Study on wavelet repetitive control

C.M. Chang and T.S. Liu

Abstract: Repetitive control has been applied to robots, disk drives, etc. However, the memory size in repetitive controllers linearly increases with both the repetition time and the sampling frequency. Hence, the authors aim to apply discrete wavelet transforms to lower the memory requirements of repetitive controllers. The repetitive error signal is decomposed using a wavelet transform. Only the highest-level approximation coefficients and a few detail coefficients in the wavelet transform have to be saved and hence the error signal is compressed. The compressed error signal is in turn reconstructed to compensate the control system. Simulation results are presented to demonstrate the effectiveness of the proposed method.

1 Introduction

Repetitive control systems have been successfully applied in industrial settings [1–4]. Usually, repetitive controllers contain not only a memory to memorise past error signals, but also a low-pass filter to ensure stability. The memory demand in repetitive controllers linearly increases with both the repetition time and the sampling frequency.

The wavelet transform [5] is a powerful tool used in many signal processing applications including audio, image and video compression and de-noising. The benefits from using wavelet transforms in image compression systems have been highlighted in [6]. We now intend to use a wavelet transform to decompose a signal into a high-pass component with detail coefficients and a low-pass component with approximation coefficients. After downsampling and omitting those detail coefficients whose values are small, the original signal is compressed while still retaining necessary information. Signals after reconstruction can be used in repetitive control. The use of a wavelet decomposition tree followed by saving the highest-level approximation coefficients and some detail coefficients means that the signal can be further compressed.

If the repetition time is long, the memory size becomes a significant issue in the implementation of a repetitive controller in a digital signal processor. Based on a wavelet transform method [7], we now develop a wavelet repetitive controller to significantly reduce the memory requirement in repetitive controllers. A simulation study will be conducted to investigate appropriate wavelet orders and decomposition trees in wavelet repetitive control.

2 Repetitive control

For linear time-invariant plants, a repetitive control can be developed, based on an internal model principle [8], that is able to provide an exact asymptotic output tracking of repetitive inputs. The internal model principle states that the output of a plant can be made to asymptotically track a class of reference commands without a steady-state error if the generator for the reference signal is included in a stable closed-loop system [9]. A repetitive control system is shown in Fig. 1. The controller includes a low-pass filter $F(s)$ and a memory e^{-Ls}. The main idea of repetitive control is that the error signal e in the previous repetition is used to reduce the current error caused by a periodic reference input r. To stabilise the repetitive control system, based on the small gain theorem, the low-pass filter $F(s)$ has to satisfy [10]:

$$|F(s)[1+P(s)]^{-1}| < 1 \tag{1}$$

where $P(s)$ is a plant transfer function. In repetitive control, the number of memory data sets is the same as the sampling time during one repetition; i.e. the sampling time determines the memory size.

3 Wavelet transform

Wavelets are viewed as a basis for representing any function and the 'wavelet transform' technique can be considered to be a technique for time-frequency analysis [11]. In the z-domain, $X(z)$ and $X_k(z)$ denote respectively the input and output of a scalar filter $H_k(z)$ and are related by:

$$X_k(z) = H_k(z)X(z)$$

so that

$$x_k(n) = \sum_m x(m)h_k(n-m) \tag{2}$$

In the discrete-time form, the discrete wavelet transform (DWT) is define as [12]:

$$y_k(n) = \sum_{m=-\infty}^{\infty} x(m)h_k(2^{k+1}n-m) \quad 0 \le k \le M-2$$

$$y_{M-1}(n) = \sum_{m=-\infty}^{\infty} x(m)h_k(2^{M-1}n-m) \tag{3}$$

and the inverse discrete wavelet transform IDWT is defined as:

$$x(n) = \sum_{k=0}^{M-1}\sum_{m=-\infty}^{\infty} y_k(m)f_k(n-2^{k+1}m) \tag{4}$$

IEE Proceedings online no. 20040337

doi: 10.1049/ip-cta:20040337

Paper first received 11th September 2002 and in revised form 18th December 2003

The authors are with the Department of Mechanical Engineering, National Chiao Tung University, Hsinchu 30010, Taiwan, ROC

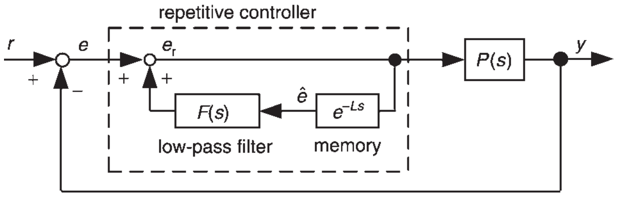


Fig. 1 *Block diagram of a repetitive control system*

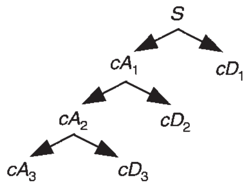


Fig. 3 *Structure of three-level wavelet decomposition tree*

where $h_k(t)$ denotes the analysis filters, $f_k(t)$ the synthesis filters defined in [12] and the signals $y_k(n)$ are the wavelet coefficients. Wavelet analysis captures both the low-frequency components (approximation coefficients) and high-frequency components (detail coefficients) of a signal. The approximations and details can be extracted using a discrete sampling or successive filtering technique [5]. A discrete wavelet decomposition tree (asymmetric dyadic filter bank) is shown in Fig. 2 [13–15]. In level 1, an original signal S is passed through a pair of complementary low-pass and high-pass filters (L and H) and is then downsampled to yield a detail coefficient cD_1 and an approximation coefficient cA_1. The respective data set numbers for cD_1 and cA_1 reduce to become half the number of data sets in S. In the subsequent levels the resultant approximation coefficient are filtered and downsampled and thus the signal S is broken down into many lower-resolution components. Furthermore, through upsampling and the use of the reconstruction filters L′ and H′, the wavelet coefficients synthesise a signal S^r without losing information. The discrete wavelet decomposition tree can also be expressed as a tree structure. Figure 3 shows the tree structure of a three-level decomposition. An approximation coefficient cA_j for the jth level describes signal S in the frequency band [13]:

$$f = [0,\ 2^{-j-1}f_s] \tag{5}$$

where f_s denotes the sampling frequency, whereas a detail coefficient cD_j describes signal S in the frequency band:

$$f = [2^{-j-1}f_s,\ 2^{-j}f_s] \tag{6}$$

The detail coefficient cD_j can be further decomposed using a discrete wavelet decomposition tree (symmetric dyadic filter bank) to obtain information on the signal S in different frequency sub-bands [15]. In other words, the DWT decomposes a signal into different frequency sub-bands.

The concept behind compression is that a regular signal S can be accurately approximated by using a small number of approximation coefficients at a suitably chosen level and a few detail coefficients [16]. To compress signal S, a threshold is prescribed and a DWT is carried out to a suitable level. Those detail coefficients that are greater than the threshold are retained whereas the others are removed so as to save memory. In reconstruction, all the omitted detail coefficients are set to zero and the reconstructed signal S^r can still accurately represent the original signal S.

A plant $P(s)$, as shown in Fig. 1, usually behaves like a low-pass filter and the error signal $\hat{e}$ in the memory passes through the low-pass filter $F(s)$ before adding to the forward path. In the initial reduction only low-frequency information in the repetitive error e_r in effect passes through the repetitive controller. This makes it possible to maintain a proper control performance after setting all the detail coefficients cD_j to zero in the reconstruction with a suitable decomposition level. Hence, the removal of the detail coefficients cD_j will be utilised for the initial reduction of the required memory. In subsequent reductions, retaining the necessary detail coefficients can increase the decomposition level, thereby further reducing the memory size. Those detail coefficients to be retained can be determined by decomposing the detail coefficients or examining detail coefficient values. Some simulation results are discussed later in Sections 5.4 and 5.6.

4 Wavelet transform for the repetitive controller

We now intend to apply the DWT signal compression technique to reduce the amount of memory data. Figure 4 depicts a repetitive controller containing DWT, which deals with a repetitive error signal $e_r(k)$. After the signal $e_r(k)$ is compressed, the memory z^{-L} memorises those approximation coefficients that remain in the highest level and retained detail coefficients, which are in turn reconstructed by using the IDWT. The synthesised signal $\hat{e}(k)$ that approximates $e_r(k)$ is used as the input signal of the low-pass filter $F(z)$ to compensate the control system. After removing the detail coefficient vectors cD_j in the preliminary stage, Fig. 5 illustrates the wavelet decomposition and reconstruction for the wavelet repetitive controller. In order to reduce the memory requirements, buffers are used to memorise the approximation coefficients in intermediate levels. The number of data sets d in each buffer is equal to the size of the decomposition filter or the reconstruction filter. Assume that the system samples n times in each repetition. The repetitive controller would require, in one repetition, n data sets worth at memory to memorise the error signals. At sampling time k in the $(m-1)$th repetition, an error buffer memories error signals $e_r(k)$, $e_r(k-1)$, … and $e_r(k-d+1)$ by shifting data. The data in this error buffer is compressed into cA_1 by the filter L and downsampling. In a similar manner, cA_1 is further compressed into cA_2. Finally, the wavelet coefficients cA_3 stored in the memory have $n/2^3$ data sets at level 3 due to three compressions. At sampling time k in the mth repetition, the data in the memory, i.e. cA_3, is reconstructed to become cA_2^r by upsampling and use of the filter L′. The cA_2^r is stored in the cA_2^r buffer to subsequently reconstruct cA_1^r. Finally, at

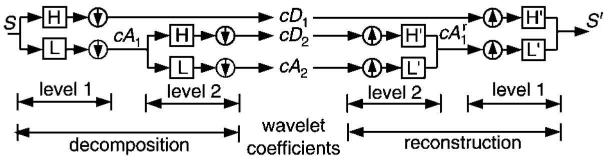


Fig. 2 *Wavelet decomposition and reconstruction tree*

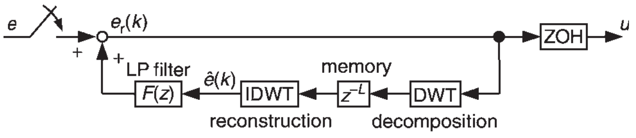


Fig. 4 *Wavelet repetitive controller*

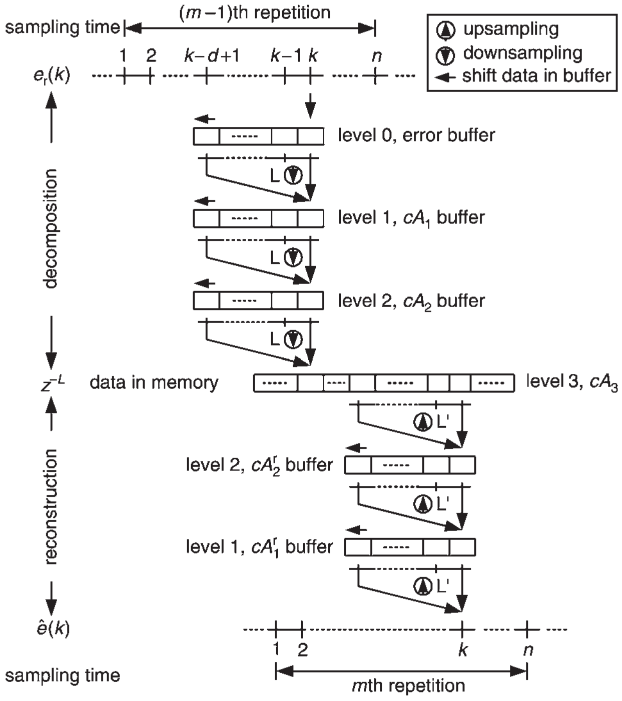


Fig. 5 *Schematic of wavelet decomposition and reconstruction*

sampling time k in the mth repetition the signal $\hat{e}(k)$ is reconstructed from the cA_1^{r} buffer.

