Title	Communication packet loss concealment for pattern generation with robotic swarms
Author(s)	Shinnoh, Atsushi; Chong, Nak Young; Lee, Geunho
Citation	2015 IEEE International Conference on Advanced Intelligent Mechatronics (AIM): 537-542
Issue Date	2015
Type	Conference Paper
Text version	author
URL	http://hdl.handle.net/10119/14231
Rights	This is the author's version of the work. Copyright (C) 2015 IEEE. 2015 IEEE International Conference on Advanced Intelligent Mechatronics (AIM), 2015, 537-542. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
Description	

Communication Packet Loss Concealment for Pattern Generation with Robotic Swarms

Atsushi Shinnoh[1], Nak Young Chong[1], and Geunho Lee[2]

***Abstract*— In swarm robotics, a multitude of very simple robots often move to achieve pre-defined geometric patterns while communicating with each other. Robots are able to estimate their position and relocate themselves by obtaining other robots' position information. However, as is the often the case in wireless communication, robots cannot receive other robots' position information reliably. Thus, they need to deal with the packet loss problem and estimate the missing information in near-real time. Several nonlinear filters such as extended Kalman filters or particle filters have been widely used for many years. But these filters require enormous computational complexity, so it is difficult to be applied to low-cost mobile robots with limited computational and memory resources. To overcome this problem, we propose an extrapolating method with reduced computational cost yet high estimation accuracy. Specifically, we propose a novel exponential function concealment with linear blends, and validate its effectiveness and rate of convergence through extensive simulations.**

I. INTRODUCTION

One of the most important requirements for robots is fault tolerance or robustness under real-world conditions. Due to several advantages to the use of multiple robots over a single robot including the fault-tolerance and scalability, swarm robotics is gaining increasing attention [1]. Swarm robotics is based on the swarm intelligence which generates emergent behavior of a whole group with no leaders [2], and is further applied to the field of robotic perimeter surveillance and coverage, and cooperative transport [3]. Robotic swarms, consisting of many simple individual robots, act as a single entity, and thus the failure of one or a small number of robots does not affect the whole group behavior. For this, a variety of local interaction and motion coordination algorithms have been proposed. Recently, from the viewpoint of control theory, the idea of Glocal control attempts to control the motion of a whole system by controlling the motion of individual agents [4], which can be effectively applied to robotic swarms [5], [6].

Specifically, distributed formation control is the key to deploying a swarm of robots to pre-defined formations, such as equilateral triangle formation [7] and circle formation [8], [9]. Moreover, extensive efforts are being devoted to navigate robots in cluttered environments, while keeping the desired formation and avoiding unknown obstacles [10]. Therefore, it is necessary for individual robots to exchange their state information to realize cooperative group behaviors. One technical challenge is that unavoidable packet losses/collisions happen when large populations of robots are communicating at the same time in the same location. Furthermore, note that wireless technology may not be available in communication-restrict environments including the nuclear power plants. If individual robots cannot obtain the positional information of other robots, they need to minimize the impact of missing information. In this paper, specifically, we attempt to solve the problem in such a way that each robot memorizes the other robots' positional information and estimates the lost information from past states when secure and reliable inter-robot communications are not maintained.

In estimation and control theory, nonlinear filters such as Extended Kalman Filters (EKF) or Particle Filters (PF) have been widely used for many years. However, the main drawback is that these filters in most cases require a lot of computation. They cannot be employed for packet loss concealment by simple robots with low computational power and limited memory. On the other hand, there are several ways to estimate/extrapolate the missing packet information such as zero-order hold or linear extrapolation. These methods reduce the necessary computation and storage, but do not guarantee the accuracy of estimation. We have confirmed in our previous study that exponential function based extrapolation methods can estimate missing information more accurately compared to the aforementioned extrapolation methods such as zero-order hold and EKF. To further extend this idea, in this paper, we propose a novel exponential function based extrapolation method with linear blends, and show that the proposed estimation method has low computational complexity almost equivalent to the zero-order hold case, and its estimation accuracies and rate of convergence are superior to any existing methods.

