magnetic resonance image (MRI) datasets. Compared with the existing approaches, the proposed method can achieve the most accurate results.

Key Words: Fuzzy clustering, modified fuzzy c-means, medical image segmentation.

1 Introduction

With increasing use of magnetic resonance imaging (MRI) for diagnosis, treatment planning and clinical studies, it has become almost compulsory to use computers to assist radiological experts in clinical diagnosis and treatment planning. Medical images tend to suffer much more noise than realistic images due to the nature of the acquisition devices. This of course poses great challenges to any image segmentation technique. In order to reduce noise, many devices increase the partial voluming, that is, they average acquisition on a thick slice. This leads to blurring the edges between the objects, which make decisions very hard for automatic tools [2]. The different acquisition modalities, the different image manipulations and variability of organs all contribute to a large verity of medical images. It can be safely said that there is no single image segmentation method that suits all possible images. This can pose great problems for any segmentation method. Therefore, several types of image segmentation techniques [1, 4-7, 15, 18-19] were found to achieve accurate segmentations. Among them, fuzzy segmentation methods have considerable benefits, because they could retain much more information from the original image than hard segmentation methods [19]. In particular, the fuzzy c-means (FCM) algorithm assigns pixels to fuzzy clusters without labels. Since the conventional FCM algorithms classify pixels in the feature space without considering their spatial distribution in the image, they are highly sensitive to noise and other imaging artifacts. Many extensions of the FCM algorithm have been proposed to overcome the above mentioned problem and reduce errors in the segmentation process [1, 5, 7, 15, 18, 23-24, 28]. Among them, algorithms based on modified FCM objective function is widely used in medical image clustering to suppress the noise effect during the segmentation processes. Modified FCM objective function is to add penalty term into the objective function to constrain the membership values. Based on the traditional FCM objective function, most improved approaches embodied regularization terms to show the increased robustness of the classification of the noisy images. Pham and Prince [17] modified the FCM objective function by introducing a spatial penalty for enabling the iterative algorithm to estimate spatially smooth membership functions. Ahmed et al. [1] introduced a neighborhood averaging additive term into the objective function of FCM. They named the algorithm bias corrected FCM (BCFCM). Liew and Yan [14] introduced a spatial constraint to a fuzzy cluster method where the inhomogeneity field was modeled by a b-spline surface. The spatial voxel connectivity was implemented by a dissimilarity index, which enforced the connectivity constraint only in the homogeneous areas. This way preserves significantly the tissue boundaries. Szilágyi et al. [20] modified the FGFCM (MFGFCM) to improve the precision of segmentation. They proposed EnFCM algorithm to accelerate

[*] Dept. of Computer Science, College of Computing and IT.
[†] E-mail: nbu10@nbu.edu.sa.
[‡] E-mail: aljahdali@tu.edu.sa.
[§] E-mail: zanaty22@yahoo.com.

the image segmentation process. EnFCM is based on a simple fact about images, which is usually overlooked in many FCM-type algorithms. Cai et al. [6] introduced a new local similarity measure by combining spatial and gray level distances. They used their method as an alternative pre-filtering to EnFCM. They named this approach fast generalized FCM (FGFCM). This method is able to extract local information that causes less blur than averaging filter. However, it still has an experimentally adjusted parameter and the precision of the segmentation is not good enough. Kang et al. [12] improved FCM with adaptive weighted averaging filter (FCM AWA). Kang et al. [11] proposed a spatial homogeneity-based FCM (SHFCM). Wang et al. [22] incorporated both the local spatial context and the non-local information into the standard FCM cluster algorithm. They used a novel dissimilarity measure in place of the usual distance metric. These approaches could overcome the noise impact, but the intensity homogeneity cannot be handled at the same time. FCM-based algorithms are known to be vulnerable to outliers and noise. To address this problem, possibilistic clustering which is pioneered by the possibilistic c-means (PFCM) algorithm [13] is developed. It has shown to be more robust to outliers than FCM. However, the robustness of PFCM comes at the expense of the stability of the algorithm [3]. The PCM-based algorithms suffer from the coincident cluster problem, which makes them too sensitive to initialization [3]. Many efforts have been presented to improve the stability of possibilistic clustering [10, 21, 29]. However, PFCM estimates the centroids robustly in the case of outliers.

Although suppressing the impact of noise and intensity inhomogeneity to some extent, these algorithms still produce misclassified small regions [25-27, 30]. They still depend on a fixed spatial parameter which needs to be adjusted. Furthermore, the cost of estimating the neighbors for each point in an image is still high. Therefore, these drawbacks will reduce the clustering performance in real applications.

This paper addresses these problems for overcoming the shortcomings of existing fuzzy methods. In order to reduce the noise effect during segmentation, a new fuzzy c-means algorithm based on Gaussian function is presented that could improve the medical image segmentation. The proposed algorithm is realized by modifying the objective function of the conventional FCM algorithm with a Gaussian function and to allow the labeling of a pixel to be influenced by its neighbors in the image. This function is fed by initial centers for the generation of fuzzy terms.