In order to be able to store the wavelet coefficients, the buffer size must be $2d(j-1)$, where j is the number of levels in the wavelet decomposition tree. Adding the number of data sets d in the error buffer, the total memory size can be expressed by:

$$M = d + 2d(j-1) + n/2^j \tag{7}$$

We will use the Daubechies' wavelet as a transform base. Suppose that m_0 is a trigonometric polynomial such that $|m_0(\omega)|^2 + |m_0(\omega+\pi)|^2 = 1$ and $m_0(0) = 1$; the Daubechies wavelet is defined by [17]:

$$\psi(x) = \sqrt{2}\sum_k (-1)^k h_{-k+1}\phi(2x-k) \tag{8}$$

where $\phi(x) = \sqrt{2}\sum_k h_k\phi(2x-k)$ is a scale function and h_k is a filter coefficient determined by using $m_0(\omega) = (1/\sqrt{2})\sum_{k=0}^{2N-1} h_k e^{-ik\omega}$. For order $N = 1$, the Daubechies' wavelets become a Haar's wavelet [18]; i.e.:

$$\psi(x) = \begin{cases} 1 & \text{if } 0 \le x < 0.5 \\ -1 & \text{if } 0 \le x < 1 \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

The wavelet filter size d of the Daubechies' wavelets is related to the order N by [17]:

$$d = 2N \tag{10}$$

Substituting (10) into (7), the total memory size required for the initial reduction is:

$$M = 2N(2j-1) + n/2^j \tag{11}$$

Differentiating (11) with respect to level j yields the optimum decomposition level that results in the minimum memory size M:

$$j = \mathrm{round}(\ln(n\ln 2/4N)/\ln 2) \tag{12}$$

where round$(\cdot)$ rounds the value $(\cdot)$ to the nearest integer. However, wavelet repetitive control may not achieve such a high level due to losing too much information in the high frequency sub-bands. However, correctly retaining the detail coefficients can improve the decomposition level and the required memory can be further reduced. The total required memory can be calculated by summing the memory occupied by the high-level approximation coefficients in (11) and the memory size of retained detail coefficients.

According to Fig. 5, the time window of an approximation coefficient $cA_j(i)$, $i = 1, 2, \ldots, n/2^j$ in a wavelet coefficient vector cA_j in one repetition is written as a vector:

$$[cA_{j-1}(2i-(d-1)), \ldots, cA_{j-1}(k), \ldots, cA_{j-1}(2i)] \tag{13}$$

where $cA_{j-1}(k) = cA_{j-1}(k + n/2^{j-1})$ for $k \le 0$ and wavelet filter size $d = 1, 2, 3, \ldots$. Hence, the time window width of $cA_j(i)$ on cA_{j-1} is equal to the length of the coefficient vector in (13) and can be expressed by:

$$\tau_{j,j-1} = (2i) - (2i-(d-1)) + 1 = 1 + (d-1) \tag{14}$$

Furthermore, the time windows of the approximation coefficients $cA_{j-1}(2i)$ and $cA_{j-1}(2i-(d-1))$ in (13) are $[cA_{j-2}(2^2i-(d-1)), \ldots, cA_{j-2}(2^2i)]$ and $[cA_{j-2}(2^2i - 2(d-1)-(d-1)), \ldots, cA_{j-2}(2^2i-2(d-1))]$, respectively. The time window of $cA_j(i)$ on cA_{j-2} is written as a vector

$$[cA_{j-2}(2^2i - 2(d-1) - (d-1)), \ldots, cA_{j-2}(2^2i)] \tag{15}$$

The time window width of $cA_j(i)$ on cA_{j-2} is equal to the length of the coefficient vector in (15) and can be expressed by:

$$\begin{aligned}\tau_{j,j-2} &= (2^2i) - (2^2i - 2(d-1) - (d-1)) + 1 \\ &= 1 + \sum_{\alpha=0}^{1} 2^\alpha(d-1)\end{aligned} \tag{16}$$

By iteration, the time window width of $cA_j(i)$ on $cA_{j-\beta}$ is:

$$\begin{aligned}\tau_{j,j-\beta} &= (2^\beta i) - \left(2^\beta i - \sum_{\alpha=0}^{\beta-1} 2^\alpha(d-1)\right) + 1 \\ &= 1 + \sum_{\alpha=0}^{\beta-1} 2^\alpha(d-1) \\ &= d(2^\beta - 1) - 2^\beta + 2\end{aligned} \tag{17}$$

where $\beta = 1, 2, \ldots, j$. Substituting (10) into (17) gives:

$$\tau_{j,j-\beta} = 2N(2^\beta - 1) - 2^\beta + 2 \tag{18}$$

Substituting $\beta = j$ into (18), the width $\tau_{j,0}$ of the time window for $cA_j(i)$ on the repetitive error $e_{\mathrm{r}}(k)$ is:

$$\tau_{j,0} = 2N(2^j - 1) - 2^j + 2 \tag{19}$$

Equations (18) and (19) also apply to the detail coefficients in wavelet coefficient vector cD_j. According to (19), a wavelet repetitive controller using fewer levels j and lower wavelet orders N has narrower time window $\tau_{j,0}$ on the repetitive error $e_{\mathrm{r}}(k)$ and better time localisation.