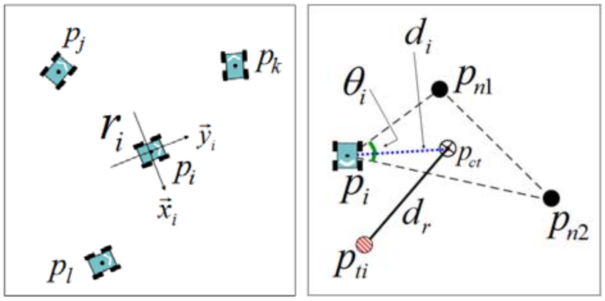
II. PROBLEM STATEMENT

A. Robot Model and Motion Controls

In this paper, we consider a swarm of autonomous mobile robots $r_1, r_2, \cdots, r_n$ located on a two-dimensional (2D) plane. In the swarm, each robot r_i modeled as a point freely moves on a 2D plane. It is assumed that an initial distribution of all robots is arbitrary and their positions are distinct. As illustrated in Fig. 1-(a), the local coordinate system of r_i is represented as $\vec{x}_i$ and $\vec{y}_i$. Here, $\vec{y}_i$ defines the vertical axis of the r_i's coordinate system as its heading direction. It is straightforward to determine the horizontal axis $\vec{x}_i$ by rotating the vertical axis 90 degrees clockwise. Moreover, p_i denotes the position of r_i. Next, a path is defined as

[1]Atsushi Shinnoh and Nak Young Chong are with the School of Information Science, Japan Advanced Institute of Science and Technology, 1-1 Asahidai, Nomi-shi, Ishikawa-ken, 923-1292 Japan {s1120902,nakyoung}@jaist.ac.jp

[2]Geunho Lee is with the Department of Environmental Robotics, University of Miyazaki, 1-1 Gakuen Kibanadai-nishi, Miyazaki-shi, Miyazaki-ken, 889-2192 Japan geunho@miyazaki-u.ac.jp



(a) local coordinates: $\vec{x}_i$ and $\vec{y}_i$ (b) motion controls: d_i and θ_i

Fig. 1. Illustration of definitions and notations with respect to a robot r_i

the shortest straight line between p_i and p_j occupied by another robot r_j, and denoted by $\overline{p_ip_j}$. The distance between p_i and p_j is denoted as $dist(p_i, p_j)$. In particular, a uniform distance, the predefined desired interval between r_i and r_j, is defined as d_u. Moreover, $ang(\vec{s}, \vec{t})$ denote the angle between two arbitrary paths $\vec{s}$ and $\vec{t}$.

As depicted in Fig. 1-(a), r_i obtains the positions p_j, p_k, and p_l of other robots through communications, yielding a set of the positions C_i (=$\{p_j, p_k, p_l\}$) with respect to its local coordinates. When r_i selects two robots r_{n1} and r_{n2} in C_i, we call r_{n1} and r_{n2} the neighbors of r_i, and a set of the positions $\{p_{n1}, p_{n2}\}$ is denoted as N_i. Given p_i and N_i, a set of three distinct positions $\{p_i, p_{n1}, p_{n2}\}$ with respect to r_i is called triangular configuration, denoted as $\mathbb{T}_i$, namely $\{p_i, p_{n1}, p_{n2}\}$. In Fig. 1-(b), p_{ct} and p_{ti} denote a centroid of $\mathbb{T}_i$ and a target point to be moved based on $\mathbb{T}_i$. Specifically, we define an equilateral configuration, denote as $\mathbb{E}_i$, as a configuration that all distances between any two of p_i, p_{n1}, and p_{n2} of $\mathbb{T}_i$ are equal to d_u.

In our previous work [7], a locally interacting geometric technique for mobile robot swarms was proposed, allowing them to exponentially converge to a uniform equilateral triangle formation by forming a desired $\mathbb{E}_i$ with their two neighbors. Based on $\mathbb{E}_i$, r_i's motion controls were designed by controlling the distance d_i from p_{ct} and the internal angle θ_i between $\overrightarrow{p_ip_{n1}}$ and $\overrightarrow{p_ip_{n2}}$ (see Fig. 1-(b)). First, in detail, d_i is controlled:

$$\dot{d}_i(t) = -a\big(d_i(t) - d_r\big) \tag{1}$$

where a is a positive constant and d_r represents the length $d_u/\sqrt{3}$. Indeed, the solution of (1) is given:

$$d_i(t) = \big|d_i(0)\big|e^{-at} + d_r \tag{2}$$

Here, $d_i(t)$ converges exponentially to $d_r(= d_u/\sqrt{3})$ as t approaches infinity. Next, using the geometric features of a triangle whose total external angles is π, θ_i is controlled:

$$\dot{\theta}_i(t) = k\big(\frac{\pi}{3} - \theta_i(t)\big) \tag{3}$$

where k is a positive constant. Here, the solution of (1) is obtained:

$$\theta_i(t) = \big|\theta_i(0)\big|e^{-kt} + \frac{\pi}{3} \tag{4}$$

Similarly, $\theta_i(t)$ converges exponentially to $\frac{\pi}{3}$ as t approaches infinity. Note that (2) and (4) imply that the trajectory of r_i converges to the desired equilibrium state $X_e = [d_r\ \frac{\pi}{3}]^T$ exponentially.

