論文93-30A-12-3

표본화 벡터 개념을 이용한 분산 표본 혼화기의 간단한 구현

(Simple Realizations of Distributed Sample Scramblers Using the Concept of Sampling Vectors)

金 錫 昌*, 李 秉 基*

(Seok Chang Kim and Byeong Gi Lee)

要 約

본 논문에서는 표본화 벡터 개념을 도입하여 분산 표본 혼화기를 간단하게 구현하는 방법을 제시한다. 분산 표본 혼화기에서 혼화기 상태 표본값이 취해지는 표본화 시간과 이것이 역혼화기로 전달되는 표본 전송 시간이 일치하지 않으면 표본값이 지연되어 전송되며, 이 경우 표본값을 저장하는 부가적인 메모리와 표본 전송 시간을 알려주는 부가적인 클럭이 필요하게 되는 등 분산 표본 혼화기를 복잡하게 만든다. 만일 표본화 벡터 개념을 도입하게 되면, 지연되어 전송되는 표본값의 표본화 시간을 표본값 전송 시간과 일치시킬 수 있고, 결과적으로 부가적인 메모리와 클럭을 없앨 수 있다. 본 논문에서는, 표본화 벡터가 도입된 분산 표본 혼화기를 위한 혼화기와 역혼화기의 동기화 조건을 구하고, 이를 쎌 기반 ATM 분산 표본 혼화기에 적용하여 부가적인 메모리나 클럭의 필요 없이 간단하게 구현할 수 있음을 보인다.

Abstract

In this paper, the concept of sampling vectors is introduced, and used for a simple realization of DSSs(distributed sample scramblers). In DSSs, if the sampling times of the scrambler state samples are not identical to their transmission times, samples are delayed-transmitted to the descrambler, and in this case the DSSs need additional memory elements storing the samples and additional clocks for informing their transmission times. The concept of sampling vectors helps move the sampling times of delayed samples to their transmission times, thus eliminating the additional memory elements and clocks in the DSSs. In the paper, the conditions on the synchronization of the scrambler and descrambler are derived for the DSS employing sampling vectors, and demonstrations are given on their applicaitons to cell-based ATM DSSs.

Ⅰ. 서론

디지탈 전송에서는, 전송 신호를 불규칙하게 해 줌으로써 수신측의 클럭 복원을 용이하게 해 주는 혼화 과정이 필요하다. 기존의 PDH(Plesiochronous Digital Hierarchy) 전송 시스템에서는 자기 동기식 혼화가 주로 사용되었고[1] [2], SDH(Synchronous Digital Hierarchy) 전송 시스템에서는 프레임 동기식 혼화가 사용되고 있다.[3] [4] 더우기, 최근 BIDSN을 위한 ATM (Asynchronous Transfer

*正會員, 서울大學校 電子工學科
(Dept. of Elec. Eng., Seoul Nat'l Univ.)
接受日字 : 1993年 5月 3日

Mode) 전송 시스템을 위해서 분산 표본 혼화(DSS : distributed sample scrambling)가 도입되었다.[5]

분산 표본 혼화는 프레임 동기식 혼화와 기본적으로 동일한 혼화 및 역혼화 방식을 사용한다. 즉, 혼화기에서는 혼화하고자 하는 입력 데이타 수열에 SRG(shift register generator) 수열을 더해 줌으로써 입력 수열을 혼화시키고, 역혼화기에서는 동일한 SRG 수열을 혼화된 입력 수열에 다시 더해 줌으로써 입력 수열을 복원시킨다.[1]

그러나, 분산 표본 혼화는 프레임 동기식 혼화와는 다른 혼화기와 역혼화기의 동기화 방법을 사용함으로써 혼화 효과를 증대시킨다. 프레임 동기식 혼화에서는 각 프레임의 출발점에서 혼화기와 역혼화기를 동일한 상태로 만듦으로써 혼화기와 역혼화기와의 동기화를 이루지만, 분산 표본 혼화에서는 혼화기의 상태를 알려주는 표본값을 역혼화기로 보내주고 역혼화기에서는 이를 이용하여 혼화기와 동기화를 이룬다.

분산 표본 혼화는 짧은 길이의 프레임(또는 패킷) 열로 구성되어 있는 전송 신호의 혼화에 적합한 방식이다. BISDN의 셀 기반 ATM 전송에서는, 전송 신호가 53 바이트로 구성된 ATM 셀 열로 구성된다.[6] 만약, 프레임 동기식 혼화가 ATM 셀 열의 혼화에 사용된다면, 프레임의 길이가 짧아서 혼화 효과가 만족스럽지 못하다. 또한 자기 동기식 혼화는 오류를 증대시키는 문제점이 있다.[2] 그러나 분산 표본 혼화를 사용하면, 이러한 문제들이 해결된다.

분산 표본 혼화에서는 혼화기와 역혼화기의 동기화를 위해서 혼화기 SRG의 표본값이 역혼화기로 전송된다. 이 때, 표본값이 취해지는 표본 시간과 표본값이 전송되는 전송 시간이 일치하지 않으면 표본값이 지연되어 전송되며, 이 경우 표본값을 저장하는 메모리와 표본값 전송 시간을 알려 주는 부가적인 클럭이 필요하게 된다. 실제로, ATM 셀 혼화를 위한 분산 표본 혼화의 경우, 균일한 시간 간격으로 취해진 표본값들이 인접한 두 개의 헤더 오류 제어 구간(HEC field)에서 전송되기에 부가적인 메모리와 클럭이 필요하다.[5]

본 논문에서는, 이러한 표본값 지연 전송 문제를 해결하기 위해서 표본화 벡터를 도입하고, 이를 이용하여 부가적인 클럭이나 메모리의 필요없이 분산 표본 혼화기를 간단히 구현하는 방법을 제시하고자 한다. 표본화 벡터를 도입하게 되면, 지연되어 전송되는 표본값의 표본화 시간을 표본값 전송 시간으로 옮길 수 있게 되고, 결국 표본값 지연 전송을 없앰으로써 부가적인 클럭이나 메모리가 필요 없게 된다. 표본화 벡터가 도입된 분산 표본 혼화기의 표본화 조건과 정정 조건을 구하고[3], 이를 이용하여 ATM 셀 혼화를 위한 분산 표본 혼화기를 간단히 구현하도록 하겠다.

Ⅱ. 표본화 벡터를 도입한 분산 표본 혼화기의 동작 및 모델링

본 절에서는 표본화 벡터를 도입한 분산 표본 혼화기의 동작을 설명하고, 이를 수학적으로 모델링하도록 한다.

1. 동작

표본화 벡터를 도입한 분산 표본 혼화기의 기능 블럭 구성도는 그림 1과 같다. 혼화기는 SRG 엔진 기능 블럭, 수열 발생 기능 블럭, 표본화 기능 블럭 그리고 혼화 기능 블럭으로 구성되며, 역혼화기는 혼화기에 있는 기능 블럭에서 혼화 기능 블럭이 역혼화 기능 블럭으로 바뀌고, 여기에다가 동기화를 위한 비교 기능 블럭과 정정 기능 블럭이 추가된다.