5 Simulation results

By using the proposed method, a second-order plant model:

$$P(s) = \frac{\omega_{\mathrm{n}}^2}{s^2 + 2\xi\omega_{\mathrm{n}} + \omega_{\mathrm{n}}^2} \tag{20}$$

is dealt with where $\omega_n = 2000\,\text{rad/s}$ and $\xi = 0.707$. To satisfy (1), the low-pass filter $F(s)$ is prescribed as:

$$F(s) = \frac{1400}{s + 1400} \tag{21}$$

5.1 Repetitive control system

Assume a sampling frequency of $f_s = 20.48\,\text{kHz}$ and a reference signal with repetitive frequency $f_r = 10\,\text{Hz}$, the memory in a repetitive control system has to store $f_s/f_r = 20\,480/10 = 2048$ sets of data. From (21), the discrete form of the low-pass filter $F(s)$ is expressed as:

$$F(z) = \frac{0.6608}{z - 0.9339} \tag{22}$$

Based on the system depicted in Fig. 1, Fig. 6*a* compares the reference signal $r(t) = \sin 20\pi t + 0.5 \sin 100\pi t$ and the system output y against sampling steps. The repetitive controller is not turned on until the 4096th sampling step and rapidly reduces the steady-state error in three repetitions i.e. since the 10 240th step. In order to evaluate the tacking performance we define an error function:

$$E = \sum e^2(k), 4096 \leq k < T \tag{23}$$

where T is the final step in the simulation and $e(k)$ denotes the sampled error signal at the kth step. Up to $k = T = 12\,288$, E is equal to 1284 using unity feedback control, whereas E is equal to 140 using the repetitive controller.

5.2 Wavelet repetitive control system

In contrast to repetitive controllers, the currently presented wavelet repetitive controller is developed on the wavlet base comprising a second-order Daubechies' wavelet. Accordingly, each filter (low-pass or high-pass) in the DWT or the IDWT has $d = 2 \times 2 = 4$ coefficients. As shown in Fig. 6*b* where a six level decomposition tree is used, the output y is almost the same as that in the conventional controller depicted in Fig. 6*a*. The value of E remains at 141 whereas its memory size M reduces to, from (11), $2 \times 2 \times (2 \times 6 - 1) + 2048/2^6 = 76$ compared with 2048 in the repetitive controller. Hence, the present controller decreases the memory size by 96.3%. Summing up, the memory used by the compressed data size (CDS) and buffer sizes yields the total memory sizes under different conditions. Dividing 2048 by CDS yields the data compression ratio (CR) as listed in Fig. 7. As a matter of fact, the wavelet order and the decomposition tree level together determine the memory size and system performance. Figure 7 also shows that the memory size of the wavelet repetitive controller decreases with higher decomposition levels. However, the number of data sets in the buffers increases with the number of levels in the decomposition tree. Since the buffer size in each level is twice as large as the filter size, a larger wavelet order is disadvantageous in reducing the memory size. Furthermore, according to Fig. 5, the number of compressed data sets is not less than the sizes of the required filters in DWT and IDWT. Those cases where the size of the compressed data sets equals the filter size are marked in grey in Fig. 7.

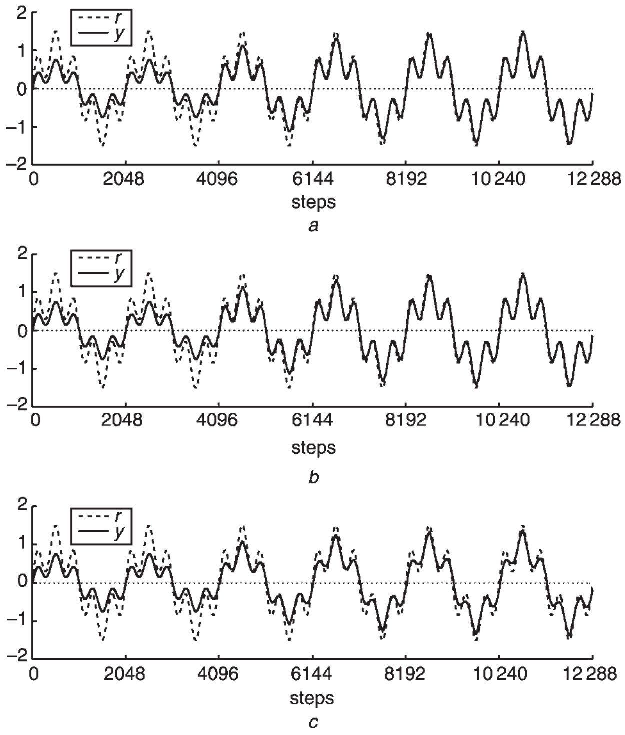


Fig. 6 *System output y when tracking reference r(t) = sin 20πt + 0.5 sin 100πt*

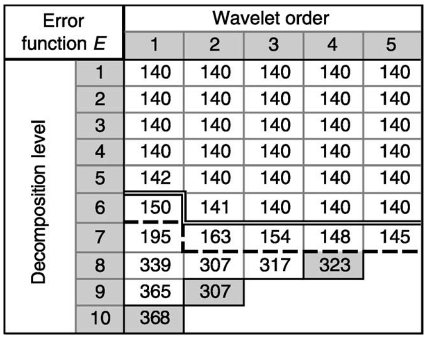
a With repetitive controller
b With a second-order wavelet repetitive controller and a six level decomposition tree
c With a second-order wavelet repetitive controller and an eight level decomposition tree

