Cyclic D/A Converters Based on Iterated Function Systems

Junya SHIMAKAWA†, Student Member and Toshimichi SAITO†, Member

SUMMARY This letter considers relationship between cyclic digital-to-analog converters (DACs) and iterated function systems (IFSs). We introduce the cyclic DACs as inverse systems of analog-to-digital converters in terms of one-dimensional maps. We then compare the DACs with a typical example of existing applications of IFSs: chaos game representation for analysis of DNA structures. We also present a simple test circuit of a DAC for Gray decoding based on switched capacitors and confirm the basic operation experimentally.

key words: cyclic digital-to-analog converters, cyclic analog-to-digital converters, iterated function systems, chaos game representation, switched capacitors

1. Introduction

Cyclic analog-to-digital converters (ADCs) and digital-to-analog converters (DACs) are indispensable tools in various signal processing systems. There exist various ADC architectures including binary-coding-based ADCs, Gray-coding-based ADCs and their analogues [1]–[4]. The architectures of DACs are usually based on inverse operation of the ADCs. In order to realize these data conversion systems, switched-capacitor-based implementation has been studied intensively [1], [5]–[7]. On the other hand, iterated function systems (IFSs) [8] have been studied as interesting nonlinear dynamical systems having rich phenomena including Sierpinski triangle attractor. In the field of nonlinear dynamical systems, there exist various applications of the IFSs including DNA sequence analysis, image information compression and encryption [8]–[10]; however, DACs can not be found in the applications. On the other hand, in the field of analog signal processing, there exist many DAC circuits including cyclic DACs [5]–[7]; however, IFSs are not discussed in the existing literatures. There exists a gap between these two fields.

As a first step to bridge the gap, this letter compares a class of cyclic DACs with an existing application of IFSs and presents a novel DAC circuit. First, as a preparation, we introduce basic cyclic ADCs in terms of 1D maps [11]. A basic ADC example for Gray coding [2] is shown. Second, we introduce the cyclic DACs as inverse systems of the ADCs and compare the DACs with a typical example of existing applications of IFSs (general discussion of the applications is hard). The example is the chaos game representation (CGR [9]): it has digital input sequence consisting four bases of a DNA sequence and the output is used to analyze the DNA structures. This discussion may contribute to introduce new researchers to the nonlinear field and may provide useful information for future development of efficient DACs. Third, we present a simple circuit of a DAC for Gray decoding based on switched capacitors [12]. The basic circuit operation can be confirmed experimentally. The influence of capacitance variation for data conversion characteristics is also discussed. The Gray decoding circuit cannot be found in the existing literatures and is based on switched circuits for binary decoding [5]–[7].

2. ADCs and Chaos

As a preparation to consider the DACs, we introduce basic cyclic ADCs in terms of one-dimensional maps (1D map, [1]). Let us consider ADCs which convert a constant analog input $X \in I \equiv [0, 1]$ to a digital output sequence $y \equiv \{y(1), \ldots, y(l)\}$, $y(l) \in [0, 1]$ where $l$ is a finite code length. The operation is described by Eq. (1).

\[
\begin{align*}
x(n + 1) &= f(x(n)) \\
x(n) &= \begin{cases} 
  f_0(x(n)) & \text{for } x(n) \in I_0 \\
  f_1(x(n)) & \text{for } x(n) \in I_1 
\end{cases} \\
y(n) &= Q(x(n)) = \begin{cases} 
  0 & \text{for } x(n) \in I_0 \\
  1 & \text{for } x(n) \in I_1 
\end{cases}
\end{align*}
\]

where $n$ denotes discrete time, the input $X$ is applied as an initial value $x(1)$ and the output is given via one-bit quantizer $Q$. Also, $f_0$ and $f_1$ are expanding affine mappings:

\[
\begin{align*}
f_0 : I_0 &\rightarrow I, \quad \left| \frac{d}{dx} f_0(x) \right| > 1 \quad \text{for } x \in I_0 \\
f_1 : I_1 &\rightarrow I, \quad \left| \frac{d}{dx} f_1(x) \right| > 1 \quad \text{for } x \in I_1
\end{align*}
\]

where $I = I_0 \cup I_1$ and $I_0 \cap I_1 = \emptyset$. Since the 1-D map $f : I \rightarrow I$ is expanding, it can generate chaos if $l = \infty$ [11]. Equation (1) includes the following typical example as shown in Fig. 1.

\[
x(n + 1) = \begin{cases} 
  -2x(n) + 1 & \text{for } x(n) \in I_0 \\
  2x(n) - 1 & \text{for } x(n) \in I_1 
\end{cases}
\]

where $y$, $x(1)$ and $n$ are given by Eq. (1). This is based on the
of the DAC can be described by DACs. We refer to estimation are given, Eq. (4) is updated and the final value is to be the ≤ where 1

As an ADC for Gray coding [3]. The Gray code for l

f

corresponds to the example in Eq. (2) for Gray coding. In the chaos from ergodic theoretical viewpoint [11], we pay attention to statistic properties depending on the initial value X correspond to the chaos game [9]. This DAC can provide estimation X = X(l) from Y and we have the following criterion:

| X − X | < 2−l for any X(1) ∈ I.