다수의 시프트 레지스터와 배타적 논리합 소자로 구성되어 있는 SRG 엔진 기능 블럭은 각 시프트 레지스터로 하여금 의사 불규칙 이진 수열을 발생시키도록 한다. 이 의사 불규칙 수열들은 수열 발생 기능 블럭에서 선형적으로 결합되어 혼화에 사용되는 SRG 수열 $\{s_k\}$가 발생되며, 또한 표본화 기능 블럭에서 선형적이지만 시간 가변적으로 결합되어 표본 수열 $\{z_i\}$가 발생된다.[4]

혼화기 입력 데이타 수열 $\{b_k\}$는 혼화기 SRG 수열 $\{s_k\}$가 더해짐으로써 혼화되며, 마찬가지로 혼화된 입력 데이타 수열 $\{b_k+s_k\}$는 역혼화기 SRG 수열 $\{\hat{s}_k\}$가 더해짐으로써 역혼화된다.[5] 역혼화된 수열 $\{s_k+b_k+\hat{s}_k\}$가 원래의 입력 데이타 수열 $\{b_k\}$와 동일한 것이 되기 위해서는, 역혼화기 SRG의 상태가 혼화기의 SRG 상태와 동일하게 되어야 한다. 이러한 혼화기와 역혼화기의 동기화를 위해서, 혼화기는 표본화 기능 블럭에서 발생시킨 표본 수열 $\{z_i\}$를 역혼화기로 전달하며, 역혼화기에서는 혼화기와 동일한 방법으로 발생시킨 자신의 표본 수열 $\{\hat{z}_i\}$를 전달된 혼화기 표본 수열 $\{z_i\}$와 비교한다. 그림 1의 윗쪽에 있는 배타적 논리합 소자는 이러한 비교 기능을 수행하는 것으로서, 두 표본값 z_i와 $\hat{z}_i$가 서로 다를 경우에만 정정 기능 블럭을 동작시킨다. 정정 기능 블럭은 역혼화기 SRG 상태를 정정하여 궁극적으로는 혼화기 SRG 상태와 동일하게 만든다. 여기서 혼화기와 역혼화기에 공통적으로 있는 SRG 엔진 기능블럭, 수열 발생 기능 블

력 및 표본화 기능 블럭은 서로 동일하다.

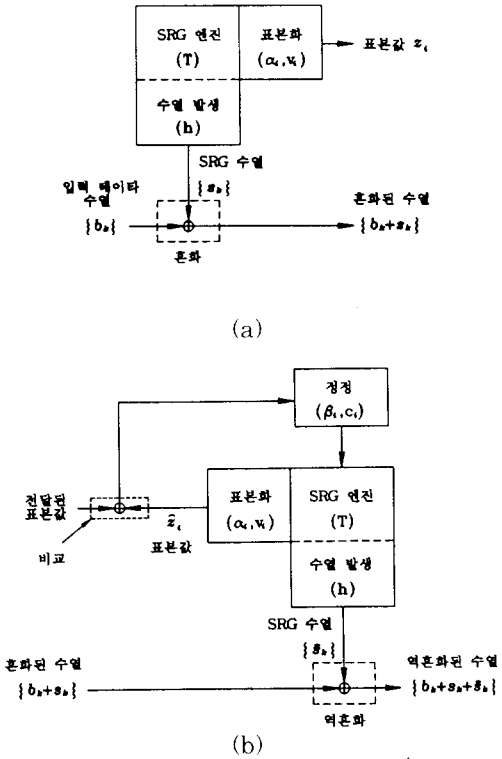

그림 1. 표본화 벡터를 도입한 분산 표본 혼화기의 기능 블럭 구성도
(a)혼화기 (b) 역혼화기

Fig. 1. Functional blockdiagram of the DSS employing sampling vectors.
(a) Scrambler, (b) descrambler.

2. 각 기능 블럭의 모델링

L개의 시프트 레지스터로 구성되어 있는 SRG엔진 기능 블럭에 대해서, 상태 벡터 d_k(역 혼화기의 경우에는 $\hat{d}_k$로 표기)를 시프트 레지스터들의 시간 k에서의 상태를 나타내는 L-벡터로 정의하고, 상태 천이 행렬 $\mathbf{T}$를 인접한 시간의 두 상태 벡터 d_k(또는 $\hat{d}_k$)와 d_{k-1}(또는 $\hat{d}_{k-1}$)와의 관계를 나타내는 $L\times L$행렬로 정의한다. 즉,

$$d_k = T\cdot d_{k-1} \quad (\text{또는} \hat{d}_k = T\cdot \hat{d}_{k-1}). \tag{1}$$

수열 발생 기능 블럭에 대해서, 발생 벡터 h를 SRG 수열 원소 s_k(또는 $\hat{s}_k$)와 상태 벡터 d_k(또는 $\hat{d}_k$)와의 관계를 나타내는 L-벡터로 정의한다. 즉,

$$s_k = h^t\cdot d_k \quad (\text{또는} \ \hat{s}_k = h^t\cdot \hat{d}_k) \tag{2}$$

SRG 엔진이 L개의 시프트 레지스터로 구성된 경우, L보다 적은 갯수의 표본값으로 L개 의 시프트 레지스터 상태값들을 예측한다는 것은 불가능하다. 즉, L개 혹은 그 이상의 혼화기 SRG 상태 표본값이 역혼화기에 전송되어야만 역혼화기 SRG를 혼화기 SRG에 동기화시킬 수 있다. 그러므로, 효율적인 동기화 과정을 위해서 혼화기에서는 L번의 표본화 과정이 있다고 가정하고, 또한 역혼화기에서는 L번의 정정 과정이 있다고 가정한다.

표본화 기능 블럭에 대해서, i번째 표본화 시간 α_i, $i=0,1,\cdots,L-1$를 i번째 표본값 z_i(또는 z_i)가 표본화되는 시간으로 정의하고[6], i번째 표본화벡터v_i, $i=0,1,\ldots,L-1$를 i번째 표본화 시간 α_i에서의 상태 벡터 $d_{\alpha i}$(또는 $\hat{d}_{\alpha i}$)로부터 어떻게 표본화되는가를 나타내는 L-벡터로 정의한다.[7] 그러면,i번째 표본값 z_i와 $\hat{z}_i$는

$$\begin{cases} z_i = v_i^t\cdot d_{\alpha_i}, \ i=0,1,\ldots,L-1, \\ \hat{z}_i = v_i^t\cdot \hat{d}_{\alpha_i}, \ i=0,1,\ldots,L-1 \end{cases} \tag{3}$$

로 표현될 수 있다.

역혼화기의 정정 기능 블럭에 대해서, i번째 정정 시간 β_i, $i=0,1,\ldots,L-1$를 SRG 상태의 i번째 표본값 z_i(또는 $\hat{z}_i$)를 이용하여 역혼화기 SRG 상태를 정정하는 시간으로 정의하고, i번째 정정 벡터 c_i,$i=0,1,\ldots,L-1$를 i번째 정정 시간 β_i에 정정되는 시프트 레지스터의 위치를 나타내는 L-벡터로 정의한다.[8]

표본화 과정과 정정 과정을 시간축에서 나타내면, 그림 2와 같이 그려질 수 있다. 여기서 i번째 정정은 i번째 표본화 후에 일어나지만$(i+1)$번째 표본화보다 늦을 수 없음에 유의해야 한다. 즉 $\alpha_0<\beta_0\le\alpha_1<\beta_1\le\cdots\le\alpha_{L-1}<\beta_{L-1}$이다.

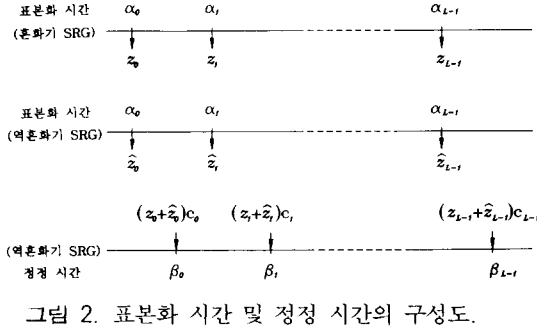

그림 2. 표본화 시간 및 정정 시간의 구성도.

Fig. 2. Timing diagram for sampling and correction times.

Ⅲ. 동기화 조건

본 절에서는 앞 절의 모델링으로부터 혼화기와 역혼화기의 동기화 문제를 수학적 문제로 변화시키고, 이를 풀어서 동기화 조건을 이끌어내도록 한다.

1. 동기화 문제

동기화 문제를 수학적으로 표현하기 위해서, 상태 거리 벡터 δ_k를 혼화기 상태 벡터 d_k와 역혼화기의 상태 벡터 $\hat{d}_k$의 차이를 나타내는 L-벡터, 즉

$$\delta_k \equiv d_k + \hat{d}_k \tag{4}$$

로 정의하자. 그러면, 혼화기와 역혼화기가 동기화되었다는 것은 혼화기 SRG의 상태 벡터 d_k와 역혼화기 SRG 상태 벡터 $\hat{d}_k$가 동일하게 되는 것을 의미하므로, 이것은 상태 거리 벡터 δ_k가 0 이라는 것과 동치가 된다.