5.3 Decomposition tree level

The reconstructed signal is smoother than the original signal since the detail signal has been omitted in DWT. A higher decomposition level will lose more signal detail such that the local behaviour of the system output cannot be compensated in the next repetition. In contrast to Fig. 6*b* that depicts the result of a six level decomposition tree, increasing to eight levels worsens the system output as depicted in Fig. 6*c* and the error function E jumps to 307 in Fig. 8. Therefore, the wavelet repetitive controllers fail if the level is too high. Figure 8 compares the magnitudes of the error function E in (23) under different simulation conditions. The cases of the wavelet repetitive controllers above the double line (performance line) work as well as do the repetitive controllers. Between the dotted line (distortion line) and double line system outputs degrade. In the other case, wavelet repetitive controllers fail to compensate the repetitive error. According to (5) and a sampling frequency of $f_s = 20.48\,\text{kHz}$, the frequency sub-band of approximation coefficient cA_6 in the wavelet repetitive controller is $f = [0, 160]\,\text{Hz}$. In Fig. 8, the wavelet repetitive controller fails to maintain control performance when the decomposition tree has more than six levels since the wavelet repetitive controller must reserve information on the repetitive error

Filter size		2	4	6	8	10		
Memory size		Wavelet order						
		1	2	3	4	5	CDS	CR
Decomposition level	1	1026	1028	1030	1032	1034	1024	2:1
	2	518	524	530	536	542	512	4:1
	3	266	276	286	296	306	256	8:1
	4	142	156	170	184	198	128	16:1
	5	82	100	118	136	154	64	32:1
	6	54	76	98	120	142	32	64:1
	7	42	68	94	120	146	16	128:1
	8	38	68	98	128		8	256:1
	9	38	72				4	512:1
	10	40					2	1024:1

Fig. 7 *Required memory sizes under different wavelet orders and decomposition levels*



Error function E		Wavelet order				
		1	2	3	4	5
Decomposition level	1	140	140	140	140	140
	2	140	140	140	140	140
	3	140	140	140	140	140
	4	140	140	140	140	140
	5	142	140	140	140	140
	6	150	141	140	140	140
	7	195	163	154	148	145
	8	339	307	317	323	
	9	365	307			
	10	368				

Fig. 8 *Values of error function E when reference input r(t)= sin 20πt + 0.5 sin 100 πt*

signal $\hat{e}(k)$ in sub-band $f = [0,\ 160]$ Hz to maintain control performance.

5.4 Decomposing the detail coefficient

From Fig. 8 it can be seen that first-order wavelets can not maintain control performance when the decomposition tree has more than five levels since information in the detail coefficient vector cD_6 cannot be omitted. However, the coefficient cD_6 can be decomposed further. Figure 9*a* depicts the full decomposition tree (symmetric dyadic filter bank) of the first-order approximation coefficient cA_5. Notations $c(j, q)$ denote coefficients of the repetitive error $e_r(k)$ in the jth level and in a frequency band:

$$f = \left[2^{-j-1} q f_s,\ 2^{-j-1}(q+1) f_s\right] \qquad (24)$$

where $q = 0, 1, 2, \ldots, (2^j - 1)$ denotes a frequency parameter. The numbers in the brackets in Fig. 9*a* denote calculated values of the error function E when corresponding coefficients and their sub-branches are omitted. The approximation coefficient cA_6 and its sub–branches contain primary information on the repetitive error $e_r(k)$ and cannot be abandoned. However, coefficient $c(7,\ 3)$ i.e. the detail coefficient of cD_6 after decomposition can be omitted so as to reduce the memory size. Hence, by calculating (5) and (24), the coefficients cA_6 in sub-band $f = [0,\ 160]$ Hz and $c(7,\ 2)$ in sub-band $f = [160,\ 240]$ Hz are retained. Adding the memory size of 54 for the first-order six level case (cA_6) in Fig. 7 to the four data sets of the cD_6 buffer and the 16 data sets of the approximation coefficient $c(7,\ 2)$, the total memory size is $54 + 4 + 16 = 74$. Finally, the above decomposition tree structure decreases the memory size by 96.4% and still maintains the performance with the error function $E = 143$ as shown in the bracket under $c(7,\ 3)$ in Fig. 9*a*. Figure 9*b* depicts the output y which is almost the same as that using a repetitive controller as depicted in Fig. 6*a*.

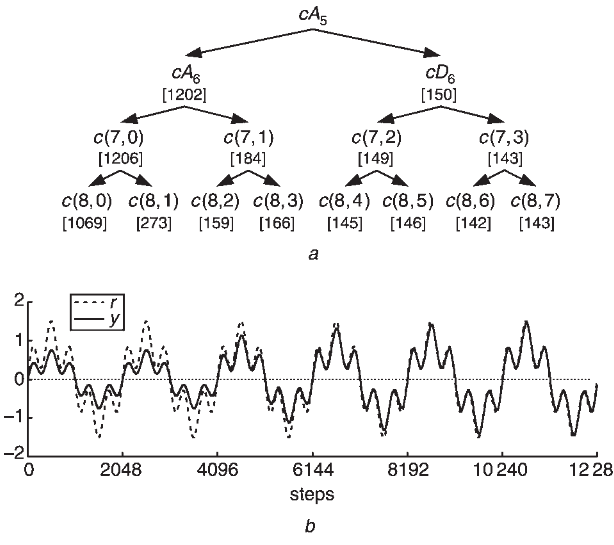
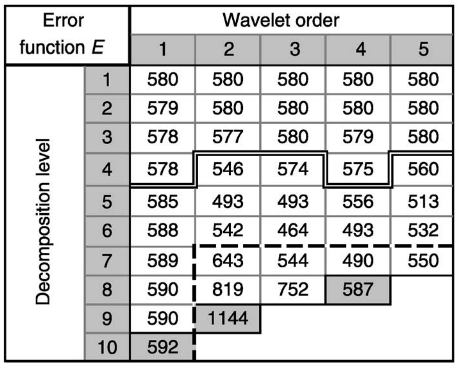