The efficiency of the proposed algorithm is demonstrated by extensive segmentation experiments using real MR images and by comparison with other state-of –the-art algorithms.

The rest of this paper is organized as follows. In Section 2, the background of FCM is discussed. In Section 3, we propose a FCM algorithm. Experimental comparisons are given in Section 4. Finally, Section 5 gives our conclusions.

2 Background

The FCM algorithm [4] is an iterative clustering method that produces optimal C partitions by minimizing the weighted objective function J_{FCM}:

$$J_{FCM}(V,U,X)=\sum_{i=1}^{C}\sum_{j=1}^{n}u_{ij}^{m}d^{2}(x_j,v_i),1<m<+\infty \tag{1}$$

Where $X=\{x_1,x_2,...,x_n\}\subseteq R^p$ is the data set in the p-dimensional vector space, n is the number of points, p is the number of data items, C is the number of clusters with $2<C<n\text{-}1$. $V=\{v_1, v_2, . . . , v_C\}$ is the C centers or prototypes of the clusters, v_i is the p-dimension center of the cluster i, and $d^2(x_j,v_i)$ is a square distance measure between object x_j and cluster center $v_i, U=\{u_{ij}\}$ represents a fuzzy partition matrix with $u_{ij}=u_i(x_j)$ is the degree of membership of x_j in the i^{th} cluster; x_j is the j^{th} of p-dimensional measured data. The fuzzy partition matrix satisfies:

$$0<\sum_{j=1}^{n}u_{ij}<n,\forall i\in\{1,......,C\} \tag{2}$$

$$\sum_{i=1}^{c}u_{ij}=1,\forall j\in\{1,.....,n\} \tag{3}$$

The parameter m is a weighting exponent for each fuzzy membership and determines the amount of fuzziness of the resulting classification; it is a fixed number greater than one. The objective function J_{FCM}can be minimized under the constraint of U.

The objective function J_{FCM} is minimized with respect to u_{ij} and v_i , respectively:

$$u_{ij}=\left[\sum_{k=1}^{C}\left(\frac{d^2(x_j,v_i)}{d^2(x_j,v_k)}\right)^{2/(m-1)}\right]^{-1} \quad 1\le i\le C, i\le j\le n. \tag{4}$$

$$v_i=\frac{\sum_{k=1}^{n}u_{ik}^{m}x_k}{\sum_{k=1}^{n}u_{ik}^{m}},1\le i\le C. \tag{5}$$

Although FCM and the modified FCM [22-23, 30] are useful clustering methods, their memberships do not always correspond well to the degree of belonging of the data, and may be inaccurate in a noisy environment, because the real data unavoidably involves noise. In order to alleviate weakness of FCM and to produce memberships that have a good explanation for the degree of belonging of the data, Wang *et al.* [22] relaxed the constrained condition (3) of the fuzzy *C*-partition to obtain a possibilistic type of membership function and propose PCM for unsupervised clustering. The component generated by the PCM corresponds to a dense region in the data set; each cluster is independent of the other clusters in the PCM strategy. The objective function of the PCM can be formulated as follows:

$$J_{PCM}(V,U,X)=\sum_{i=1}^{C}\sum_{j=1}^{n}u_{ij}^{m}d^{2}(x_j,v_i)+\sum_{i=1}^{C}\eta_i\sum_{j=1}^{n}(1-u_{ij})^{m} \tag{6}$$

Where

$$\eta_i=\frac{\sum_{j=1}^{n}u_{ij}^{m}d^{2}(x_j,v_i)}{\sum_{j=1}^{n}u_{ij}^{m}} \tag{7}$$

is the scale parameter at the i^{th} cluster, and

$$u_{ij}=\frac{1}{1+\left(\frac{d^{2}(x_j,v_i)}{\eta_i}\right)^{\frac{1}{m-1}}} \tag{8}$$

is the possibilistic typicality value of training sample x_j belonging to the cluster $i, m\in\,]1,\infty[$ is a weighting factor called the possibilistic parameter.

Zanaty and Sultan [28] proposed a method for automatic fuzzy algorithms by considering some spatial constraints on the objective function. The algorithm starts to subdivide the data from a set of N vector $X=\{x_j,\, j=1,\ldots,N\}$ into *M* clusters using a well-known fuzzy method [24] (see Equations (1-4)). Assume the data is divided into *M* cluster, $R_1,R_2,..,R_M$ with centres $v_1,v_2,..,v_M$, respectively. The proposed algorithm processes every two neighbours clusters individually, i.e., if we have three clusters A,B,C with centres c_A,c_B, and c_c. We start to hold our validity function between clusters A and B if:

$$\|v_A-v_B\|<\|v_A-v_c\|$$

Our validity function is proposed to use the intra-cluster distance measure, which is simply the distance between a centre of cluster A and cluster centre B multiplied by the objective function of fuzzy. We can define the validity function as:

$$V_1=\|v_A-v_B\|^{2}\frac{1}{n}\sum_{i=1}^{n}u_{i}^{m}(d_{iA}+d_{iB})$$
$$V_2=(Max(A)-Min(B))\sum_{i=1}^{n}\sum_{k=1}^{2}u_{ik}^{m}d_{iA\cup B} \tag{9}$$

Where $Max(B)$ and $Min(B)$ are the maximum and minimum values of clusters A and B, respectively. While $d_{A\cup B}$ is the distance of the data x_i (of number *n*) of A union B i.e., $A\cup B$. This algorithm works iteratively and the number of clusters increases automatically according to the decision of validity function in Equations (9). More discussion can be shown in [28].