혼화기 SRG 상태 벡터들은 식 (1)에 의해서

$$d_{\alpha_i} = \begin{cases} T^{\alpha_0}\cdot d_0, & i=0, \\ T^{\alpha_i-\beta_{i-1}}\cdot d_{\beta_{i-1}}, & i=1,2,\ldots,L-1, \end{cases} \tag{5a}$$

$$d_{\beta_i} = \begin{cases} T^{\beta_0}\cdot d_0, & i=0, \\ T^{\beta_i-\beta_{i-1}}\cdot d_{\beta_{i-1}}, & i=1,2,\ldots,L-1, \end{cases} \tag{5b}$$

와 같은 관계를 가지고 : 역혼화기 SRG에 대해서는 정정 시간 β에만 정정이 일어나고, 정정 과정은 상태 벡터에 $(z_i+\hat{z}_i)\cdot c_i$를 더해주는 것과 동일하다는 것에 유의하면

$$\hat{d}_{\alpha_i} = \begin{cases} T^{\alpha_0}\cdot \hat{d}_0, & i=0, \\ T^{\alpha_i-\beta_{i-1}}\cdot \hat{d}_{\beta_{i-1}}, & i=1,2,\ldots,L-1, \end{cases} \tag{6a}$$

$$\hat{d}_{\beta_i} = \begin{cases} T^{\beta_0}\cdot \hat{d}_0 + (z_0+\hat{z}_0)c_0, & i=0, \\ T^{\beta_i-\beta_{i-1}}\cdot \hat{d}_{\beta_{i-1}} + (z_i+\hat{z}_i)c_i, & i=1,2,\ldots,L-1, \end{cases} \tag{6b}$$

와 같은 관계식을 얻을 수 있다.

식 (3), (4), (5a) 그리고 (6a)에 의해서

$$(z_i+\hat{z}_i) = \begin{cases} v_0^t\cdot T^{\alpha_0}\cdot \delta_0, & i=0 \\ v_i^t\cdot T^{\alpha_i-\beta_{i-1}}\cdot \delta_{\alpha_i}, & i=1,2,\ldots,L-1 \end{cases}$$

를 얻을 수 있고, 이 식과 식 (4), (5b), (6b)을 이용하면 관계식

$$\delta_{\beta_i} = \begin{cases} T^{\beta_0}\cdot\delta_0 + (v_0^t\cdot T^{\alpha_0}\cdot\delta_0)c_0, & i=0 \\ T^{\beta_i-\beta_{i-1}}\cdot\delta_{\beta_{i-1}} (v_i^t\cdot T^{\alpha_i-\beta_{i-1}}\cdot\delta_{\beta_{i-1}})c_i, & i=1,2,\ldots,L-1 \end{cases}$$

를 도출해 낼 수 있다. 이 식의 둘째항에 있는 두 상수를 바꾸어 곱하고 재정리하면

$$\delta_{\beta_i} = \begin{cases} (T^{\beta_0}+c_0\cdot v_0^t\cdot T^{\alpha_0})\cdot\delta_0, & i=0 \\ (T^{\beta_i-\beta_{i-1}}+c_i\cdot v_i^t\cdot T^{\alpha_i-\beta_{i-1}})\cdot\delta_{\beta_{i-1}}, & i=1,2,\ldots,L-1 \end{cases} \tag{7}$$

와 같이 된다. 그러므로 최종적으로 정정된 상태 거리 벡터 $\delta_{\beta L-1}$와 초기의 상태 거리 벡터 δ_0와의 관계식은

$$\delta_{\beta_{L-1}} = \Lambda\cdot\delta_0 \tag{8}$$

로 표현될 수 있으며, 여기서 정정 행렬 Λ는 다음과 같이 정의되는 $L\times L$행렬이다.

$$\Lambda \equiv \left(T^{\beta_{L-1}-\beta_{L-2}}+c_{L-1}\cdot v_{L-1}^t\cdot T^{\alpha_{L-1}-\beta_{L-2}}\right)\cdot\left(T^{\beta_{L-2}-\beta_{L-3}}+c_{L-2}\cdot v_{L-2}^t\cdot T^{\alpha_{L-2}-\beta_{L-3}}\right) \cdots\left(T^{\beta_1-\beta_0}+c_1\cdot v_1^t\cdot T^{\alpha_1-\beta_0}\right)\cdot\left(T^{\beta_0}+c_0\cdot v_0^t\cdot T^{\alpha_0}\right) \tag{9}$$

앞서 언급한 것처럼, 역혼화기 SRG를 혼화기 SRG에 동기화시키는 것은 초기의 상태 거리 벡터 δ_0와는 무관하게 최종적으로 정정된 상태 거리 벡터 $\delta_{\beta L-1}$를 0으로 만드는 것을 의미한다. 그러나 이것은 정정 행렬 Λ가 0 행렬인 경우에만 가능하므로, 동기화 문제는 식 (9)에 있는 정정 행렬 Λ를 0 행렬로 만드는 α_i, v_i, β그리고 c들을 구하는 수학적 문제와 동일하게 된다.

2. 동기화 조건

먼저 정정 행렬 Λ를 0 행렬로 만들 수 있는 표본화 시간 α_i들 및 표본화 벡터 v_i들의 조건을 구하기 위해서 판별 행렬 Δ를 다음과 같은 $L\times L$행렬로 정의한다.

$$\Delta \equiv \begin{bmatrix} v_0^t\cdot T^{\alpha_0} \\ v_1^t\cdot T^{\alpha_2} \\ v_2^t\cdot T^{\alpha_3} \\ \vdots \\ v_{L-1}^t\cdot T^{\alpha_L} \end{bmatrix}. \tag{10}$$

그러면, 다음과 같은 표본화 조건 정리를 얻을 수 있다.

정리 1(표본화 조건). 가역 상태 천이 행렬 $\mathbf{T}$에 대해서[9], 식 (10)에 있는 판별 행렬 Δ가 비가역이 되도록 표본화 시간 α_i들과 표본화 벡터 v_i들을 선택한다면, 어떠한 정정 시간 β들과 정정 벡터 c_i들에 대해서도 식 (9)의 정정 행렬 Λ는 0 행렬이 될 수 없다.

(증명) 귀류법을 사용하여 위의 정리를 증명하겠다. 판별 행렬 Δ가 비가역이 되도록 선택한 표본화 시간 α_i, $i=0,1,\cdots,L-1$와 표본화 벡터 v_i, $i=0,1,\cdots,L-$

1에 대해서, 어떤 정정 시간 β_i, $i=0,1,\cdots,L-1$와 정정 벡터 c_i, $i=0,1,\cdots L-1$가 정정 행렬 Λ를 0 행렬로 만든다고 가정하자. 그러면, $\triangle\cdot\hat{\delta}=0$ 이 되는 0이 아닌 L-벡터 $\hat{\delta}$가 존재하고, 이러한 $\hat{\delta}$에 대해서는, 식 (10)으로부터 관계식 $v_i^t\cdot T^{\alpha_i}\cdot\hat{\delta}=0,\ 1,\ \cdots,\ L-1$를 얻는다. 이들과 식 (9)를 $\Lambda\cdot\hat{\delta}=T^{\beta_{L-1}}\cdot\hat{\delta}$에 반복적으로 적용하면, 최종적으로 $\Lambda\cdot\hat{\delta}$를 얻을 수 있고, 또한 T가 가역이므로 $\delta=T^{-\beta_{L-1}}\cdot\Lambda\cdot\hat{\delta}$가 된다. 그러나, Λ가 0 행렬이므로 $\hat{\delta}$는 0 벡터가 되고, 이것은 처음의 가정 $\delta\neq 0$ 에 모순이다. (증명끝)

이 정리는 식 (10)에 있는 판별 행렬 $\triangle$가 비가역이 되도록 α_i들과 v_i들이 선택한다면, 식 (9)에 있는 정정 행렬 Λ를 0 행렬로 만드는 β들과 c_i들이 존재하지 않는다는 것을 의미한다. 그러므로, 역혼화기 SRG를 혼화기 SRG에 동기화시키려면, 식 (10)에 있는 판별 행렬 $\triangle$가 가역이 되도록 표본화 시간 α_i들과 표본화 벡터 v_i들을 선택하여야 한다.

