A WAVELET FILTERING APPLICATION FOR ON-LINE DYNAMIC DATA RECONCILIATION

Kuo-Yuan Luo, Hsiao-Ping Huang

Department of Chemical Engineering,
National Taiwan University,
Taipei, Taiwan, R.O.C.

Abstract: Discrete wavelet transform (DWT) is known for its signal processing ability. In the recent researches, DWT is adopted for signal filtering before executing dynamic data reconciliation. While on-line dynamic data reconciliation is concerned, the computation is heavy due to the filtering in every time instant. In this article, a shift property of the DWT is indicated and is applied to reduce the computation duty. The efficiency of this application is also discussed. Copyright © 2006 IFAC

Keywords: filtering technique, data processing, discrete wavelets transform, integral method

1. INTRODUCTION

Data reconciliation can be used to reduce the measurement errors which come from imperfect instrument measurements. In this way, the adjusted data obey the natural laws. Measurement redundancies, which include functional and temporal redundancy, are the first important consideration in processing data reconciliation. Measurements are functionally redundant if there are more than enough data to completely define the process model at any time instant in time. Measurements are temporally redundant if past measurements are available to reconcile the measurements. Then the data reconciliation problem is defined as a constrained, weighted, least-squares optimization problem.

The steady-state data reconciliation is well-documented (Narasimhan and Jordache, 2000; Romagnoli and Sanchez, 2000) and have applications in industrial processes (Crowe, 1996). However, dynamic data reconciliation is better to deal with this kind of problem in the real conditions. The formulation of the dynamic data reconciliation can be written in Eqs. (1)-(4).

\[
\min \phi[y, \hat{y}; \Sigma] \quad (1)
\]

\[
g_1 \left( \frac{d\hat{y}(t)}{dt} \right) = 0 \quad (2)
\]

\[
g_2 (\hat{y}(t)) = 0 \quad (3)
\]

\[
g_3 (\hat{y}(t)) \geq 0 \quad (4)
\]

Where, \( \phi \) is the objective function, \( g_1 \) is the differential equation constraint, \( g_2 \) is the algebraic equality constraint, \( g_3 \) is the inequality constraint, \( y \) is the measurement variable, \( \hat{y} \) is the reconciled variable and \( \Sigma \) is the covariance matrix. A general data reconciliation problem is thus formulated as minimizing an objective function of Eq. (1) subjected to equality and inequality constraints of Eqs. (2)-(4).

There have been different kinds of solutions to the optimization problems. The discretization-based methods solve the problems by transforming the differential equations into algebraic equations (Liebman et al., 1992; Rollins and Devanathan, 1993; Albuquerque and Biegler, 1996). But discretization increases the number of variables and equations, which increase the computation effort, and is impractical for applications in the real process. Dynamic data reconciliation methods by pre-treating the variables with wavelets analysis are also proposed (Kong et al., 2000; Luo and Huang, 2005; Tona et al., 2005).
If these methods are used in the on-line application, the algorithms need to be executed in every time instant which leads to a lot of computation duties. In this article, an integral approach method for dynamic data reconciliation with wavelets pre-treatment of the measurement signals is proposed. A shift property of the discrete wavelet transform (DWT) is also addressed. Based on this property, a method is proposed which can reduce the computation duties of the traditional DWT. The analysis of the efficiencies is also discussed. Finally, a four-tank process is taken to testify the proposed method.

2. FILTERING BY WAVELETS

Wavelets are families of mathematical functions, which have been applied widely for signal and image processing. Wavelets theories are introduced friendly in the text book (Burrus et al., 1998) and some simple application examples are illustrated in the Matlab user guide for wavelet toolbox (Misiti et al., 1997). In practice, the wavelets analysis is accomplished by discrete wavelet transform (DWT), whose algorithm is well matched to the digital computer. The DWT is commonly employed using dyadic filter banks, which are sets of filters that divide a signal frequency band into sub-bands. These filter banks are comprised of low-pass, high-pass filters and the outputs are the approximation coefficients and the detail coefficients. The process of obtaining the approximation and detail coefficients is called decomposition. This process can be repeated with successive approximations (the output of the low-pass filter in the first bank) being decomposed in turn, called multilevel decomposition, so that one signal is broken down into a number of components. The inverse discrete wavelet transform (IDWT) reconstructs a signal from the approximation and detail coefficients derived from decomposition at certain level. And the filtering is accomplished by omitting the detail coefficients at the proper decomposition level (Luo and Huang, 2005). The details of the proposed are described in the following article.

2.1 Moving-Horizon Filtering

As carrying out data reconciliation on-line, the filtering is repeated at every time instant with a finite length. The choices of the wavelets and the length of the moving-horizon are discussed in the following part of this section.