B. Problem Definition

Even though r_i exponentially converge into $\mathbb{T}_i$ by controlling d_i and θ_i, the following question arises: how does r_i obtain the positions of other robots? As mentioned in the previous subsection, it was assumed that r_i could configure and update C_i of its adjacent robots through communications. When the inherent unreliable nature of *ad hoc* networks is considered, r_i might suffer from data loss necessary for C_i, resulting from latent aspects such as a loss of communication (and mutual interference). Specifically, in our paper, the terminology of data loss defines that any data transmitted by r_j are destroyed by failures in transmission. The happening frequency of data loss is expressed as α during communications. Furthermore, α is considered as a constant in probability distribution. Therefore, r_i needs to estimate the positions of robots in C_i against data loss. Then, we formally address the *Position Extrapolation Problem* for a swarm of robots based on aforementioned model definitions as follows: *Given a swarm of robots $r_1, \cdots, r_n$ with arbitrarily distinct positions in 2-D plane, despite the communication losses among robots, how can r_i converge to $\mathbb{E}_i$ under the motion controls (1) and (3) after a finite number of time steps.*

Let us consider the following a situation: r_i cannot configure $C_i(t)$ due to a loss of communication at a time t. Accordingly, $\mathbb{T}_i(t)$ is not updated. This means that it would be impossible for r_i to determine $[d_i(t)\ \theta_i(t)]^T$ obtained from (1) and (3). To overcome this undesirable situation, we need to extrapolate lost information. Specifically, the proposed scheme allows r_i to extrapolate its neighbor's position at t based on the previous neighbor's locations obtained at several past steps. Consequently, our goal aims at achieving the r_i's coordinated motions through communications with α toward its $\mathbb{E}_i$.

Next, another aspect is considered: a trade-off between computational complexity and estimation accuracy. Moderate computation to coordinate a swarm of n robots cannot be simply obtained by reducing estimation accuracy. Due to scalability, it would be difficult to be in complementary relations between two requirements. Despite its computational complexity, our solution approach to the above problem focuses on the coordinated motions of robot swarms using the proposed extrapolation scheme with high estimation accuracy.

III. EXPONENTIAL EXTRAPOLATION

This section introduces the proposed extrapolating scheme featuring high estimation accuracy yet less computational complexity, allowing r_i extrapolate its $N_i(Ts)$ at Ts when there was a loss of communication, where T stands for sampling time. To assist the explicit understanding of the proposed scheme, it is assumed that $N_i(T(s-1))$ and $N_i(T(s-2))$ of r_i are observable, respectively, at $T(s-1)$ and $T(s-2)$. Formally, $N_i(T(s-1))$ and $N_i(T(s-2))$ of r_i are defined as the positions of the neighbors with respect to $\vec{x}_i$ and $\vec{y}_i$, respectively. According to a differences between $dist(p_i(Ts), N_i(T(s-1)))$ and $dist(p_i(Ts), N_i(T(s-2)))$, three cases are considered as follows: $dist(p_i(Ts), N_i(T(s-1))) < dist(p_i(Ts), N_i(T(s-2)))$, $dist(p_i(Ts), N_i(T(s-1))) > dist(p_i(Ts), N_i(T(s-2)))$, and $dist(p_i(Ts), N_i(T(s-1))) = dist(p_i(Ts), N_i(T(s-2)))$. For simplicity, these distance differences are defined as $|N_i(T(s-1))| < |N_i(T(s-2))|$, $|N_i(T(s-1))| > |N_i(T(s-2))|$, and $|N_i(T(s-1))| = |N_i(T(s-2))|$. Note that there are no changes for N_i despite the variations of Ts. This means r_i and its neighbors could converge into $\mathbb{E}_i$. Since it is useless to continue the description any longer, our discussions for the third case are omitted afterward.