이제, 판별 행렬 $\triangle$가 가역이 되도록 표본화 시간 α_i들과 표본화 벡터 v_i들을 선택했다고 가정하고, 정정 행렬 Λ를 0 행렬로 만드는 정정 시간 β들과 정정 벡터 c_i들를 어떻게 결정할 수 있는지를 알아 본다. 다음의 정리가 이것을 보여 준다.

정리 2(정정 조건). 가역 상태 천이 행렬 T에 대해서, 식 (10)에 있는 판별 행렬 $\triangle$가 가역이 되도록 표본화 시간 α_i들과 표본화 벡터 v_i들이 선택했다고 하자. 그러면, 정정 시간 β들과 정정벡터 c_i들을 다음과 같이 선택했을 때에만, 식 (9)의 정정 행렬 Λ가 0 행렬이 된다.

$$c_i=\begin{cases}T^{\beta_i}\cdot\Delta^{-1}\cdot\left(e_i+\sum_{j=i+1}^{L-1}u_{i,j}e_j\right), & i=0,1,\cdots,L-2l,\\ T^{\beta_{L-1}}\cdot\Delta^{-1}\cdot e_{L-1}, & i=L-1.\end{cases}\tag{11}$$

여기서 L-벡터 e_i, $i=0,1,\cdots,L-1$는 i번째 원소만 1이고 나머지는 모두 0인 기저 벡터이고, u_{ij} $i=0,1,\cdots L-2$, $j=i+1,i+2,\cdots,L-1$는 0 또는 1인 임의의 값이다.

(증명) 위의 정리를 증명하기 위해서는 다음과 같은 두 개의 보조 정리를 필요로 한다. 이 보조 정리들의 증명은 부록에 있다.

보조 정리 1. 가역 판별 행렬 $\triangle$에 대해서,

$$v_i^t\cdot T^{\alpha_i}\cdot\Delta^{-1}=e_i^t,\ i=0,1,\cdots,L-1.\tag{12}$$

보조 정리 2. $L\times L$행렬 Λ_i, $i=1,2,\cdots,L-1$를

$$\Lambda_i\equiv\left(T^{\beta_{i-1}-\beta_{i-2}}+c_{i-1}\cdot v_{i-1}^t\cdot T^{\alpha_{i-1}-\beta_{i-2}}\right)\cdots\left(T^{\beta_i-\beta_{i-1}}+c_i\cdot v_i^t\cdot T^{\alpha_i-\beta_{i-1}}\right)\tag{13}$$

로 정의하면, 다음과 같은 관계식들이 성립한다.

$$\begin{cases}\Lambda=\Lambda_1\cdot\left(T^{\beta_0}+c_0\cdot v_0^t\cdot T^{\alpha_0}\right),\\ \Lambda_i=\Lambda_{i+1}\cdot\left(T^{\beta_i-\beta_{i-1}}+c_i\cdot v_i^t\cdot T^{\alpha_i-\beta_{i-1}}\right),i=1,2,\cdots,L-2,\end{cases}\tag{14a}$$

$$\Lambda_i\cdot T^{\beta_{i-1}}\cdot\Delta^{-1}\cdot e_j=\begin{cases}T^{\beta_{i-1}}\cdot\Delta^{-1}\cdot e_j, & 0\le j<i\le L-1,\\ \Lambda\cdot\Delta^{-1}\cdot e_j, & 1\le i<j\le L-1,\end{cases}\tag{14b}$$

$$\Lambda\cdot\Delta^{-1}\cdot e_j=\begin{cases}\Lambda_{i+1}\cdot\left(T^{\beta_{i-1}}\cdot\Delta^{-1}\cdot e_j+c_i\right), & i=0,1,\cdots,L-2,\\ T^{\beta_{i-1}}\cdot\Delta^{-1}\cdot e_{L-1}+c_{L-1}, & i=L-1.\end{cases}\tag{14c}$$

먼저, 식 (11)으로 표현되는 c_i, $i=0,1,\ldots,L-1$에 대해서 Λ가 0이 됨을 보이겠다. 그런데, 이것은 $\Lambda\cdot\triangle^{-1}\cdot e_j=0$, $i=0,1,\cdots,L-1$과 동치이므로, 귀납법을 이용하여 후자를 증명하겠다. $i=L-1$인 경우에는, 식 (14c)로부터

$$\Lambda\cdot\Lambda^{-1}\cdot e_{L-1}=T^{\beta_{L-1}}\cdot\Delta^{-1}\cdot e_{L-1}+c_{L-1}$$

를 얻고, 여기에 식 (11)을 대입하면

$$\Lambda\cdot\Lambda^{-1}\cdot e_{L-1}=c_{L-1}+c_{L-1}=0$$

이 된다. 이제 $j=L-1,\ L-2,\ \ldots,i$에 대해서 $\Lambda\cdot\triangle^{-1}\cdot e_j=0$이라고 가정하고, $\Lambda\cdot\triangle^{-1}\cdot e_{i-1}=0$ 이 됨을 보이겠다. 식 (14c)로부터

$$\Lambda\cdot\Delta^{-1}\cdot e_{i-1}=\Lambda_i\cdot(T^{\beta_{i-1}}\cdot\Delta^{-1}\cdot e_{i-1}+c_{i-1})$$

를 얻고, 여기에 식 (11)을 대입하면

$$\Lambda\cdot\Delta^{-1}\cdot e_{i-1}=\sum_{j=i}^{L-1}u_{i-1,j}\Lambda_i\cdot T^{\beta_{i-1}}\cdot\Delta^{-1}\cdot e_j$$

가 된다. 또한 이 식은 식 (14b)에 의해서

$$\Lambda\cdot\Delta^{-1}\cdot e_{i-1}=\sum_{j=i}^{L-1}u_{i-1,j}\Lambda_i\cdot\Delta^{-1}\cdot e_j$$

로 변형될 수 있다. 그러나,

$$\Lambda\cdot\Delta^{-1}\cdot e_{i-1}=0,j=i,i+1,\cdots,L-1$$

이라고 가정하였으므로,

$$\Lambda\cdot\Delta^{-1}\cdot e_{i-1}=0$$

이 된다.

반대로 $\Lambda = 0$ 이면, c_i, $i=0,1,\cdots,L-1$가 식 (11)과 같이 됨을 보이겠다. T와 △가 가역이므로, c_i는 기저 벡터들 e_j, $j=0,1,\cdots,L-1$에 대해서 다음과 같이 유일하게 표현될 수 있다.

$$c_i = T^{\beta_i}\cdot\Delta^{-1}\cdot\sum_{j=0}^{L-1} v_{i,j}\cdot e_j,\quad i=0,1,\cdots,L-1. \tag{15}$$

그러므로, $\Lambda = 0$ 일 때, $v_{i,j}=1$, $v_{i,j}=0$, $i=0,1,\ldots,L-1$, $j=0,1,\cdots,i-1$임을 보이면 충분하다.

$i=0,1,\cdots,L-2$인 경우에 대해서는, 식 (15)를 식 (14c)에 대입하면

$$\Lambda\cdot\Delta^{-1}e_i = \Lambda_i\cdot(T^{\beta_i}\cdot\Delta^{-1}\cdot e_i + \sum_{j=0}^{L-1} v_{i,j}e_j)$$

를 얻고, 이것은 식 (14b)에 의해서

$$\Lambda\cdot\Delta^{-1}e_i = \Lambda_i\cdot\sum_{j=0}^{i-1} v_{i,j}T^{\beta_{L-1}}\cdot\Delta^{-1}\cdot e_i + (1+v_{i,i})T^{\beta_{L-1}}\cdot\Delta^{-1}\cdot e_i + \sum_{j=i+1}^{L-1} v_{i,j}\Lambda\cdot\Delta^{-1}\cdot e_i$$

가 된다. 그런데 $\Lambda=0$에 의해서

$$T^{\beta_{L-1}}\cdot\Delta^{-1}\cdot\left\{\sum_{j=0}^{i-1} v_{i,j}e_j + (1+v_{i,i})e_i\right\} = 0$$

가 되므로, $v_{i,i}=1$ 이고, $v_{i,j}=0$, $j=0,1,\cdots,i-1$이다.