2.2 Shift Property of DWT

In mathematics, the DWT decomposition includes two procedures: 1. convolution, 2. down-sampling. The two procedures can be expressed by Eq. (5) (Jensen and Cour-Harbo, 2001).

\[ c_{i}[x] = \sum h_{k}[2^{k} - j] X[j] = \sum h_{k}[j] X[2^{k} - j] \]  

(5)

\( c_{i} \) is the ‘approximate’ coefficient, \( h_{k} \) is the low pass filter, \( f \) is the length of the filter and \( X \) is the original signal. The procedure is shown in the Figure 1 and constant padding is used to solve the problems of end distortion. In order to simplify the condition, \( n \) is set to be even. And assuming that the previous moving-horizon, \( X_{1} \), is represented by Eq. (6) and the instant moving-horizon, \( X_{2} \), is represented by Eq. (7).

\[ X_{1} = [x_{1}, x_{2}, \ldots, x_{n}] \]  

(6)

\[ X_{2} = [x_{3}, x_{4}, \ldots, x_{n+2}] \]  

(7)

According to the Eq. (6), as the moving-horizon moving by two points at a time, some coefficients of this window are identical with some coefficients of the previous window. So the DWT decomposition can be executed by just calculating some coefficients on two sides and shifting some coefficients of the previous window. Precisely, the shift relation can be expressed as Eq. (8).

\[ c_{12} = c_{i+6} \]  

(8)

\( c_{i} \) is the coefficient of the present moving-horizon and \( c \) is the coefficient of the previous moving-horizon. The corrected terms of both sides are calculated by Eqs. (9) and (10). In this manner, the DWT procedure is accomplished in another way but the computation is saved.

\[ \begin{bmatrix} c_{0} \\ c_{1} \\ \vdots \\ c_{i-2} \\ c_{i} \\ c_{i+2} \\ \vdots \\ c_{n} \end{bmatrix} = \begin{bmatrix} h_{0} \vdots + h_{j} & 0 & \vdots & 0 \\ h_{0} \vdots + h_{j} & h_{1} & \vdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ h_{0} \vdots + h_{j} & \vdots & \ddots & h_{j} \\ 0 & h_{0} \vdots + h_{j} & \vdots & \vdots \\ \vdots & 0 & \vdots & \vdots \\ 0 & 0 & \vdots & h_{0} \vdots + h_{j} \end{bmatrix} \begin{bmatrix} x_{0} \\ x_{1} \\ \vdots \\ x_{i-2} \\ x_{i} \\ x_{i+2} \\ \vdots \\ x_{n} \end{bmatrix} \]  

(9)

\[ \begin{bmatrix} c_{0} \\ c_{1} \\ \vdots \\ c_{i-2} \\ c_{i} \\ c_{i+2} \\ \vdots \\ c_{n} \end{bmatrix} = \begin{bmatrix} h_{0} \vdots + h_{j} & \vdots & \vdots & \vdots \\ \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & h_{0} \vdots + h_{j} \end{bmatrix} \begin{bmatrix} x_{0} \\ x_{1} \\ \vdots \\ x_{i-2} \\ x_{i} \\ x_{i+2} \\ \vdots \\ x_{n} \end{bmatrix} \]  

(10)

2.3 Applying the Shift Property to Multilevel DWT Decomposition

The shift property described in the previous article can be revealed in the multilevel decomposition if the following two conditions are held.

1. The length of the coefficients of different levels must be even (identical to the original assumption).
2. The coefficients must be moved by two points compared with the previous moving-horizon.

Due to the first reason, as processing multilevel DWT decomposition, the length of the moving-horizon must be with a specific length according the types of the wavelets. In order words, there are specific lengths of the chosen wavelets to satisfy this condition. Some specific lengths of the Daubechies wavelets are listed in Table 1. For example, if five-level DWT decomposition is desired, the smallest length of the moving-horizon will be the number at the column No. 5 at Table 1 to make sure the length of different level to be even.

The second condition is that coefficients at the different DWT decomposition level must be moved by two points compared with the certain previous moving-horizon.

A phenomenon of DWT is described firstly. From Eq. (8) (disregard the corrected terms), it is known that the 1st level coefficients are shifted by
one point if the 0th level coefficients (i.e., original signal) are moved by two points. The 2nd level coefficients are shifted by one point if the 1st level coefficients are moved by two points or the 0th coefficients are moved by four points. In this manner, the shift property holds at level $m$ if the original moving-horizon is moved by $2^m$ points, which restricts the application for on-line filtering that needs to execute at every time instant (if $m$ level DWT decomposition is desired). But this can be solved by setting a dataset containing the information of different levels coefficients shown in Figure 3. In Figure 3, the moving-horizon is represented by a solid line. The moving-horizon in the time instant is the line on the rightest. The size of dataset is determined by the chosen DWT level. If the desired multilevel DWT is $m$, then $2^m$ moving-horizons need to be built. Furthermore, the first level coefficients can be obtained by calculation from the current moving-horizon and shifting some coefficients from the first level coefficients of the previous two moving-horizons. The second level coefficients can be obtained by calculation from the current first level coefficients and shifting some coefficients from the second level coefficients of the previous four moving-horizon. And the corrected terms at each level are listed in Table 2. In this manner, the on-line filtering by DWT using shift property can be executed at each time instant. In Figure 1, which illustrate just one of the moving-horizon in the test signals, the results are identical between direct DWT decomposition and proposed method. (In the following examples, wavelets Daubechies 6 is adopted and the length $f$ is equal to 12.)