3 The Proposed Algorithm

The choice of an appropriate objective function is a key to the success of cluster analysis and to obtain better quality clustering results; hence, clustering optimization is based on the objective function [29]. To identify a suitable objective function, one may start from the following set of requirements: the distance between the data points assigned to a cluster should be minimized and the distance between clusters should be maximized [10]. To obtain an appropriate objective function, we take into consideration the following:

- The distance between clusters and the data points allocated to them must be reduced.
- Coincident clusters may occur and must be controlled.
- Selecting the initialization sensitive parameters for decreasing noise affect.

To overcome the limitation of the fuzzy methods, we present a novel fuzzy c-means algorithm based on gaussian function. As for the common value used for this parameter by every data for iterations, we propose a new weight function which is based on Gaussian membership of a point p achieving every point of the data set that has a weight in relation to every cluster. The usage of weights produces good classification particularly in the case of noisy data.

The proposed algorithm starts by partitioning the image into *C* regions of intensity by the known minimum and maximum values of intensity using a well-known histogram algorithm [8]. The median point of each region R_k (including points $x_i, i=1,2,..N_k$, N_k is the number of points of R_k) is selected to be as initial centers of the region, and then both region and centers are fed to the method. While the constraints term $\sum_{k=1}^{C}\sum_{i=1}^{N}\alpha u_{ki}^{m}e^{-\eta(x_i-c_k)^2}$ is only considered in the objective function if a point x_i belongs to a region R_k, *k*=1,2,…, *C* with initial center c_k which takes $\hat{c}_k$ for the first iterative i.e., we only estimate this term if $x_i\in R_k$. The cost of computations can be reduced using a region and not neighbors for all points.

The objective function of the proposed modified fuzzy c-means is modified to:

$$J_m = \sum_{k=1}^{C}\sum_{i=1}^{N} u_{ki}^{m}\left\|x_i - c_k\right\|^2 + \sum_{k=1}^{C}\sum_{i=1}^{N} \alpha u_{ki}^{m} e^{-\eta(x_i - c_k)^2} \tag{1}$$

where

$u = (u_{ki})_{C \times N}, C = (c_1, c_2, .., c_k), \eta = 1/2\sigma^2$,
σ is the standard deviation and $\alpha = 1/\sigma\sqrt{2\pi}$

is a constant.

However, the crucial parameter is *m*, which represents the power of the memberships. More datasets are experimented in [9, 16], and they proved that there is a relation between data shape and *m*. For instance, the triangular shape will fit better if *m*=3 is used, and more discussion can be shown in [9]. Therefore, we take into account the data shape in the objective function, which is to be general for all tested data sets. This penalty term also contains preprocessing centers information, which acts as a regularizer and biases the solution toward piecewise-homogeneous labeling. Such regularization is helpful in segmenting images corrupted by noise. The objective function J_m under the constraint of u_{ki} and c_k can be solved by using the following theorem [1]:

Theorem: Let $X = \{x_i, i = 1,2,..., N \mid x_i \in R^d\}$ denotes an image with N pixels to be partitioned into C classes (clusters), where x_i represents feature data. The algorithm is an iterative optimization that minimizes the objective function defined by Equation (1). Then u_{ki} and c_i must satisfy the following equalities:

$$u_{ki} = \frac{1}{\sum_{L=1}^{C}\left[\frac{\left\|x_i - c_L\right\|^2 + \alpha e^{-\eta(x_i - c_L)^2}}{\left\|x_i - c_k\right\|^2 + \alpha e^{-\eta(x_i - c_k)^2}}\right]^{\frac{1}{m-1}}} \tag{10}$$

$$c_k = \frac{\sum_{i=1}^{N} x_i u_{ki}^{m}\,(1 + e^{-\eta(x_i - \hat{c}_k)^2})}{\left[\sum_{i=1}^{N} u_{ki}^{m}\left(1 + e^{-\eta(x_i - \hat{c}_k)^2}\right)\right]} \tag{11}$$

Proof: We minimize the following equation using the Lagrange method:

$$J_m = \sum_{k=1}^{C}\sum_{i=1}^{N} u_{ki}^{m}\left\|x_i - c_k\right\|^2 + \alpha \sum_{k=1}^{C}\sum_{i=1}^{N} u_{ki}^{m} e^{-\eta(x_i - c_k)^2} \tag{12}$$

where

$u = (u_{ki})_{C \times N}, C = (c_1, c_2, .., c_k), \eta = 1/2\sigma^2$ and $\alpha = 1/\sqrt{2\pi}$

is a constant.