(6)

This is an important factor to evaluate performance of the DACs, however, it is hard to give such factor for the generalized cases of Eq. (4).

As discussed in section 1, the IFSs have various applications and their general discussion is hard. We then focus on an interesting example of the existing applications of IFSs: the chaos game representation (CGR [9]) for analysis of DNA sequences. The dynamics of the CGR is described by

\[
X(n + 1) = \begin{cases} 
\frac{1}{2}(-X(n) + 1) & \text{for } Y(n) = 0 \\
\frac{1}{2}(X(n) + 1) & \text{for } Y(n) = 1
\end{cases}
\]

(5)

where Y(n) ∈ {0, 1}. Figure 2 illustrates the map with Gray codes and the conversion characteristics. This DAC can realize the following DAC.

\[
X(n + 1) = \frac{1}{2}(X(n) + Y(n))
\]

(7)

where X ∈ (X1, X2) and Y ∈ (Y1, Y2). Y can take four values corresponding to four bases of DNA sequences: Y ∈ \{a, t, c, g\}, a ≡ (0, 0), t ≡ (1, 0), c ≡ (0, 1) and g ≡ (1, 1). As a DNA sequence is applied to the CGR, the sequence \{X(n)\} may construct a proper image for any X(1) ∈ I \times I that can be used to investigate/classify patterns in the DNA sequence. In addition, if one element of the four bases lacks, e.g., Y ∈ \{a, t, g\}, the image is to be the Sierpinski triangle as shown in Fig. 3: it corresponds to the chaos game [9]. Although IFS can be common mathematical model for the cyclic DACs (Eq. (4)) and CGRs (Eq. (7)), we should note the following contrasts:

1. In the DACs, X is an encoded digital input. In the CGR, Y is a DNA sequence data.
2. In the DACs, the output is a point X(l) at finite time l. In the CGR, we pay attention to the limit set \{Xn\} such as the Sierpinski triangle.

3. DACs and CGR

We have cyclic DACs corresponding to the cyclic ADCs in Eq. (1). Let an ADC of Eq. (1) convert an analog value X to a digital sequence y. In order to decode y, the DAC uses the following digital input Y that is the inverse sequence of y:

\[
Y ≡ \{Y(1), \cdots, Y(l)\} = [y(l), \cdots, y(1)]
\]

(3)

where Y(l) = y(l − i + 1). Let \tilde{X} ∈ I be the output of the DACs. We refer to \tilde{X} as an estimation of X. The operation of the DAC can be described by

\[
X(n + 1) = \begin{cases} 
 f_0^{-1}(X(n)) & \text{for } Y(n) = 0 \\
 f_1^{-1}(X(n)) & \text{for } Y(n) = 1
\end{cases}
\]

(4)

where 1 ≤ n ≤ l < ∞. If an initial value X(1) and an input Y are given, Eq. (4) is updated and the final value is to be the estimation \tilde{X} = X(l). Since \( f_0 \) and \( f_1 \) are affine mappings, we can guarantee existence of their inverse mappings \( f_0^{-1} : I \to I_0 \) and \( f_1^{-1} : I \to I_l \). It should be noted that \( f_0^{-1} \) and \( f_1^{-1} \) are contractive since \( f_0 \) and \( f_1 \) are expanding: Eq. (4) is an IFS consisting of contractive mappings. We then give a DAC corresponding to the example in Eq. (2) for Gray coding. As Y is given by Eqs. (1), (2) and (3), the Gray decoding is

valley map (an analogue of the tent map [11]) and is used as an ADC for Gray coding [3]. The Gray code for l = 3 (code length 3) are shown in Fig. 1 where \( f^3 \) denotes the 3-fold compositions of f. It should be noted that there exist various examples including ADC for binary coding based on the dyadic map [3], [4], [11]. Although 1-D map can be a common mathematical model for cyclic ADCs and chaos generators, we should note the following.

1. ADCs operates within finite time (l < ∞), however, chaos can not be recognized within finite time.
2. In the ADCs, we pay attention to the finite output sequence y depending on the initial value X that is an analog input. In the chaos from ergodic theoretical viewpoint [11], we pay attention to statistic properties of infinite sequence \{x(n)\} which is independent of the initial value.