A. Exponential Extrapolation (EE)

Here, we consider x (y) component $x_i(Ts)$ ($y_i(Ts)$) of $N_i(Ts)$. In this subsection, we only disccuss the case $x_i(Ts)$. For $y_i(Ts)$, same treatment with $x_i(Ts)$ is applied.

- Case-1: $x_i(T(s-2)) > x_i(T(s-1))$

To begin, we recall that robots trigger their exponential motions based on (1) and (3). As a general formula of $x_i(Ts)$ with respect to r_i, at Ts, the motion of r_j is defined:

$$x_i(Ts) = A_i e^{-B_i Ts}, \tag{5}$$

where $A_i > 0$ and $B_i > 0$. By using (5), $x_i(T(s-1))$ and $x_i(T(s-2))$ are given:

$$x_i(T(s-1)) = A_i e^{-B_i T(s-1)} \tag{6}$$

and

$$x_i(T(s-2)) = A_i e^{-B_i T(s-2)}. \tag{7}$$

From (6) and (7), $e^{-B_i T}$ is easily derived:

$$e^{-B_i T} = \frac{x_i(T(s-1))}{x_i(T(s-2))}. \tag{8}$$

By substituting (8) into (5), $x_i(Ts)$ is written:

$$\begin{aligned} x_i(Ts) &= A_i e^{-B_i Ts} = A_i e^{-B_i(T(s-1))} \cdot e^{-B_i T} \\ &= \frac{x_i(T(s-1))}{x_i(T(s-2))} \cdot x_i(T(s-1)). \end{aligned} \tag{9}$$

- Case-2: $x_i(T(s-2)) < x_i(T(s-1))$

Similar with (5), as a general formula for another case, the motion of r_j with respect to r_i at Ts is defined:

$$x_i(Ts) = A_i(1 - e^{-B_i Ts}), \tag{10}$$

where $A_i > 0$ and $B_i > 0$. By using (10), $x_i(T(s-1))$ and $x_i(T(s-2))$ are given:

$$x_i(T(s-1)) = A_i(1 - e^{-B_i T(s-1)}) \tag{11}$$

and

$$x_i(T(s-2)) = A_i(1 - e^{-B_i T(s-2)}). \tag{12}$$

When we replace $T(s-1)$ and $T(s-2)$, respectively, with $2T$ and T, $x_i(2T)$ and $x_i(T)$ are expressed:

$$x_i(2T) = A_i(1 - e^{-2B_i T}) \tag{13}$$

and

$$x_i(T) = A_i(1 - e^{-B_i T}). \tag{14}$$

By using (10), (13), and (14), A_i and $e^{-B_i T}$ are computed:

$$A_i = \frac{\left[x_i(T)\right]^2}{x_i(2T)} \tag{15}$$

and

$$e^{-B_i T} = \frac{x_i(T) - x_i(2T)}{x_i(T)} = 1 - \frac{x_i(2T)}{x_i(T)}. \tag{16}$$

Similarly, the following equation holds:

$$x_i(3T) = A_i(1 - e^{-3B_i T}) = \frac{\left[x_i(T)\right]^2}{x_i(2T)} \cdot \Big[1 - \big[1 - \frac{x_i(2T)}{x_i(T)}\big]^3\Big]. \tag{17}$$

The relation for Tt becomes the following generalized equation:

$$x_i(Ts) = \frac{\left[x_i(T(s-2))\right]^2}{x_i(T(s-1))} \cdot \Big[1 - \big[1 - \frac{x_i(T(s-1))}{x_i(T(s-2))}\big]^3\Big]. \tag{18}$$

B. Extended Exponential Extrapolation (EEE)

As described in Section III-A, the EE scheme calculates the relative distance difference between $dist(p_i(Ts), N_i(T(s-1)))$ and $dist(p_i(Ts), N_i(T(s-2)))$. Based on the computations, r_i extrapolates where r_j moved. Specifically, the EE scheme is represented as the sign of inequality such as $|N_i(T(s-2))| < |N_i(T(s-1))|$ and $|N_i(T(s-2))| > |N_i(T(s-1))|$. In other words, r_i determines that r_j gradually recedes or approaches. Furthermore, r_j triggers its exponential motions under its (1) and (3). Since r_j's motions are matched to the extrapolation, $N_i(Ts)$ can be obtained. However, when it was assumed that α was many losses of data, we examined how accurate r_i could extrapolated $N_i(Ts)$ under the scheme. Due to the losses, the results were not good enough.