$i=L-1$인 경우에는, 식 (14c)로부터

$$c_{L-1} = T^{\beta_{L-1}}\cdot\Delta^{-1}\cdot e_{L-1}$$

를 얻는다. 그러므로 $v_{L-1,L-1}=1$이고, $v_{L-1,j}=0$, $j=0,1,\ldots L-2$이다. (증명끝)

이 정리는 식 (10)에 있는 판별 행렬 △가 가역이 되도록 α_i들과 v_i들을 선택했을 경우, 임의로 선택된 β들에 대해서 식 (11)에 있는 c들은 식 (9)에 있는 정정 행렬 Λ를 0 행렬로 만든다는 것을 의미한다. 그러므로, 일단 판별 행렬 △가 가역이 되도록 표본화 시간 α_i들과 표본화 벡터 v_i들을 선택했다면, 정정 시간 β들은 임의로 선택할 수 있고, 정정 벡터 c들은 식 (11)에 α_{L-1}따라 결정할 수 있다. 식 (11)로부터 정정 벡터 c_{L-1}은 유일하게 존재하지만, 정정 벡터 c_{L-1-i}, $i=1,2,\ldots,L-1$는 u_{ij}의 선택에 따라서 2개만큼 존재함을 알 수 있다.

Ⅳ. ATM 셀 혼화에의 적용

본 절에서는 지금까지의 결과를 이용하여, 셀 기반 ATM 셀 전송에 사용되는 분산 표본 혼화기를 어떻게 간단히 구현할 수 있는지를 보이겠다.

1. 원래의 분산 표본 혼화기

ATM 셀 혼화에 관한 CCITT 문서에 따르면[5] 분산 표본 혼화기는 그림 3과 같은 SRG를 사용하도록 권고하고 있다. 이 SRG는 31 개의 시프트 레지스터로 구성되어 있으며, 그것의 상태 천이 행렬 T와 발생 벡터 h는 다음과 같다.

$$T = [t_{i,j}]_{31\times 31},\ t_{i,j} = \begin{cases} 1, & i=0,1,\cdots,29,\ j=i+1, \\ 1, & i=30,\ j=0 \quad 3, \\ 0, & \text{그 밖의 경우} \end{cases} \tag{16}$$

$$h = e_0 + e_3. \tag{17}$$

그림 3. 셀 기반 ATM의 분산 표본 혼화에 사용되는 SRG

Fig. 3. The SRG employed for the DSS of the cell-based ATM.

31 개의 혼화기 SRG 상태 표본값 $z_0 \sim z_{30}$은 SRG 수열 $\{s_k\}$ 로부터 취해지며, 그들의 표본화 시간은 그림 4a(실선 화살표)와 같다. 이것은 모든 표본화 벡터 $v_0 \sim v_{30}$가 식 (17)에 있는 발생 벡터 h와 동일하고, 표본화 시간은 $\alpha_0=0$, $\alpha_1=212$, $\alpha_2=2\times212$, $\cdots$, $\alpha_{30}=30\times212$임을 의미한다.[10]

이러한 표본화 벡터 및 표본화 시간들과 식 (16)을 식 (10)에 대입하면, 판별 행렬 △가 가역임을 쉽게 확인할 수 있다.

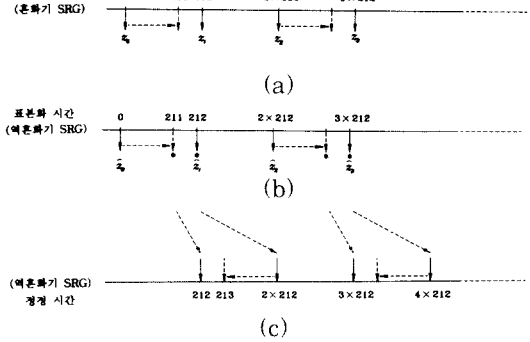

그림 4. 셀 기반 ATM의 분산 표본 혼화를 위한 표본화 시간 및 정정 시간의 구성도

Fig. 4. Timing diagram for the sampling and correction times employed in the DSS of the cell-based ATM.

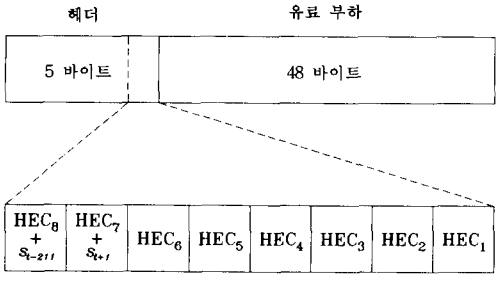

그림 5. 혼화기 SRG 표본값 전달을 위한 ATM 쎌 내부의 데이타 구조

Fig. 5. Data structure of ATM cell for the scrambler SRG sample conveyance.

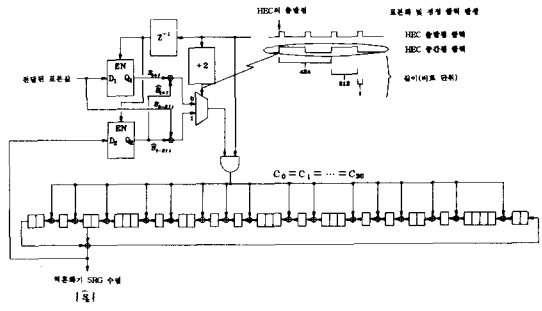

그림 6. ATM 쎌 혼화를 위한 원래의 역혼화기

Fig. 6. The original descrambler for the ATM-cell scrambling(Figure C-/I. 432).

혼화기 표본값 $z_0 \sim z_{30}$은 그림 5가 보여주는 것처럼 한 개의 ATM 쎌 당 두 개씩 역혼화기로 전달된다. 즉, 53 바이트(424 비트)로 구성되어 있는 ATM 쎌의 헤더 오류 제어 구간의 상위 두 비트가 표본값을 전송해 준다. 따라서, 짝수 첨자를 가지는 표본값들(z_0, z_2등)은 HEC7슬롯(s_{t-211}로 표시되어 있음)을 통해 전송되기 위해서 211 비트만큼의 시간 동안 지연되어져야 하며(그림 4a의 점선 화살표), 반면 홀수 첨자를 가지는 표본값들(z_1, z_3등)은 지연없이 HEC7 (s_{t+1}로 표시되어 있음)을 통해 전송된다.

역혼화기에서는 동일한 방법으로 자기 표본값 z_i들을 발생시켜서, 그림 4b(*로 표시)가 보여주는 것처럼 전달된 표본값 z_i들과 시간 211, 212, 2×212+211, 3×212,…에 비교한다. 비교된 결과는 역혼화기 SRG의 상태를 정정하는데 사용되며, 정정 시간은 그림 4c(실선 화살표)가 보여주는 것처럼 212, 2×212+211, 3×212, 4×212,…이다. 이러한 정정 시간들을 식 (16)의 T와 판별행렬△와 함께 식 (11)에 대입하면, 다음과 같은 정정 벡터를 얻을 수 있다.

$$\tilde{c}_0=\tilde{c}_1=\cdots=\tilde{c}_{30}=e_1+e_2+e_5+e_7+e_8+e_{10}+e_{11}+e_{12}+e_{15}+e_{16}+e_{19}+e_{20}+e_{21}+e_{23}+e_{24}+e_{28}.$$

지금까지 설명한 원래의 분산 표본 혼화기는 표본화 시간과 정정 시간이 균일하다는 점에서 좋은 것처럼 보인다.[11] 하지만, 실제적인 측면에서는 큰 장점이 되지 못하며, 오히려 그림 6의 역혼화기가 보여주는 것처럼 지연되어 전송되는 표본값을 저장하는 부가적인 메모리와, 전송되는 시점을 알려주는 부가적인 클럭이 필요하게 된다.