For that, Equation (12) can be rewritten as:

$$L_m = \sum_{k=1}^{C}\sum_{i=1}^{N} u_{ki}^{m}\left\|x_i - c_k\right\|^2 + \tag{13}$$

$$\frac{\partial L_m}{\partial u_{ki}} = 0 \Rightarrow \sum_{k=1}^{C}\sum_{i=1}^{N} m u_{ki}^{m-1}\left\|x_i - c_k\right\|^2 +$$

$$\frac{\partial L_m}{\partial \lambda_i} = 0 \Rightarrow \sum_{k=1}^{C} u_{ki} - 1 = 0 \tag{14}$$

$$u_{ki} = \left[\frac{\lambda_i}{m\left(\left\|x_i - c_k\right\|^2 + \alpha e^{-\eta(x_i - c_k)^2}\right)}\right]^{\frac{1}{m-1}}$$

$$u_{ki} = \left[\frac{\lambda_i}{m}\right]^{\frac{1}{m-1}}\left[\frac{1}{\left\|x_i - c_k\right\|^2 + \alpha e^{-\eta(x_i - c_k)^2}}\right]^{\frac{1}{m-1}}$$

$$\left[\frac{\lambda_i}{m}\right]^{\frac{1}{m-1}}\sum_{i=1}^{C}\left[\frac{1}{\left\|x_i - c_k\right\|^2 + \alpha e^{-\eta(x_i - c_k)^2}}\right]^{\frac{1}{m-1}} = 1$$

$$\sum_{k=1}^{c} u_{ki} = 1$$

$$u_{ki} = \sum_{L=1}^{C}\frac{(\left\|x_i - c_L\right\|^2 + \alpha e^{-\eta(x_i - c_L)^2})^{\frac{1}{m-1}}}{(\left\|x_i - c_k\right\|^2 + \alpha e^{-\eta(x_i - c_k)^{2})})^{\frac{1}{m-1}}}$$

$$u_{ki} = \frac{1}{\sum_{L=1}^{C}\left[\frac{\left\|x_i - c_L\right\|^2 + \alpha e^{-\eta(x_i - c_L)^2}}{\left\|x_i - c_k\right\|^2 + \alpha e^{-\eta(x_i - c_k)^2}}\right]^{\frac{1}{m-1}}}$$

$$\frac{\partial L_m}{\partial c_k} = 0 \tag{15}$$

$$-2\sum_{i=1}^{m} u_{ki}^{m}(x_i - c_k) - 2\alpha(x_i - c_k)\sum_{i=1}^{N} u_{ki}^{m} e^{-\eta(x_i - c_k)^2} = 0$$

$$\sum_{i=1}^{N} u_{ki}^{m} x_i - \sum_{i=1}^{N} u_{ko}^{m} c_k + \sum_{i=1}^{N} x_i u_{ki}^{m} e^{-\eta(x_i - c_k)^2} - \sum_{i=1}^{N} c_k u_{ki}^{m} e^{-\eta(x_i - c_k)^2} = 0$$

$$c_k \left[\sum_{i=1}^{N} u_{ki}^{m} \left(1 + e^{-\eta(x_i - c_k)^2} \right) = \sum_{i=1}^{N} x_i u_{ki}^{m} \left(1 + e^{-\eta(x_i - c_k)^2} \right) \right]$$

We assume that c_k of the iterative process is $\hat{c}_k$ and then $\hat{c}_k = c_{k-1}$ for the further iteration. Then we can rewrite the $\hat{c}_k$ as follows:

$$c_k = \frac{\sum_{i=1}^{N} x_i u_{ki}^{m} \left(1 + e^{-\eta(x_i - \hat{c}_k)^2}\right)}{\left[\sum_{i=1}^{N} u_{ki}^{m} \left(1 + e^{-\eta(x_i - \hat{c}_k)^2}\right)\right]} \tag{16}$$

The process of finding the best clusters are continued to update the centres c_k and the membership u_{ki} using Equations (2) and (3), respectively. The R_k neighbors of the centres $\hat{c}_k$ can be obtained using histogram preprocessing [8].

Algorithm:

- **Initialize**: the membership matrix u_{ki}^{t} with random values between 0 and 1 such that the constraints in Equation (1) are satisfied.
- **Input:** initial centres $\hat{c}_k$, i=1,…, C, the data $p_i, i = 1..N$

Repeat:

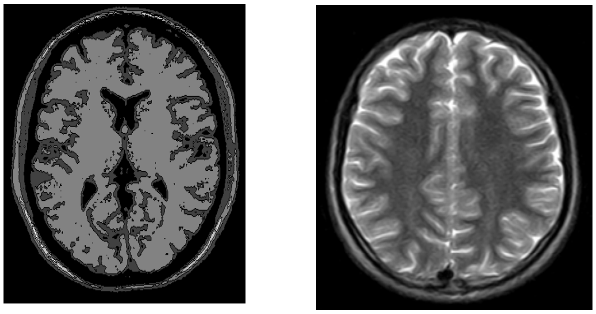
- **Compute**: u_{ki} and c_k using Equations (10) and (11).
- **Until:** $\|u_{ki} - u^{t}{}_{ki}\| \leq \varepsilon$, where ε a certain tolerance value
- **End Repeat**