To overcome this limitation, the EE scheme (i.e., (9) or (18)) is advanced. To enhance the $N_i(Ts)$ extrapolation, the following two aspects are considered. First, a rate of change to $x_i(T(s-1))$ ($y_i(T(s-1))$) from $x_i(T(s-2))$ ($y_i(T(s-2))$) is examined how far r_j moved from $x_i(T(s-2))$ ($y_i(T(s-2))$) to $x_i(T(s-1))$ ($y_i(T(s-1))$). Secondly, the direction of the rate is determined. As the r_j's displacement and direction in the rate is employed, we intend to minimize a discrepancy between the r_i's estimates (mostly caused by the inaccurate extrapolations) and the r_j's actual movements. In detail, after the computation of the rate from r_j's movements, the extended exponential extrapolation (EEE) scheme yields an opposite-sign extrapolated value with inverse proportion to the rate. According to the conceptual ideas, the EEE scheme is formally described:

Here, same as subsection III-A, we only discuss the case of $x_i(Ts)$.

- Case-1: $x_i(T(s-2)) > x_i(T(s-1))$

According to the conceptual idea of the EEE scheme, $x_i(Ts)$ is formalized:

$$
\begin{aligned}
x_i(Ts) &= x_i(T(s-1)) + (x_i(T(s-1)) - x_i'(Ts)) \\
&= \Big[2 - \frac{x_i(T(s-1))}{x_i(T(s-2))} \cdot x_i(T(s-1))\Big],
\end{aligned} \tag{19}
$$

where, by using (9), $x_i'(Ts)$ is defined:

$$
x_i'(Ts) = \frac{x_i(T(s-1))}{x_i(T(s-2))} \cdot x_i(T(s-1)). \tag{20}
$$

- Case-2: $x_i(T(s-2)) < x_i(T(s-1))$

$$
\begin{aligned}
&x_i(Ts) = x_i(T(s-1)) - (x_i'(Ts) - x_i(T(s-1))) \\
&= 2x_i(T(s-1)) + \Big[\big[1 - \frac{x_i(T(s-1))}{x_i(T(s-2))}\big]^3 - 1\Big] \cdot \frac{(x_i(T(s-1)))^2}{x_i(T(s-2))},
\end{aligned} \tag{21}
$$

where, by using (18), $x_i'(Ts)$ is given:

$$
x_i'(Ts) = \frac{\big[x_i(T(s-2))\big]^2}{x_i(T(s-1))} \cdot \Big[1 - \big[1 - \frac{x_i(T(s-1))}{x_i(T(s-2))}\big]^3\Big]. \tag{22}
$$

IV. Exponential Extrapolation with First-Degree Polynomial Function

In this section, a more advanced exponential extrapolation (EEF) scheme added to a first-degree polynomial function is introduced towards more accurate estimation yet less computation cost than EE and EEE schemes. It is assumed that $N_i(T(s-1))$, $N_i(T(s-2))$, $N_i(T(s-3))$ and $N_i(T(s-4))$ of r_i are observable, respectively, at $T(s-1)$, $T(s-2)$, $T(s-3)$, and $T(s-4)$, where T stands for sampling time. Formally, $N_i(T(s-1))$, $N_i(T(s-2))$, $N_i(T(s-3))$, and $N_i(T(s-4))$ of r_i are defined as the positions of the neighbors with respect to $\vec{x}_i$ and $\vec{y}_i$, respectively. Same as section III, we only consider x component $x_i(Ts)$ of $N_i(Ts)$. Same discussion is applied to y component $y_i(Ts)$ of $N_i(Ts)$.

A. Formalization

The EEF scheme is formalized:

$$
x_i(Ts) = A_i e^{-B_i Ts} + C_i Ts + D_i \tag{23}
$$

where A_i, C_i, and D_i are constants and B_i is a positive constant. When there is a loss of communication, r_i extrapolate its $x_i(Ts)$ at Ts based on $x_i(T(s-1))$, $x_i(T(s-2))$, $x_i(T(s-3))$, and $x_i(T(s-4))$. In detail, by substituting $x_i(T(s-1))$, $x_i(T(s-2))$, $x_i(T(s-3))$, and $x_i(T(s-4))$ for (23), respectively, individual state equations can be represented:

$$
x_i(T(s-1)) = A_i e^{-B_i T(s-1)} + C_i T(s-1) + D_i, \tag{24}
$$

$$
x_i(T(s-2)) = A_i e^{-B_i T(s-2)} + C_i T(s-2) + D_i, \tag{25}
$$

$$
x_i(T(s-3)) = A_i e^{-B_i T(s-3)} + C_i T(s-3) + D_i, \tag{26}
$$

and

$$
x_i(T(s-4)) = A_i e^{-B_i T(s-4)} + C_i T(s-4) + D_i. \tag{27}
$$

In these equations, the following four parameters A_i, B_i, C_i, and D_i are summarized:

$$
A_i = \frac{b_2^{T(s-1)}}{(b_3)^2 (b_1)^{T(s-4)}}, \tag{28}
$$

$$
e^{-B_i} = \frac{b_1}{b_2}, \tag{29}
$$

$$
C_i = x_i(T(s-3)) - x_i(T(s-4)) - \frac{(b_2)^2}{b_3}, \tag{30}
$$

and

$$
D_i = x_i(T(s-4)) - A_i e^{-B_i T(s-4)} - C_i T(s-4), \tag{31}
$$

where

$$
b_1 = x_i(T(s-1)) - 2x_i(T(s-2)) + x_i(T(s-3)),
$$

$$
b_2 = x_i(T(s-2)) - 2x_i(T(s-3)) + x_i(T(s-4)),
$$

and

$$
b_3 = x_i(T(s-1)) - 3x_i(T(s-2)) + 3x_i(T(s-3)) - x_i(T(s-4)).
$$

(Note that the detailed development processes are described in Appendix.) Finally, $x_i(Ts)$ can be extrapolate by substituting the four parameters for (23).

Next, an extended exponential extrapolation scheme (EEEF) added to a first-degree polynomial function is described as an enhanced version for the EEE scheme. Similar with the EEE scheme, r_i computes $x_i'(Ts)$ based on (23), and extrapolates $x_i(Ts)$ in (19) or (21).

B. Simulation Results

This part shows evaluation results for configuring the desired geometric pattern by using the proposed scheme. In the evaluation tests, 10 robots with initially arbitrary distributions attempt to form a swarm network composed of $\mathbb{E}_i$. The robots exchange $C_i(Ts)$ each other through communications with α while moving according to (1) and (3). In the case of a loss of communication, r_i can obtain its $C_i(Ts)$ by using one among the schemes. Here, α is defined as a constant in probability distribution.

In our tests, two types of computations according to the communication conditions are set. As an ideal situation, there is no loss of communication. On the other hand, due to a data loss, r_i needs to extrapolate its $C_i(Ts)$ by using one of the proposed schemes. During simulations, both conditions are concurrently performed. Meanwhile, to examine relative errors between the individual conditions quantitatively, an evaluation index is defined as l_α, and is formalized:

$$
l_\alpha = \frac{\sum_{j=1}^{n-1} \sqrt{(x_j - x_j')^2 + (y_j - y_j')^2}}{n-1}, \tag{1}
$$

where (x_i', y_i') and (x_i, y_i) are the representations of p_j according to the existence of data loss, respectively. Clearly, l_α indicates an average value of the position errors for two kinds of p_j occurred by the loss of communication. Next, to evaluate the proposed scheme qualitatively, simulations were

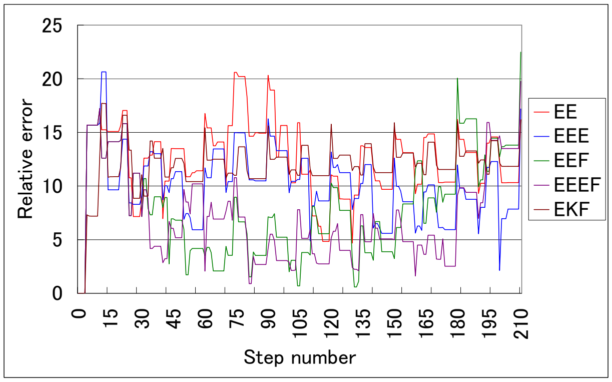


Fig. 2. Simulation results for the comparisons of l_α when α=0.2

performed by using the extended Kalman filter (EKF) under the same conditions.