Fig. 9 *For tracking reference r(t)= sin 20πt + 0.5 sin 100πt*

a Full decomposition tree of cA_5 for the first-order wavelet repetitive controller
b Output y after omitting $c(7,\ 3)$ in Fig. 7*a*



Error function E		Wavelet order				
		1	2	3	4	5
Decomposition level	1	580	580	580	580	580
	2	579	580	580	580	580
	3	578	577	580	579	580
	4	578	546	574	575	560
	5	585	493	493	556	513
	6	588	542	464	493	532
	7	589	643	544	490	550
	8	590	819	752	587	
	9	590	1144			
	10	592				

Fig. 10 *Values of error function E when reference input r(t) is a square wave*

5.5 System bandwidth and wavelet order

In contrast to Fig. 8, Fig. 10 shows values of E when the reference signal is a unit amplitude square wave at 10 Hz. It is known that a square wave contains all the frequency contents at jump points. Both the performance line and the

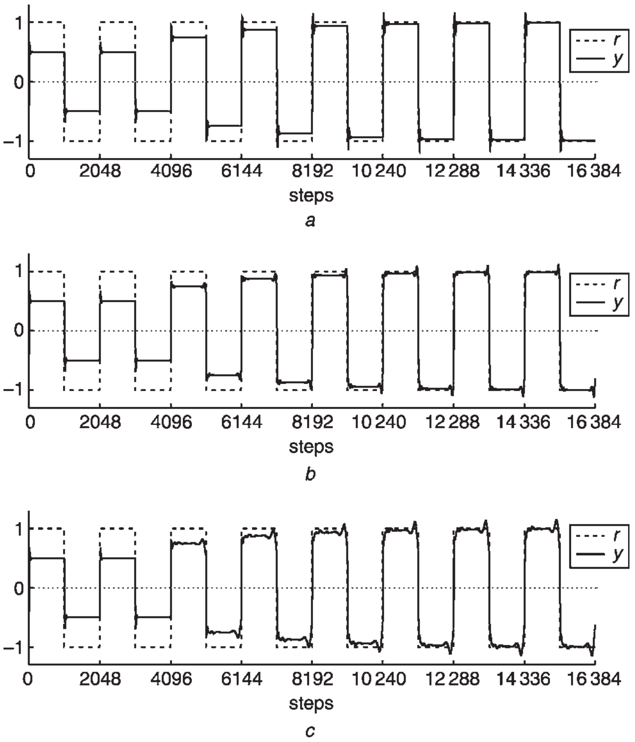


Fig. 11 *System output y when reference input r(t) is a square wave*

a With repetitive controller
b With a fourth-order wavelet repetitive controller and a six level decomposition tree
c With a fourth-order wavelet repetitive controller and a seven level decomposition tree

distortion line increase, in contrast to Fig. 8. The simulation result for the repetitive controller is shown in Fig. 11*a* with $E = 580$. The tracking error immediately after each jump has been compensated and the overshoot occurs at each jump. The wavelet repetitive controllers above the performance line maintain the performance as the repetitive controller depicted in Fig. 11*a*. According to Fig. 1, the transfer function from the memory output $\hat{e}$ to output y can be written as:

$$G(s) = \frac{F(s)P(s)}{1 + P(s)} \tag{25}$$

The bandwidth ω_b of $G(s)$ can be calculated from:

$$|G(j\omega_b)| = \frac{|G(j0)|}{\sqrt{2}} \tag{26}$$

Substituting (20) and (21) into (25) and (26) gives the bandwidth ω_b of $G(s)$ as 447 Hz. Accordingly, the wavelet repetitive controller must reserve information in sub-band $f = [0, 447]$ Hz to maintain its performance. According to (5), the bandwidths of the approximation coefficients cA_4 and cA_5 are 640 and 320 Hz respectively. The latter is smaller than $\omega_b = 447$ Hz. Hence, the wavelet repetitive controllers in Fig. 10 cannot achieve the desired output performance when the number of decomposition levels is higher than four.