Figure 1 Cyclic ADC for Gray coding. The orbit corresponds to \{y(1), y(2), y(3)\} = {0, 1, 0}.

Figure 2 The map of DAC and conversion characteristics. The orbit corresponds to \{Y(1), Y(2), Y(3)\} = {0, 1, 0}.
3. Requests of DACs researchers include simple and concrete circuit implementation with efficient performances such as high-resolution and low-distortion. Interests of CGR researchers include generalized system description and classification/recognition of the phenomena.

4. Switched-Capacitor-Based Implementation

Here we present a simple circuit of a DAC for Gray decoding based on switched capacitors (SC). In Fig. 4, the capacitors are controlled by switch $S$ such that $S$ = off for the first half period $(n - 0.5)T \leq t < nT$, $S$ = on for the second half period $nT \leq t < (n + 0.5)T$ and $S$ operates in inverse phase of $S$. In the experiment, we have used analog switches 4066 with $V_{DD} = 8V$ and $V_{SS} = -8V$. Let $v_3((n-0.5)T) = v_1(n-1), \alpha \equiv C_1/(C_1 + C_2)$ and let the circuit behavior be ideal [12]. The digital input $y \in [V_{SS}, V_{DD}]$ is applied to the circuit via switch $S_u$ with clock period $T$. If $y(t) = V_{SS}$ during one period then $S_u$ = on and the dynamics is described by Eq. (8).

$$v_1(t) = -v_2(t) = -v_1(n-1), \quad v_2(t) = E$$
for $S$ = on $((n-0.5)T \leq t < nT)$

$$v_1(t) = v_2(t) = \alpha v_1(nT-) + (1 - \alpha)v_2(nT-),$$
\begin{equation}
(8)
n = 0 \quad \text{for} \quad (n-0.5)T \leq t < nT
\end{equation}

$$v_3(t) = v_1(n)$$
for $S$ = off $((n)T \leq t < (n + 0.5)T)$

If $y(t) = V_{DD}$ during one period then $S_u$ = off and the dynamics is described by Eq. (9).

$$v_1(t) = v_1(n-1), \quad v_2(t) = E$$
for $S$ = on $(n-0.5)T \leq t < nT$)

$$v_1(t) = v_2(t) = \alpha v_1(n-1) + (1 - \alpha)E \equiv v_1(n)$$
for $S$ = off $(nT \leq t < (n + 0.5)T)$

(9)

$$v_3(t) = v_1(n)$$
for $S$ = off $(nT \leq t < (n + 0.5)T)$

Let $X(n) = \frac{v_1(n)}{E}$ and let $Y(n-1) = \frac{y(nT)-V_{SS}}{V_{DD}-V_{SS}}$. Equations (8) and (9) can be reduced into

$$X(n) = \begin{cases} 
-\alpha X(n-1) + (1 - \alpha) & \text{for } Y(n-1) = 1 \\
\alpha X(n-1) + (1 - \alpha) & \text{for } Y(n-1) = 0 
\end{cases}$$

(10)

If $\alpha = 0.5$, Eq. (10) is consistent with Eq. (5) and the circuit can realize Gray decoding. It should be noted that such a SC-based implementation has not been presented for the tent-map-based Gray decoding but has been presented for the dyadic-map-based binary decoding [5], [6]. Since such circuits correspond to IFSs consisting of contractive maps, the circuits are robust. Figure 5 shows laboratory measurements by the test circuit where the input $y$ corresponds to 4-bit sequences (0101) and (1010). As shown in Fig. 5(a), the test circuit can realize Gray decoding: capacitor voltage $v_1$ enters alternately into two bins corresponding to (0101) and (1010). In actual experiments, we can not avoid the variation of circuits elements. The variation increases as $|\alpha - 0.5|$ increases. Figure 6 shows DAC for Gray-decoding and conversion characteristics for $\alpha = 0.5$. In this case, the range of estimation $X$ is contracted as shown in Fig. 5(b). As $\alpha$ decreases from 0.5, the Gray decoding error increases. If
$\alpha > 0.5$, the range exceeds [0, 1) and we can not define the IFS.

5. Conclusions

We have considered cyclic DACs based on IFSs and have compared the DACs with the CGR. We have also presented a simple test circuit of a DAC for Gray decoding. Future problems include (1) analysis of encoding/decoding properties for wider class of ADCs/DACs based on IFSs, (2) extension to higher dimensional systems, (3) consideration of relationship between nonlinear dynamics of the IFSs and data conversion performances, and (4) practical circuit implementation.

References