In Fig.2, we show the simulation results at loss probability α=0.2, and show the results until step number n=210. Here, robots' motions under above-mentioned condition are plotted. The red, blue, green, purple, and blown lines represent the results performed by EE, EEE, EEF, EEEF, and EKF schemes, respectively. We could see that the trends of l_α under the EEF and the EEEF schemes are more accurate than the EE and the EEE schemes and EKF. Despite data loss, the EEF and the EEEF schemes allowed individual robots to extrapolate $N_i(Ts)$, resulting in collecting the local geometries can globally reach the desired geometric pattern without centralized control schemes.

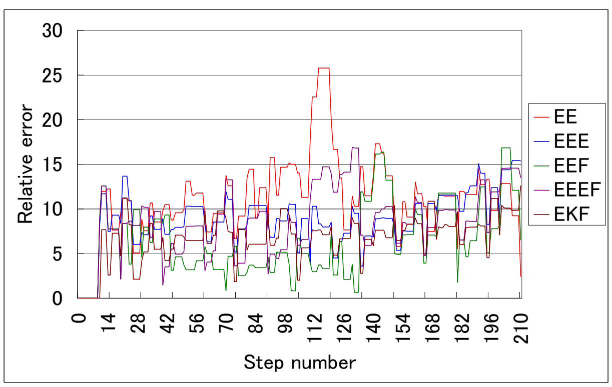


Fig. 3. Simulation results for the comparisons of l_α when α=0.5

Next, Fig. 3 is the simulation results for the comparisons of l_α when $\alpha = 0.5$. Although the increased α had a bad influence on the position extrapolations, it is verified that the EEF and the EEEF schemes allowed r_i to estimate $N_i(Ts)$ more accurately than the EE and the EEE schemes and EKF. As we expected, the EEF and the EEEF schemes could yield better accuracy despite the increased l_α. More notably, the EEF and the EEEF schemes less affected by α enabled r_i to more swiftly form $\mathbb{E}_i$.

Now, from the obtained results, we emphasize two points that highlight unique features of the EEF and the EEEF schemes. First, the schemes enable a swarm of robots to disperse in an infrastructure-less environment when the their communications are subject to range limitations and interferences. Secondly, the proposed scheme is computationally efficient, since each robot cannot memory both its past information and the position data of all other robots. When there occurs data loss, the schemes utilizes only position data of other robots at $T(s-1)$, $T(s-2)$, $T(s-3)$, and $T(s-4)$. We believe that the EEF and the EEEF schemes work well under real world conditions, but several issues remain to be addressed. It would be interesting to investigate several performances such as convergence time or optimal coverage when α is time-varying. Regarding using explicit communications, it also suffers from limited bandwidth, range, interferences, and traffic. This important engineering issue is left for future work.

V. CONCLUSIONS

In this paper, we assumed that there could be unavoidable communication packet losses in robotic swarms due to various reasons, and proposed a novel extrapolating method with low computational complexity and high accuracy. Through extensive simulations, where packets randomly dropped and lost, we tested the effectiveness of packet loss concealment algorithms under 0-50% packet loss conditions. We specifically propose an efficient extrapolation method based on an exponential function with linear blends, and showed that the proposed method estimate robots' positions more accurately than existing packet loss concealment algorithms. We believe that this method can be an attractive algorithm due to its computational efficiency and effectively applied to low-cost, even disposable robotic swarms in communication-restricted environments.

In this paper, we have validated the proposed method only for the case of equilateral triangle formation, but from the convergence properties and motion characteristics of formation generating mobile robots, it is expected that the proposed method is similarly applicable and beneficial to other formation control cases with the same equation of motion of robots.

APPENDIX

Derivation of equations of (28)-(31).
From (25) to (27),

$$b_1 = A_i e^{-B_i T(s+1)}(e^{-2B_i T} - 2e^{-B_i T} + 1) = A_i e^{-B_i T(s+1)}(e^{-B_i T} - 1)^2 \quad (a)$$

Similarly, from (24) to (26),

$$b_2 = A_i e^{-B_i Ts}(e^{-2B_i T} - 2e^{-B_i T} + 1) = A_i e^{-B_i Ts}(e^{-B_i T} - 1)^2 \quad (b)$$

From (a) and (b), we obtain (29).

$$b_1 = A_i e^{-B_i Ts}(e^{-B_i T} - 1)^2 = b_2 e^{-B_i T}$$

$$\therefore e^{-B_i T} = \frac{b_1}{b_2}$$

Substituting (c) into (a), we get (28).

$$A_i = \frac{b_1}{e^{-B_i T s}(e^{-B_i T} - 1)^2} = \frac{b_2^{T(s-1)}}{b_3^2 b_1^{T(s-4)}}$$