The system output resulting from fourth-order wavelets and six and seven level decomposition trees are depicted in Figs. 11*b* and 11*c*, respectively. In Fig. 11*b*, overshoots are found both before and after the jump points of the square reference signal. The filter sizes in DWT and IDWT increase if a larger order is adopted. According to (19), a wavelet repetitive controller using a higher-order wavelet

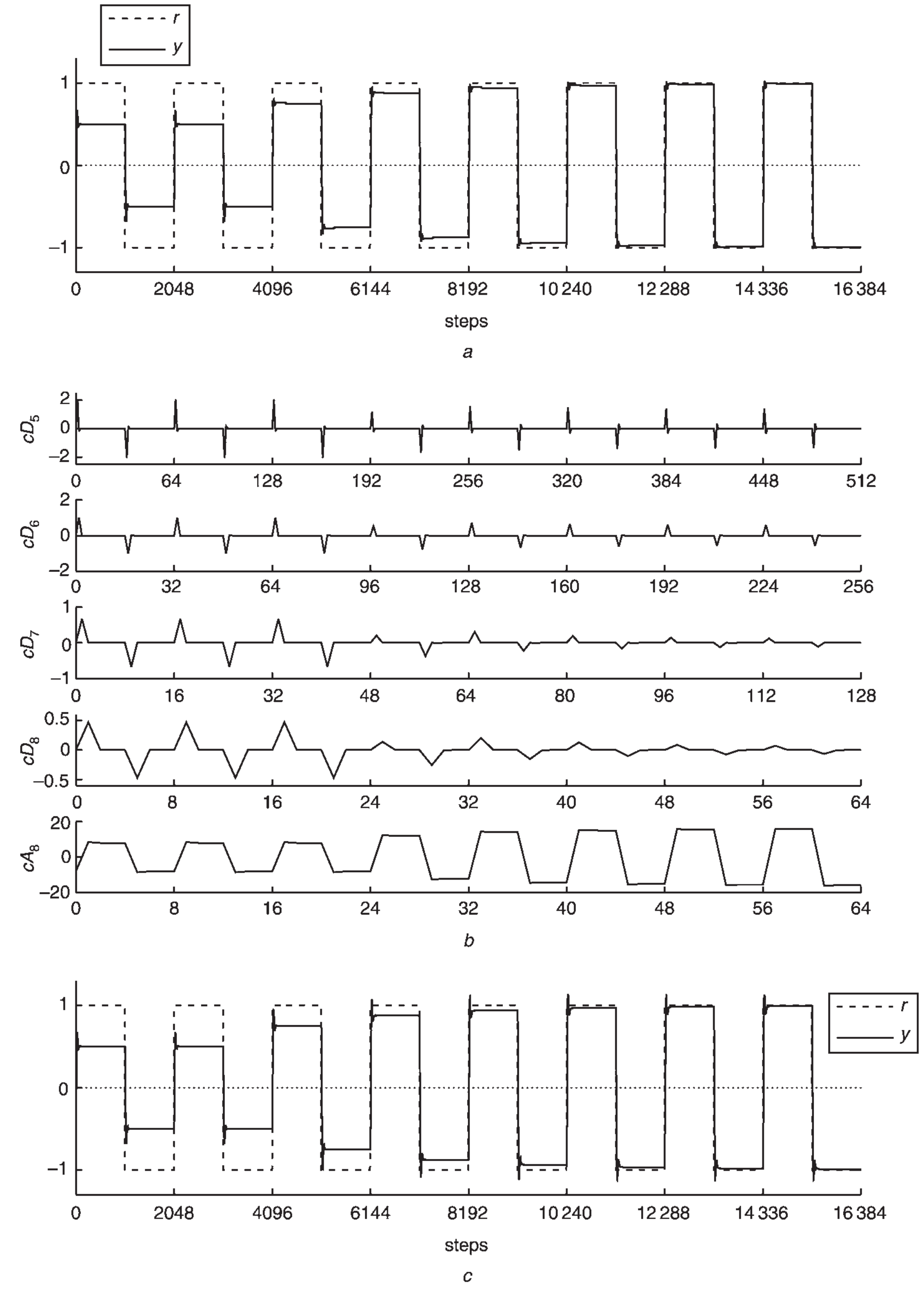


Fig. 12 *For square wave reference input r(t)*

a System output y with a first-order wavelet repetitive controller and an eight level decomposition tree
b Wavelet coefficients for Fig. 9*a*
c System output y obtained with eight detail coefficients

has a wider time window. Therefore, analysing the error signal $e_r(k)$ shown in Fig. 4 using a higher-order wavelet in DWT, the synthesised error signal $\hat{e}(k)$ of IDWT will involve broader information around the time one repetition before the one under current consideration. Hence, the wavelet controllers can predict both the trend of the error signal and overshoot after the jump. The wavelet controllers try to reduce the overshoot by degrading the system output before the jump. This causes a local distortion around the jump of the reference signal when the decomposition level is higher. Except for first-order cases the system outputs of the cases between the performance line and the distortion line in Fig. 10 show local distortion as depicted in Fig. 11*b*. When the decomposition level is larger than the distortion line, the distortion spreads and the steady-state output worsens as shown in Fig. 11*c*.

5.6 Time localisation

The distortions in jumps shown in Figs. 11*b* and 11*c* do not appear in first-order cases, since the $N = 1$ wavelet has the narrowest time window according to (19). As a consequence, the good time resolution inherent for this wavelet improves the control performance. The system output resulting from first-order wavelets with an eight level decomposition tree is obtained as shown in Fig. 12*a*. Concerning the jump at the 14 336th step, the overshoot is reduced and the rising time t_r increases from 24 steps to 31 steps compared with the one in Fig. 11*a*. The first-order error function E increases progressively higher than level 4 in Fig. 10 since the bandwidth of the wavelet repetitive controllers is less than the bandwidth of $G(s)$, as described in Section 5.5. Hence, some of the detail coefficients at jumps must be retained to improve the control performance. On the other hand, the value of a square wave immediately after each jump is constant with a zero frequency and thus can be decomposed up to the optimum decomposition level $j = 8$, calculated by using (12).