2. 호환 가능한 간단한 분산 표본 혼화기

원래의 분산 표본 혼화기를 복잡하게 만드는 부가적인 클럭과 메모리는 결국 표본값 지연 전송에 기인하므로, 여기에서는 지연되어 전송되는 표본값의 표본화 시간을 전송 시점으로 옮겨서(그림 4a의 점선 화살표) 표본값 지연 전송을 없애고, 궁극적으로 부가적인 클럭과 메모리가 필요없는 간단한 역혼화기를 어떻게 구현하는지를 보이겠다. 이 때, 그림 5에 있는 표본값은 그대로 보존되게 함으로써, 원래의 분산 표본 혼화기와는 호환이 가능하도록 한다.

이를 위해서, 표본화 시간의 변화에 따른 표본화 벡터의 변화를 알아야 한다. 지연 전송되는 짝수 첨자 s_{t-211}는 식 (2)와 (1)에 의해서 $s_{t-211}=h^t \cdot d^{t-211}=(h^t \cdot T^{-211}) \cdot d_t$로 표현될 수 있다. 이것은 표본값 s_{t-211}의 표본화 시간을 실제 전송 시간 t로 옮길 경우, 표본화 벡터는 $(T^{-211})^t \cdot h$로 변화되어야 함을 의미한다. 즉, 변화된 표본화 벡터는 다음과 같다.

$$\begin{aligned}&=e_0+e_2+e_5+e_6+e_7+e_9+e_{10}+e_{13}+e_{14}+e_{19}+e_{20}+e_{21}+\\&e_{22}+e_{23}+e_{24}+e_{25}+e_{29}+e_{30}\\\tilde{v}_1&=\tilde{v}_3=\cdots\tilde{v}_{29}=h=e_0+e_1.\end{aligned}$$

이렇게 표본화 시간과 표본화 벡터를 변화시켜도, 정정이 지연되어 일어날 경우(그림 4c의 실선 화살표), 마찬가지로 부가적인 클럭과 메모리가 필요하게 된다. 그래서, 정정도 지연없이 일어나도록 정정 시간을 그림 4c의 점선 화살표로 옮긴다. 그러면, 정정 시간은 $\beta_0=212$, $\beta_1=213$, $\beta_2=2\times212+1$, $\beta_3=2\times212+213$, …가 된다. 이러한 표본화 시간, 표본화 벡터 및 정정 시간을 식 (16)과 함께 식 (11)에 대입하면, 다음과 같은 정정 벡터를 얻을 수 있다.

$$\begin{cases}\tilde{c}_0=\tilde{c}_2=\cdots=\tilde{c}_{30}=e_1+e_2+e_4+e_7+e_8+e_{10}+e_{11}+e_{12}+\\\qquad e_{15}+e_{16}+e_{19}+e_{20}+e_{21}+e_{23}+e_{24}+e_{28},\\\tilde{c}_1=\tilde{c}_3=\cdots=\tilde{c}_{29}=e_4+e_5+e_{11}+e_{13}+e_{14}+e_{15}+e_{16}+e_{17}+e_{18}+e_{19}+\\\qquad e_{20}+e_{23}+e_{25}+e_{23}+e_{25}+e_{26}+e_{27}+e_{28}+e_{30}.\end{cases}$$

이것들로부터 그림 7과 같은 역혼화기를 얻을 수 있으며, 그림에 있는 S는 각각의 표본화 시간에 따라

반전되는 스위치를 나타낸다. 그림으로부터 부가적인 클럭과 메모리는 완전히 없어졌으며, 그 대신 두 개의 표본화 벡터와 두 개의 정정 벡터가 필요하게 됨을 알 수 있다. 이 역혼화기는 원래의 역혼화기와 정정 시간은 다르지만, 그림 5에 있는 표본값에는 아무런 영향을 미치지 않으므로 그림 6에 있는 원래의 역혼화기와 호환이 가능함에 주의해야 한다.

3. 호환 불가능하지만 더욱 간단한 분산 표본 혼화기

바로 앞에서 언급한 역혼화기는 원래의 역혼화기와 호환이 가능하지만, 두 개의 표본화 벡터와 두 개의 정정 벡터가 필요하다. 이에 표본화 벡터 또는 정정 벡터를 한 개로 줄임으로써 보다 간단한 역혼화기를 얻을 수 있음을 보이겠다.[12] 물론 이렇게 얻어진 역혼화기는 원래의 역혼화기와는 호환이 불가능함에 유의해야 한다.

표본화 시간(그림 4a의 점선 화살표), 정정 시간(그림 4c의 점선 화살표) 및 홀수 첨자의 표본화 벡터(식 (19b))를 그대로 두고,[13] 짝수 첨자의 표본화 벡터를 식 (17)에 있는 발생 벡터 h와 동일하게 취하면, 식 (11)에 의해서 정정 벡터

$$\begin{cases} \hat{c}_0 = \hat{c}_2 = \cdots = \hat{c}_{30} = e_0 + e_3 + e_5 + e_{13} + e_{14} + e_{15} + e_{18} + \\ \qquad e_{21} + e_{22} + e_{25} + e_{26} + e_{28} + e_{29} + e_{30} \\ \hat{c}_1 = \hat{c}_3 = \cdots = \hat{c}_{29} = e_0 + e_1 + e_2 + e_5 + e_8 + e_{11} + e_{12} + \\ \qquad e_{15} + e_{16} + e_{17} + e_{20} + e_{30} \end{cases}$$

를 얻을 수 있고, 이로부터 그림 8과 같은 역혼화기를 얻는다.

마찬가지로, 다른 것은 그대로 두고 짝수 첨자의 표본화 벡터를

$$\bar{v}_0 = \bar{v}_2 = \cdots = \bar{v}_{30} = e_1 + e_2 + e_4 + e_6 + e_8 + e_{11} + e_{13} + e_{14} + e_{18} + e_{19} + e_{21} + e_{23} + e_{24} + e_{25} + e_{26} + e_{28} + e_{29} + e_{30}$$

로 변화시키면, 식 (11)로부터 공통의 정정 벡터

$$\bar{c}_0 = \bar{c}_1 \cdots = \bar{c}_{30} = e_0 + e_3 + e_5 + e_{13} + e_{14} + e_{15} + e_{18} + e_{21} + e_{22} + e_{25} + e_{26} + e_{28} + e_{29} + e_{30}$$

를 얻을 수 있고, 이로부터 그림 9와 같은 역혼화기를 얻는다.

그림 8은 한개의 표본화 벡터를 갖는 역혼화기이고, 그림 9는 한 개의 정정 벡터를 갖는 역혼화기이며, 이들은 그림 6이나 7의 역혼화기에 비해 더욱 간단해졌음을 알 수 있다.

V. 결론

본 논문에서는, 분산 표본 혼화에 표본화 벡터를 도입하고, 이를 이용하여 부가적인 클럭이나 메모리

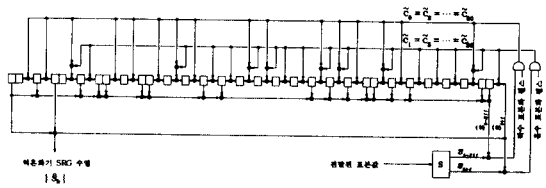

그림 7. 간단하면서도 호환가능한 역혼화기

Fig. 7. simple and compatible descrambler for the ATM-cell scrambling.

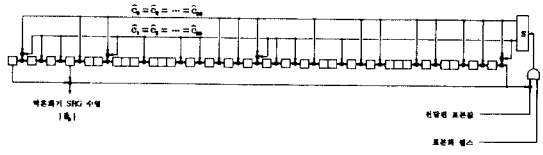

그림 8. 공통의 표본화 벡터를 가지는 간단한 역혼화기

Fig. 8. simple descrambler having common sampling vector.