4 Experimental and Comparative Results

The experiments were performed on two different sets: one corrupted by (0%, 3%, 5%, 7%, 9%) salt and pepper noise and the image size is 129×129 pixels which are shown in Figure 1a, and Figure 1b, respectively [5]. The advantages of using digital phantoms rather than real image data for soft segmentation methods include prior knowledge of the true tissue types and control over image parameters such as modality, slice thickness, noise, and intensity in homogeneities. The quality of the segmentation algorithm is of vital importance to the segmentation process. The comparison score AOM for each algorithm as proposed in [15] is defined as follows:

$$AOM = \left| \frac{A \cap A_{ref}}{A \cup A_{ref}} \right|$$

Where A represents the set of pixels belonging to a class as found by a particular method and A_{ref} represents the reference cluster pixels.



(a) (b)

Figure 1: Test images: (a) 3D simulated data, and (b) two original slices from the 3D simulated data (slice 89 and slice 65)

4.1 Experiment in the Real Image

We used a high-resolution T1-weighted MR phantom with slice thickness of 1mm obtained from the classical simulated brain database of McGill University [5]. We added various noise levels (0%, 3%, 5%, 7%, 9%) and RF levels 20% to the test image. Tthe proposed method is applied to the noise images to prove its efficiency. The segmentation results are shown through Figures 2 to 11 while AOM of WM of every image is computed and then we present AOM results in Table 1. These show that the proposed algorithms are very robust to noise and intensity homogeneities and inhomogeneities. According to Zijdenbos [31] statement that AOM> 0.7 indicates excellent agreement; the proposed method has desired performance in cortical segmentation. The best AOM is

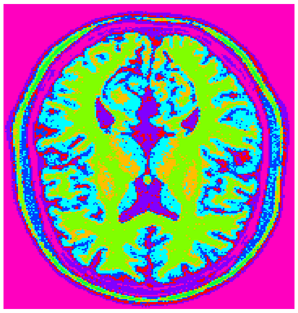


Figure 2: Results of segmentation (noise 0%)

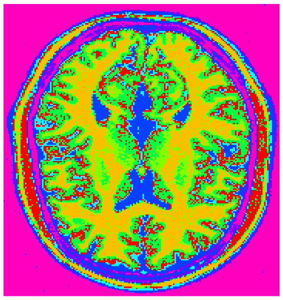


Figure 3: Results of segmentation (noise 3%)

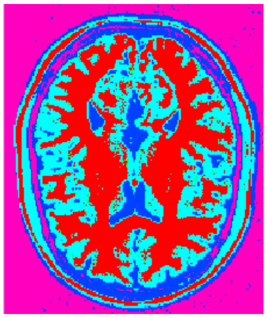


Figure 4: Results of segmentation (noise 5%)

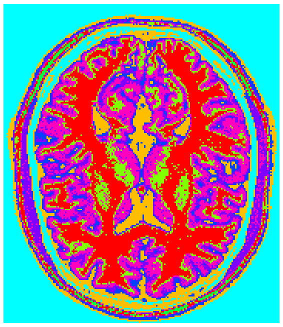


Figure 5: Results of segmentation (noise 7%)

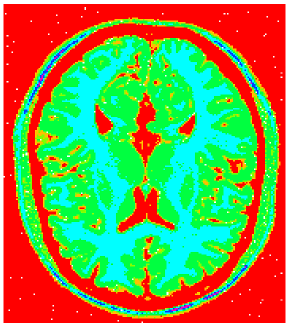


Figure 6: Results of segmentation (noise 9%)

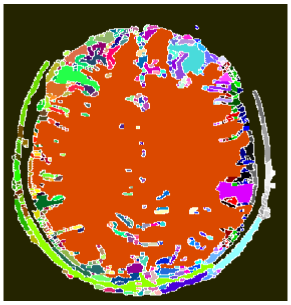


Figure 7: Results of segmentation (noise 0%)

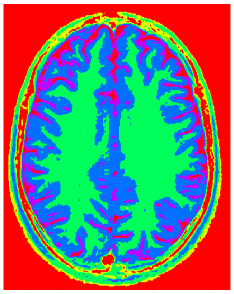


Figure 8: Results of segmentation (noise 3%)

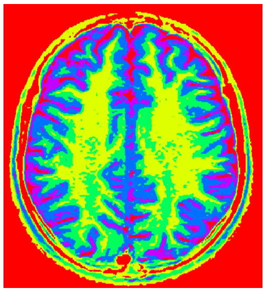


Figure 9: Results of segmentation (noise 5%)

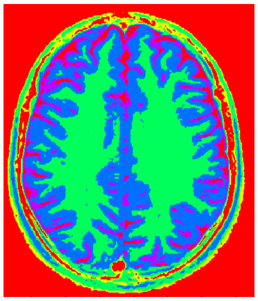


Figure 10: Results of segmentation (noise 7%)

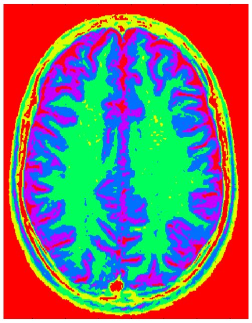


Figure 11: Results of segmentation (noise 9%)

Table 1: AOM for segmentations of WM on simulated T1-weighted MRIs data in different noise and RF levels

Noise/RF	20%
0%	0.98
3%	0.96
5%	0.96
7%	0.97
9%	0.96
7%	0.97
9%	0.96

achieved for low noise and RF levels, for which values of AOM are higher than 0.96.