The wavelet transform obtains time information on the signals as wall as frequency information in the time-frequency domain. Figure 12*b* depicts the wavelet coefficients for the cases with greater than four levels. It can be seen that the detail coefficients cD_j have clear time localisation properties. The detail coefficients have large values at the jump points; otherwise, the detail coefficients tend to zero. Retaining eight detail coefficients $cD_5(1)$, $cD_5(33)$, $cD_6(1)$, $cD_6(17)$, $cD_7(1)$, $cD_7(9)$, $cD_8(1)$ and $cD_8(5)$ in the detail coefficient vectors cD_j in one repetition, the system output y is obtained as depicted in Fig. 12*c*. Compared with Fig. 12*a*, the transient response is improved and the rise time t_r is 24 steps at the jump of the 14 336th step. The error function E reduces from 590 to 577. Finally, adding the eight detail coefficients, the memory size increases to 46 from 38 in Fig. 12*a* and decreases the memory size by 98% compared with the repetitive controller in Fig. 11*a*.

The memory size has been slashed in wavelet repetitive controllers as compared with repetitive controllers. When the wavelet repetitive controller adopts different wavelet bases (coifilets, biorthogonal, and symlets, etc.), performances are similar. The first-order Daubechies' wavelet (a Haar's wavelet) with a small filter size is the correct choice for the two studied command signals. Therefore, the proposed method enables repetitive control to reduce the memory size.

6 Conclusions

It has been demonstrated that for repetitive controllers decomposing the repetitive error signal with DWT and retaining a few wavelet coefficient can significantly reduce the memory size. Suitable decomposition tree structures and choices of wavelet coefficients to be stored in the memory depend on: (i) the repetitive error signal length in a repetition time; (ii) signal information in the time-frequency domain; (iii) the control system bandwidth; and (iv) the selected wavelet. A DWT serves as an additional filter in the repetitive controller. Accordingly, a wavelet repetitive controller using fewer levels and lower wavelet orders uses a smaller CPU time in analysing and decomposing the error signal. Without expanding the memory size of existing repetitive control systems the use of the proposed wavelet repetitive controller can increase the repetition time. Furthermore, to avoid exceeding the memory limit even for a long repetition time, the wavelet repetitive controller does not sacrifice the bandwidth and tracking performance of the system. Therefore, the wavelet repetitive controller is valuable for a repetitive control system with a long repetition time or a high sampling frequency.

7 Acknowledgment

This work was supported in part by the National Science Council and Department of Education in Taiwan under grant 89E-FA06-1-4.

8 References

1 Katayama, T., Ogawa, M., and Nagasawa, M.: 'High-precision tracking control system for digital video disk players', *IEEE Trans. Consum. Electron.*, 1995, **41**, (2), pp. 313–321

2 Hattori, S., Ishida, M., and Hori, T.: 'Suppression control method for torque vibration of brushless DC motor utilizing repetitive control with Fourier transform'. Proc. 6th Int. Workshop on Advanced Motion Control, Nagaya, Japan, March 2000, pp. 427–432

3 Arai, K., Okumura, H., Tokumaru, H., and Ohishi, K.: 'Improvement of performance of a tracking servo system for an optical disk drive', *Jpn. J. Appl. Phys.*, 2000, **39**, pp. 855–861

4 Yamada, M., Riadh, Z., and Funahashi, Y.: 'Design of robust repetitive control system for multiple periods'. Proc. IEEE Conf. on Decision and Control, Sydney, Australia, December 2000, pp. 3739–3744

5 Louis, A.K., Maaβ, P., and Rieder, A.: 'Wavelets theory and applications' (John Wiley & Sons, New York, 1997)

6 Grgic, S., Grgic, M., and Zovko-Cihlar, B.: 'Performance analysis of image compression using wavelets', *IEEE Trans. Ind. Electron.*, 2001, **48**, (3), pp. 682–695

7 Akansu, A.N., and Haddad, R.A.: 'Multiresolution signal decomposition: transforms, subbands, and wavelets' (Academic Press, Newark, NJ, 2001, 2nd Edn.)

8 Ghosh, J., and Paden, B.: 'Nonlinear repetitive control', *IEEE Trans. Autom. Control*, 2000, **45**, (5), pp. 949–954

9 Inoue, T.: 'Practical repetitive control system design'. Proc. 29th IEEE Conf. on Decision and Control, Honolulu, USA, December 1990, vol. 3, pp. 1673–1678

10 Hara, S., Yamamoto, Y., Omata, T., and Nakano, M.: 'Repetitive control system: a new type servo system for periodic exogenous signals', *IEEE Trans. Autom. Control*, 1988, **33**, (7), pp. 659–668

11 Chui, C.K.: 'An Introduction to wavelets' (Academic Press, New York, 1992)

12 Vaidyanthan, P.P.: 'Multirate systems and filter banks' (Prentice Hall, New York, 1993)

13 Mallat, S.G.: 'A Theory for multiresolution signal decomposition: the wavelet representation', *IEEE Trans. Pattern Anal. Mach. Intell.*, 1989, **11**, (7), pp. 674–693

14 'The wavelet toolbox for use with Matlab' (The Math Works, Natick, MA, 2000)

15 Fliege, N.J.: 'Multirate digital signal processing: multirate systems, filter banks, wavelets' (John Wiley & Sons, Chichester, UK, 1994)

16 Devore, R.A., Jawerth, B., and Lucier, B.J.: 'Image compression through wavelet transform coding', *IEEE Trans. Inf. Theory*, 1992, **38**, (2), pp. 719–746

17 Daubechies, I.: 'Ten lectures on wavelets' (Society for Industrial & Applied Mathematics, 1992)

18 Lutovac, M.D., Tosic, V., and Evans, B.C.: 'Filter design for signal processing using Matlab and mathematica' (Prentice-Hall, New Jersey, 2001)