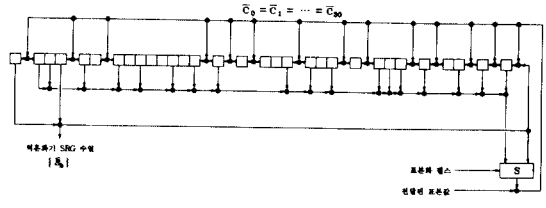

그림 9. 공통의 정정 벡터를 가지는 간단한 역혼화기

Fig. 9. Another simple descrambler having common correction vector.

없이 분산 표본 혼화기를 간단히 구현하는 방법을 제시하였다.

표본화 벡터를 도입한 분산 표본 혼화는 표본값을 취하는 방식을 일반화한 것으로서, 원래의 분산 표본 혼화는 모든 표본화 벡터가 발생 벡터와 같은 특수한 경우에 해당한다. 그러므로, 표본화 벡터를 도입한 분산 표본 혼화는 일반화된 분산 표본 혼화 방식이며, 본 논문의 표본화 및 정정 조건은 원래의 분산 표본 혼화를 위한 표본화 및 정정 조건의 일반적인 것에 해당한다.

표본화 조건은 정정 조건에 비해 보다 근본적인 것으로서, 식 (10)의 판별 행렬이 가역이 되도록 표본화 시간 및 표본화 벡터를 선택해야 한다는 것이다. 즉, 판별 행렬이 비가역이 되도록 표본화 시간과 표

본화 벡터를 취했다면, 동기화시킬 수 있는 정정 시간 및 정정 벡터가 존재하지 않게 된다. 일단 표본화 조건이 만족되면 정정 시간은 임의로 선택할 수 있고 정정 벡터는 식 (11)에 의해 얻을 수 있었다. 또한, 정정 시간이 고정된 경우에도, 동기화시킬 수 있는 정정 벡터가 많이 존재함을 알 수 있었다.

표본화 벡터를 도입함으로써 지연되어 전송되는 표본값들의 표본화 시간을 전송 시간과 일치시킬 수 있었다. 이렇게 함으로써, 지연되어 전송되는 표본값들을 저장하는 부가적인 메모리와 이들의 전송 시간을 알려주는 부가적인 클럭을 없앨 수 있었다.

쎌 기반 ATM 전송을 위해서 표본화 벡터를 도입하여 구현한 그림 7의 역혼화기는 CCITT가 권고한 그림 6의 원래의 역혼화기와 호환이 가능한 것이다. 즉, CCITT가 권고한 혼화기를 그대로 사용하더라도 그림 7의 역혼화기는 이것에 동기화된다. 이 역혼화기는 배타적 논리합 소자는 다소 많아졌지만, 그림 6의 원래의 역혼화기에 비해 부가적인 클럭이나 메모리가 없어서 간단함을 알 수 있다.

그림 8과 9에 있는 역혼화기는 그림 6의 역혼화기와는 호환이 불가능하다. 그러나, 이것들은 표본화 벡터만 변화시킨 것으로서 성능면에서는 그림 6의 역혼화기와 같으며, 그림 6이나 7의 표준안에 충실한 역혼화기에 비해 더욱 간단해졌음을 알 수있다. 혼화는 물리 계층의 기능임에 주의하면, 그림 8 또는 9에 있는 역혼화기는 국부적인 ATM 망에는 성공적으로 사용할 수 있겠다.

부록

본 부록에서는 보조 정리 1과 보조 정리 2를 증명하겠다.

1. 보조 정리 1의 증명

$\triangle \cdot \triangle^{-1}$는 단위 행렬이 되므로, 식 (10)으로부터 식 (12)를 얻는다.

2. 보조 정리 2의 증명

식 (14a)는 식 (9)와 식 (13)에 의해서 자명하다.

먼저 $0 \le j \le L-1$인 경우에 식 (14b)가 성립함을 보이겠다. 식 (14a)와 (12)를 이용하면,

$$\Lambda_i \cdot T^{\beta_{i+1}} \cdot \Delta^{-1} \cdot e_j = \Lambda_{i+1} \cdot \left(T^{\beta_i} \cdot \Delta^{-1} \cdot e_j + c_i \cdot v_i^t \cdot T^{\alpha_i} \cdot \Delta^{-1} \cdot e_j\right) =$$
$$\Lambda_{i+1} \cdot \left(T^{\beta_i} \cdot \Delta^{-1} \cdot e_j + c_i \cdot e_i^t \cdot e_j\right) = \Lambda_{i+1} \cdot T^{\beta_i} \cdot \Delta^{-1} \cdot e_j$$

가 되고, 이 과정을 반복적으로 적용하면, 최종적으로 $\Lambda_i \cdot T^{\beta_{i+1}} \cdot \Delta^{-1} \cdot e_j = \Lambda_{L-1} \cdot T^{\beta_{L-2}} \cdot \Delta^{-1} \cdot e_j$를 얻을 수 있다. 이 식의 오른쪽 항에 있는 Λ_{L-1}에 대해서, 식 (13)을 적용하면 관계식

$$\Lambda_i \cdot T^{\beta_{i+1}} \cdot \Delta^{-1} \cdot e_j = \Lambda_{L-1} \cdot T^{\beta_{L-2}} \cdot e_j + c_{L-1} \cdot v_{L-1}^t \Lambda_i \cdot T^{\alpha_{L-1}} \cdot \Delta^{-1} \cdot e_j$$

를 얻는다. 그러므로, 식 (12)에 의해서

$$\Lambda_i \cdot T^{\beta_{i+1}} \cdot \Delta^{-1} \cdot e_j = T^{\beta_{L-1}} \cdot e_j + c_{L-1} \cdot e_{L-1}^t \cdot e_j \Lambda_i \cdot T^{\beta_{L-1}} \cdot \Delta^{-1} \cdot e_j$$

가 된다.

다음으로, 귀납법을 사용하여 $1 \le i \le j \le L-1$인 경우에 식 (14b)가 성립함을 보이겠다. 먼저 $i=1$인 경우를 보이기 위해서, 식 (14a)를 적용하자. 그러면, 관계식 $\Lambda \cdot \Delta^{-1} \cdot e_j = \Lambda_1 \cdot (T^{\beta_0} \cdot e_j + c_0 \cdot v_0^t \cdot T^{\alpha_0} \cdot \Delta^{-1} \cdot e_j)$를 얻을 수 있고, 여기에 식 (12)를 대입하면,

$$\Lambda \cdot \Delta^{-1} \cdot e_j = \Lambda_1 \cdot (T^{\beta_0} \cdot \Delta^{-1} \cdot e_j + c_0 \cdot e_0^t \cdot e_j) = \Lambda_1 \cdot T^{\beta_0} \cdot \Delta^{-1} \cdot e_j$$

가 된다. 그러므로, $i=1$인 경우에 식 (14b)의 아래식이 성립한다. 이제 $i=1, 2, \ldots, j-1$일 때

$\Lambda \cdot \Delta^{-1} \cdot e_j = \Lambda_i \cdot T^{\beta_{i-1}} \cdot \Delta^{-1} \cdot e_j$ 가 성립한다고 가정하자. 그러면, 식 (14a)와 (12)에 의해서

$$\Lambda \cdot \Delta^{-1} \cdot e_j = \Lambda_{i+1} \cdot (T^{\beta_{i-1}} \cdot \Delta^{-1} \cdot e_j + c_i \cdot v_i^t \cdot T^{\alpha_i} \cdot \Delta^{-1} \cdot e_j)$$
$$= \Lambda_{i+1} \cdot (T^{\beta_i} \cdot \Delta^{-1} \cdot e_j + c_i \cdot e_i^t \cdot e_j) = \Lambda_{i+1} \cdot T^{\beta_i} \cdot \Delta^{-1} \cdot e_j$$

를 얻는다. 즉, $(i+1)$인 경우에도 성립한다. 그러므로, $1 \le i \le j \le L-1$인 경우에 식 (14b)가 성립된다.