4.2 Experiment on the Simulated MR Data

Table 2 shows the corresponding accuracy scores (%) of the proposed and four other methods: standard FCM [2], Ahmed et al. [1], Chen and Zhang [7], and Kang et al. [12] for the nine classes. Obviously, the FCM gives the worst segmentation accuracy for all classes, while the proposed method gives the best. On the other hand, the method of Ahmed et al. [1], Chen and Zhang [7], and Kang et al. [12] acquire the good segmentation performance in classes 9, 4, and 1, respectively. Overall, the proposed method is more stable and achieves much better performance than the others in all different classes even with the misleading of true tissue of validity indexes.

5 Conclusions

In this paper, we have proposed a fuzzy c-means method that is based on gaussian to control the coincident clusters. The proposed algorithm incorporates the local spatial context into the standard FCM cluster algorithm and its complexity is reduced using initial centers as the prior information. It is formulated by modifying the objective function of the standard FCM algorithm to allow the labelling of a pixel to be influenced by other pixels and to suppress the noise effect during segmentation. We have tested the proposed algorithm on MRI images with 3%, 5%, 7% and 9% noise. We noted that the proposed method has the desired performance in cortical segmentation. The superiority of the proposed algorithm is also demonstrated by comparing its performance

Table 2. Segmentation accuracy (%) (AOM) of the proposed and the existing methods on brain classes

Method	Class 1	Class 2	Class 3	Class 4	Class 5
Standard FCM [4]	61.87	67	69.087	64.67	75.32
Ahmed et al. [1]	77.55	61.14	78.83	73.88	67.96
Chen and Zhang [7]	69.54	78.55	68.34	82..01	78.65
Kang et al. [12]	66.87	60.43	66.98	78.54	77.09
The proposed method	83.76	78.45	80.09	90.34	83.56
Method	Class 6	Class 7	Class 8	Class 9	Overall
Standard FCM [4]	47.96	73.99	13.12	90.66	62.63
Ahmed et al. [1]	61.87	89.21	15.27	81.97	67.52
Chen and Zhang [7]	81.98	80.7	18.54	78.54	69.355
Kang et al. [12]	80.98	66.87	16.43	79.09	65.92
The proposed method	68.12	89.64	59.34	96.98	81.12

with the standard FCM, Ahmed et al. [1], Chen and Zhang [7], and Kang et al. [12]. In addition, quantitative results are also given in our experiments. We noted that the segmentation accuracy of the proposed method is increased over the existing methods between 21% and 14% for volumetric MR data (nine slices) over the best one. From the quantitative evaluation and the visual inspection, we can conclude that our proposed algorithm yields a robust and precise segmentation.

References

[1] M. N. Ahmed, S. M. Yamany, N. Mohamed, A. A. Farag, and T. Moriaty, "A Modified Fuzzy C-Means Algorithm for Bias Field Estimation and Segmentation of MRI Data," *IEEE Trans. Med. Imag.*, 21:193-199, 2002.

[2] R. Bakshi, S. Ariyaratana, R. H. B. Benedict, and L. Jacobs, "Fluid-Attenuated Inversion Recovery Magnetic Resonance Imaging Detects Cortical and Juxtacortical Multiple Sclerosis Lesions," *Arch. Neurol.* 58:742-748, 2001.

[3] M. Barni, V. Cappel
lini, and A. Mecocci, "Comments on a Possibilistic Approach to Clustering," *IEEE Transactions on Fuzzy Systems*, 4(3):393-396, 1996.

[4] J. C. Bezdek, "Pattern Recognition with Fuzzy Objective Function Algorithms," Plenum Press, NewYork, 1981.

[5] Brain Web, "Simulated Brain Database," McConnell Brain Imaging Centre, Montreal Neurological Institute, McGill, http://brainweb.bic.mni.mcgill.ca/brainweb/.

[6] W. Cai, S. Chen, and D. Zang, "Fast and Robust Fuzzy C-Means Clustering Algorithms Incorporating Local Information for Image Segmentation," *Pattern Recognition*, 40:825-838, 2007.

[7] C. Chen and D. Zhang, "A Novel Kernelized Fuzzy C-Means Algorithm with Application Medical Image Segmentation," *Artificial Intelligence in Medicine*, 32:37-50, 2004.

[8] Z. Chen, "Histogram Partition and Interval Thresholding for Volumetric Breast Tissue Segmentation," *Computerized Medical Imaging and Graphics*, 32:1-10, 2008.