Here, (26)-(27),

$$x_i(T(s-3)) - x_i(T(s-4)) = A_i e^{-B_i T s}(e^{-B_i T} - 1) + C_i \quad (c)$$

From (c), we obtain (29).

$$C_i = x_i(T(s-3)) - x_i T(s-4)) - A_i e^{-B_i T s}(e^{-B_i T} - 1) = \frac{b_2^2}{b_3}$$

From (27), we get (31).
□

REFERENCES

[1] E. Sahin, Swarm Robotics: From Sources of Inspiration to Domains of Application, Lecture Notes in Computer Science, 3342, 2005, pp.10-20.

[2] G. Beni, J.wang, Swarm Intelligence, In Proceedings of the Seventh Annual Meeting of the Robotics Society of Japan, 1989, pp.425-428.

[3] G. Beni, From Swarm Intelligence to Swarm robotics, Lecture Notes in Computer Science, 3342, 2005, pp.1-9.

[4] S. Hara, J. Imura, and K. Tsumura, Multi-Resolved Dynamical System Theory for Large Scale Complex Systems, Sice Annual Conference 2008, pp.3115-3118.

[5] S. Hara, T. Hayakawa, and H. Sugata, Stability Analysis of Linear Systems with Generalized Frequency Variables and Its Application to Formation Control, Proceedings of the 46th IEEE Annual Conference on Decision and Control, 2007, pp.1459-1466.

[6] T. Hayakawa, T. Matsuzawa, and S. Hara, Formation Control of Multi-Agent Systems with Sampled Information, Mathematical Engineering Technical Report by Tokyo University, 2006-55, 2006.

[7] G. Lee, N. Y. Chong , A Geometric Approach to Deploying Robot Swarms, Annals of Mathematics and Artificial Intelligence, 52, 2008, pp.257-280.

[8] G. Lee, S. Yoon, N. Y. Chong, and H.Christensen, A Mobile Sensor Network Forming Concentric Circles Through Local Interaction and Consensus Building, Journal of Robotics & Mechatronics, 21(4), 2009, pp.469-473.

[9] X. Defago, S. Souissi, Non-uniform Circle Formation Algorithm for Oblivious Mobile Robots with Convergence toward Uniformity, Theoretical Computer Science, 396, 2008, pp.97-112.

[10] G. Lee, N. Y. Chong, Adaptive Flocking of Robot Swarms: Algorithms and Properties, IEICE Transactions of Communication, E91-B(9), 2008, pp.2848-2854.

[11] V. Jacobson, Congestion Avoidance and Control, Annual Conference of the ACM Special Interest Group on Data Communication ’88, pp.314-329.

[12] J. Postel, User Datagram Protocol, Technical Report Request for Comments 768 Internet Engineering Task Force, 1980.

[13] B. A. McElhoe, An Assessment of the Navigation and Course Corrections for a Manned Flyby of Mars and Venus, IEEE Transactions of Aerospace and Electric Systems, 2(4), 1966, pp.613-623.

[14] G. L. Smith, S. F. Schmidt and L. A. McGee, Application of Statistical Filter Theory to the Optimal Estimation of Position and Velocity on Board a Circumlunar Vehicle, National Aeronautics and Space Administration Technical Documents, 1962.

[15] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation, IEE Proceedings F, 140(2), 1993, pp.107-113.

[16] G. Kitagawa, Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models, Journal of Computational and Graphical Statistics, 5(1), 1996, pp.1-25.

[17] B. A. Sadjadi, Stability of Networked Control Systems in the Presense of Packet Losses, Proceedings of 42th IEEE Annual Conference on Decision and Control, 2003, pp.676-681.

[18] O. C. Imer, S. Yuksel, and T. Basar, Optimal Control of LTI Systems under Unreliable Communication Links, Automatica, 42, 2006, pp.1429-1439.

[19] A. Ohori, K. Kogiso, and K. Sugimoto, Design of a Networked Control System with Unreliable Communication Links via Switching and a Common Lyapunov Function, the 7th SICE Conference on Control Systems, 2007 (in Japanese).

[20] T. Katayama, Non-linear Kalman Filter, Asakura Publshing, 2011 (in Japanese).

[21] W. W. Peterson, D. T. Brown, Cyclic Codes for Error Detection, Proceedings of the Institute of Radio Engineers, 49, 1961, pp.228-235.