마지막으로, 식 (14c)가 성립함을 보이겠다. 먼저 $i=0$인 경우를 고려하면, 식 (14a)와 (12)에 의해서

$$\Lambda \cdot \Delta^{-1} \cdot e_0 = \Lambda_1 \cdot (T^{\beta_0} \cdot \Delta^{-1} \cdot e_0 + c_0 \cdot v_0^t \cdot T^{\alpha_0} \cdot \Delta^{-1} \cdot e_0)$$
$$= \Lambda_1 \cdot (T^{\beta_0} \cdot \Delta^{-1} \cdot e_0 + c_0 \cdot e_0^t \cdot e_0) = \Lambda_1 \cdot (T^{\beta_0} \cdot \Delta^{-1} \cdot e_0 + c_0)$$

가 된다. 그러므로, $i=0$인 경우는 식 (14c)가 성립한다. 다음으로 $i=1, 2, \ldots, L-2$인 경우를 보이겠다. 식 (14b)에서 j를 i로 놓으면, 관계식

$$\Lambda \cdot \Delta^{-1} \cdot e_i = \Lambda_i \cdot T^{\beta_{i-1}} \cdot \Delta^{-1} \cdot e_i$$

를 얻고, 여기에 식(14a)와 (12)를 적용하면

$$\Lambda \cdot \Delta^{-1} \cdot e_i = \Lambda_{i+1} \cdot (T^{\beta_{i-1}} \cdot \Delta^{-1} \cdot e_i + c_i \cdot v_i^t \cdot T^{\alpha_i} \cdot \Delta^{-1} \cdot e_i) =$$
$$\Lambda_{i+1} \cdot (T^{\beta_i} \cdot \Delta^{-1} \cdot e_i + c_i \cdot e_i^t \cdot e_i) = \Lambda_{i+1} \cdot (T^{\beta_i} \cdot \Delta^{-1} \cdot e_i + c_i)$$

가 된다. 끝으로, $i=L-1$인 경우를 보이기 위해서, $i=j=L-1$를 (14b)식에 대입하자. 그러면,

$$\Lambda \cdot \Delta^{-1} \cdot e_{L-1} = \Lambda_{L-1} \cdot T^{\beta_{L-2}} \cdot \Delta^{-1} \cdot e_{L-1}$$

를 얻고, 여기에 식 (13)과 (12)를 적용하면,

$$\Lambda \cdot \Delta^{-1} \cdot e_{L-1} = T^{\beta_{L-1}} \cdot \Delta^{-1} \cdot e_{L-1} + c_{L-1} v_{L-1}^t \cdot T^{\alpha_{L-2}} \cdot \Delta^{-1} \cdot e_{L-1} =$$
$$T^{\beta_{L-1}} \cdot \Delta^{-1} \cdot e_{L-1} + c_{L-1}$$ 가 된다. 그러므로 $i=L-1$인 경우에도 식(14c)가 성립한다.

參 考 文 獻

[1] CCITT Recommendations G.702-704, 1990.

[2] J.E.Savage, "Some simple self-synchronizing digital data scramblers", *Bell Syst. Tech. J.*, vol.46, pp.449-487,

Feb. 1967.
[3] CCITT Recommendations G.707-709, June 1992.
[4] H.Kasai, S.Senmoto, and M. Matsushita, "PCM jitter suppression by scrambling", *IEEE Trans. Commun.*, vol.COM-22, pp.1114-1122, Aug. 1974.
[5] CCITT Recommendation I.432, "B-ISDN user-network interface-Physical layer specification", June 1992.
[6] CCITT Recommendation I.361, "B-ISDN ATM layer specification", June 1992.
[7] S. C. Kim and B. G. Lee, "Sampling and correction time conditions in destributed sample scrambling", in Proc. *IC²N(First International Conference on Computer Communications and Networks)*, June 1992, pp.226-229. (San Diego)
[8] S. C. Kim and B. G. Lee, "Synchronization of shift register generators in distributed sample scramblers", to appear in *IEEE Trans. Commun.* March 1994.

각 주

1) 분산 표본 혼화와 프레임 동기식 혼화에서는, 의사 불규칙 이진 수열 발생기가 혼화 및 역혼화를 위해서 사용된다. 그러나, 의사 불규칙 이진 수열 발생기는 SRG의 부분 집합이므로, 본 논문에서 얻어진 일반적인 이론은 의사 불규칙 이진 수열 발생기에도 적용할 수 있음에 유의해야 한다.

2) SDH 기반 ATM 전송에서는, ATM 셀 열의 혼화를 위해서 자기 동기식 혼화가 사용된다 [5] . 그러나, 오류 증대를 최소한으로 줄이기 위해서 특성 다항식 x43+1인 자기 동기식 혼화를 사용하기에, 혼화 효과가 만족스럽지 못하다. 그래서, 자기 동기식으로 혼화된 ATM 셀 열은 SDH 프레임에 실어진 후 다시 프레임 동기식으로 혼화된다.

3) 표본화 벡터가 도입되지 않은 원래의 분산 표본 혼화에 대한 표본화 조건 및 정정 조건은 참고 문헌 [7] 및 [8] 에 있다.

4) 원래의 분산 표본 혼화에서는 표본 수열{zi}는 SRG 수열 {SK}의 부분 집합이다. 그러나, 본 논문의 표본화 벡터를 도입한 분산 혼화에서는 {zi}가 {sk}와는 독립적으로 취해짐에 주의해야 한다.

5) 본 논문의 덧셈과 곱셈은 아래첨자 및 윗첨자에 있는 경우를 제외하고는 모두 이법(modulo-2)이다.

6) 역혼화기의 실제 표본화 시간은 혼화기의 각 표본값이 역혼화기에 전달되는 시간이다. 그러나, 혼화된 입력 데이타 수열과 표본값이 겪는 전송 지연은 동일하므로, 전송 지연을 무시하여도 아무런 문제가 없다. 그러므로, 혼화기의 표본화 시간과 역혼화기의 표본화 시간을 동일하게 생각할 수 있다.

7) 원래의 분산 표본 혼화는 각각의 표본 벡터 vi, i=0,1,…,L-1가 발생 벡터 h와 같은 특수한 경우이다.

8) 정정 벡터에 관한 보다 자세한 설명은 참고 문헌 [7] 과 [8] 에 있다.

9) 의사 불규칙 이진 수열은 발생시키는 SRG의 상태 천이 행렬은 항상 가역이다.

10) 표본화 시간과 정정 시간은 상대적인 값만이 의미를 가짐에 주의해야 한다. 즉, 식(10)의 판별 행렬의 가역성 여부와 식(11)의 정정 벡터에 관한 식은 0번째 표본화 시간 a0와는 무관함을 알 수 있다. 그러므로, 표본화 시간 a0를 0으로 놓아도 무방하다.

11) 표본화 시간과 정정 시간이 균일한 경우, 공통인 정정 벡터가 항상 존재한다. 이에 대한 증명은 참고 문헌 [8] 에 있다.

12) 그림 3과 같은 SRG를 사용하고, 표본값이 그림 5와 같이 전달되는 ATM 셀 혼화에 있어서는, 한 개의 표본화 벡터와 한 개의 정정 벡터를 가지면서 부가적인 클럭이나 메모리가 없는 분산 표본 역혼화기는 존재하지 않는다.

13) 부가적인 클럭과 메모리를 없애려면, 표본화 시간은 그림 4a의 점선 화살표처럼 되어야 하고, 정정 시간은 그림 4c의 점선 화살표처럼 되어야 한다.

著者紹介

金 錫 昌(正會員)
1964年 8月 16日生. 1987年 서울대학교 전자공학과(공학사). 1989年 서울대학교 전자공학과(공학석사). 1989年 현재 본 대학원 박사과정 재학중. 주관심 분야는 BISDN과 동기식 전송 방식 등임.

李 秉 基(正會員) 第 30卷 第 3號 參照
현재 서울대학교 전자공학과 부교수