[9] W. L. Hung, M Yang, and D. Chen, "Parameter Selection for Suppressed Fuzzy C-Means with an Application to MRI Segmentation," *Pattern Recognition Letters*, 27(5):424-438, 2006.

[10] Z. Ji, Q. Sun, and D. Xia, "A Modified Possibilistic Fuzzy C-Means Clustering Algorithm for Bias Field Estimation and Segmentation of Brain MR Image," *Computerized Medical Imaging and Graphics*, doi: 10.1016/j, 35(5):383-397, 2011.

[11] B. Y. Kang, D. W. Kim, and Q. Li, "Spatial Homogeneity-Based Fuzzy C-Means Algorithm for Image Segmentation," *FSKDLNAI*, 3613:462-469, 2005.

[12] J. Y. Kang, L. Q. Min, Q. X. Luan, X. Li, and J. Z. Liu, "Novel Modified Fuzzy C-Means Algorithm with Applications," *Digital Signal Processing*, 19(2):309-319, 2009.

[13] R. Krishnapuram and J. M. Keller, "Possibilistic Approach to Clustering," *IEEE Transactions On Fuzzy Systems*, 1(2):98-100, 1993.

[14] A. W. C. Liew and H. Yan, "An Adaptive Spatial Fuzzy Clustering Algorithm for 3-D MR Image Segmentation," *IEEE Transactions on Medical Imaging*, 22(9):1063-1075, 2003.

[15] F. Masulli and A. Schenone, "A Fuzzy Clustering Based Segmentation System as Support to Diagnosis in Medical Imaging," *Artificial Intelligent Med*, 16(2):29-47, 1999.

[16] T. N. Pappas, "An Adaptive Clustering Algorithm for Image Segmentation," *IEEE Trans. Signal Process*, 40(4):901-914, 1992.

[17] D. Pham and J. Prince, "Adaptive Fuzzy Segmentation of Magnetic Resonance Images," *IEEE Transactions on Medical Imaging*, 18(9):737-752, 1999.

[18] S. Shen, W. Sandham, M. Granat, and A. Sterr, "MRI Fuzzy Segmentation of Brain Tissue using Neighborhood Attraction with Neural-Network Optimization", *IEEE Trans. Inf. Technology Biomedicine*, 9:459-467, 2005.

[19] M. C. Su and C. H. Chou, "A Modified Version of the K-Means Algorithm with a Distance Based on Cluster Symmetry," *IEEE Transactions Pattern Analysis and Machine Intelligence*, 23(6):680, 2001.

[20] L. Szilágyi, S. M. Szilágyi, and Z. Benyó, "A Modified FCM Algorithm for Fast Segmentation of Brain MR Images," *ICIARLNCS*, 4633:866-877, 2007.

[21] H. Timm, C. Borgelt, C. Doring, and R. Kruse, "An Extension to Possibilistic Fuzzy Cluster Analysis," *Fuzzy Sets and Systems*, 147(1):3-16, 2004.

[22] J. Z. Wang, J. Kong, Y. H. Lu, M. Qi, and B. X. Zhang, "A Modified FCM Algorithm for MRI Brain Image Segmentation using Both Local and Non-Local Spatial Constrains," *Computerized Medical Imaging and Graphics*, 31(8):685-698, 2008.

[23] Z. M. Wang, Y. C. Soh, Q. Song, and K. Sim, "Adaptive Spatial Information-Theoretic Clustering for Image Segmentation," *Pattern Recognition Letters*, 42:2029-2044, 2009.

[24] J. H. Xue, A. Piˇzurica, W. Philips, E. Kerre, R. Van de Walle, and I. Lemahieu, "An Integrated Method of Adaptive Enhancement for Unsupervised Segmentation of MRI Brain Images," *Pattern Recognition Letters*, 24:2549-2560, 2003.

[25] E. A. Zanaty, S. Aljahdali, and N. Debnath, "Improving Fuzzy Algorithms for Automatic Magnetic Resonance Image Segmentation," *Proceedings of seventeenth International Conference of Software Engineering and Data Engineering*, Los Angeles, California, USA, pp.60-66, June 2008.

[26] E. A. Zanaty and S. Aljahdali, "Fuzzy Algorithms for

Automatic Magnetic Resonance Image Segmentation," *International Arab Journal of Information Technology (IAJIT)*, 7(3):271-279, 2009.

[27] E. A. Zanaty, S. Aljahdali, and N. Debnath, "A Kernelized Fuzzy C-Means Algorithm for Automatic Magnetic Resonance Image Segmentation," *Journal of Computational Methods in Science and Engineering (JCMSE)*, 1:123-136, 2009.

[28] E. A. Zanaty and Sultan Aljahdali, "Automatic Fuzzy Algorithms for Reliable Image Segmentation," *IJCA Journal*, 19(3):166-175, Sept. 2012.

[29] J. S. Zhang and Y. W. Leung, "Improved Possibilistic C-Means Clustering Algorithms," *IEEE Transactions On Fuzzy Systems,* 12(2):209-17, 2004.

[30] L. Zhu, F. L. Chung, and S. Wang, "Generalized Fuzzy C-Means Clustering Algorithm with Improved Fuzzy Partitions," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 39(3):578-591, 2009.

[31] A. P. Zijdenbos, *MRI Segmentation and the Quantification of White Matter Lesions*, PhD Thesis, Vanderbilt University, Engineering Department, Nashville, Tennessee, December 1994.

Khalid A. Buragga: is an Associate Professor in the faculty of Computing and Information Technology at Northern Border University in Saudi Arabia, where he also holds the position of Vice President for Academic Affairs. Dr. Buragga received his B.Sc. in Computer Information Systems from King Faisal University and received his M.Sc. in Computer Information Systems from University of Miami, USA. Dr. Buragga obtained his Ph.D. in Information Technology from George Mason University, USA. His research interests include software design, software development, software quality, software reliability, E-Commerce and web development, business process re-engineering, communications, networking and signal processing.

Sultan Aljahdali is a Professor of Computer Science in the College of Computers and Information Technology at Taif University, Taif, Saudi Arabia. Professor Aljahdali obtained a Ph.D. in Information Technology from George Mason University, USA in 2003, and completed a Leadership Program at Harvard Business School. Harvard University, Boston, USA in 2013. He received his B.S from Winona State University and M.S. with honors from Minnesota State University in Mankato. Professor Aljahdali has made research contributions on software testing, developing Software Reliability Models, Soft Computing for Software Engineering, Computer Security, Reverse Engineering and Medical Imaging. He is an author or co-author of more than 110 peer reviewed academic publication. Professor Aljahdali is a member of several professional societies including ACM, IEEE, ACS and ISCA.

E. A. Zanaty is Professor at Taif University, Saudi Arabia. He received a BSC from Assuit University in 1992 and MSC Degree in Computer Science in 1997 from South Valley University, Egypt. He completed his PhD. studies at TU-Chemnitz, Germany, during the period 2000-2003.

Professor Zanaty is an Editorial board member of several journals and member of KACST and IAENG. His research interests include artificial intelligence, data classification, reverse engineering data reduction, medical image segmentation and reconstruction. In these areas he has published several technical papers in refereed international journals or conference proceedings..

Instructions For Authors

The International Journal of Computers and Their Applications is published multiple times a year with the purpose of providing a forum for state-of-the-art developments and research in the theory and design of computers, as well as current innovative activities in the applications of computers. In contrast to other journals, this journal focuses on emerging computer technologies with emphasis on the applicability to real world problems. Current areas of particular interest include, but are not limited to: architecture, networks, intelligent systems, parallel and distributed computing, software and information engineering, and computer applications (e.g., engineering, medicine, business, education, etc.). All papers are subject to peer review before selection.

A. Procedure for Submission of a Technical Paper for Consideration

1. Email your manuscript to the Editor-in-Chief, Dr. Fred Harris, Jr., Fred.Harris@cse.unr.edu.

2. Illustrations should be high quality (originals unnecessary).

3. Enclose a separate page (or include in the email message) the preferred author and address for correspondence. Also, please include email, telephone, and fax information should further contact be needed.

B. Manuscript Style:

1. The text should be **double-spaced** (12 point or larger), **single column** and **single-sided** on 8.5 X 11 inch pages.
2. An informative abstract of 100-250 words should be provided.
3. At least 5 keywords following the abstract describing the paper topics.
4. References (alphabetized by first author) should appear at the end of the paper, as follows: author(s), first initials followed by last name, title in quotation marks, periodical, volume, inclusive page numbers, month and year.
5. Figures should be captioned and referenced.

C. Submission of Accepted Manuscripts

1. The final complete paper (with abstract, figures, tables, and keywords) satisfying Section B above in **MS Word format** should be submitted to the Editor-in-Chief.
2. The submission may be on a CD/DVD or as an email attachment(s) . **The following electronic files should be included:**

    - Paper text (required).
    - Bios (required for each author). Integrate at the end of the paper.
    - Author Photos (jpeg files are required by the printer, these also can be integrated into your paper).
    - Figures, Tables, Illustrations. These may be integrated into the paper text file or provided separately (jpeg, MS Word, PowerPoint, eps).

3. Specify on the CD/DVD label or in the email the word processor and version used, along with the title of the paper.

4. Authors are asked to sign an ISCA copyright form (http://www.isca-hq.org/j-copyright.htm), indicating that they are transferring the copyright to ISCA or declaring the work to be government-sponsored work in the public domain. Also, letters of permission for inclusion of non-original materials are required.

Publication Charges

After a manuscript has been accepted for publication, the contact author will be invoiced for publication charges of **$50.00 USD** per page (in the final IJCA two-column format) to cover part of the cost of publication. For ISCA members, $100 of publication charges will be waived if requested.

January 2014

ISCA INTERNATIONAL JOURNAL OF COMPUTERS AND THEIR APPLICATIONS Vol. 22, No. 3, Sept. 2015