![Image of Table 2](image)

**Table 2 Corrected terms at each level**

<table>
<thead>
<tr>
<th>level</th>
<th>Left side</th>
<th>Right side</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$(f-2)/2 \ l_1$</td>
<td>$f/2 \ r_1$</td>
</tr>
<tr>
<td>2</td>
<td>floor$(l_1+f-1)/2 \ l_2$</td>
<td>ceil$(r_1+1)/2+floor(f-1)/2 \ r_2$</td>
</tr>
<tr>
<td>3</td>
<td>floor$(l_2+f-1)/2 \ l_3$</td>
<td>ceil$(r_2+1)/2+floor(f-1)/2 \ r_3$</td>
</tr>
</tbody>
</table>

*“floor” rounds the elements of X to the nearest integers towards minus infinity.

**Table 1 Specific length of different Daubechies wavelets**

<table>
<thead>
<tr>
<th>wavelets</th>
<th>Filter length</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>db02</td>
<td>4</td>
<td>4</td>
<td>6</td>
<td>10</td>
<td>18</td>
<td>34</td>
<td>66</td>
<td>130</td>
</tr>
<tr>
<td>db04</td>
<td>8</td>
<td>8</td>
<td>10</td>
<td>14</td>
<td>22</td>
<td>38</td>
<td>70</td>
<td>134</td>
</tr>
<tr>
<td>db06</td>
<td>12</td>
<td>12</td>
<td>14</td>
<td>18</td>
<td>26</td>
<td>42</td>
<td>74</td>
<td>138</td>
</tr>
<tr>
<td>db08</td>
<td>16</td>
<td>16</td>
<td>18</td>
<td>22</td>
<td>30</td>
<td>46</td>
<td>78</td>
<td>142</td>
</tr>
<tr>
<td>db10</td>
<td>20</td>
<td>20</td>
<td>22</td>
<td>26</td>
<td>34</td>
<td>50</td>
<td>82</td>
<td>146</td>
</tr>
<tr>
<td>db12</td>
<td>24</td>
<td>24</td>
<td>26</td>
<td>30</td>
<td>38</td>
<td>54</td>
<td>86</td>
<td>150</td>
</tr>
<tr>
<td>db14</td>
<td>28</td>
<td>28</td>
<td>30</td>
<td>34</td>
<td>42</td>
<td>58</td>
<td>90</td>
<td>154</td>
</tr>
<tr>
<td>db16</td>
<td>32</td>
<td>32</td>
<td>34</td>
<td>38</td>
<td>46</td>
<td>62</td>
<td>94</td>
<td>158</td>
</tr>
<tr>
<td>db18</td>
<td>36</td>
<td>36</td>
<td>38</td>
<td>42</td>
<td>50</td>
<td>66</td>
<td>98</td>
<td>162</td>
</tr>
<tr>
<td>db20</td>
<td>40</td>
<td>40</td>
<td>42</td>
<td>46</td>
<td>54</td>
<td>70</td>
<td>102</td>
<td>166</td>
</tr>
</tbody>
</table>

2.4 Modified Application

The shift property can save the computations compared with the original DWT algorithm. In order to obtain larger computation efficiency, a modified application is also proposed.

As on-line application, not all of the data in filtered signal in the moving-horizon is needed. Filtered data near the time instant is desired in the application, so the correction action on the right side is executed and the right side result of reconstruction is consistent with the right side result of the direct IDWT. The number of the identical data depends on which level is chosen. The larger chosen level leads to less identical terms. Further, as reconstruction, it can just focus on the desired terms. Not all of the filtered data needs to be reconstructed. It can reconstruct the data with certain points at each level. For example, if the last 12 terms of the filtered data are needed, then the reconstruction at each level can be set to 12 terms to obtain the final 12 terms. In this way, the computation is saved more.

![Image of Fig. 1](image)

**Fig. 1. on-line filtering application**

2.5 Determination of DWT Level

Determination of the DWT level is accomplished by the method proposed by Luo and Huang (2005). From DWT and IDWT procedure, the signal is decomposed into various parts, i.e., high frequency (detail) and low frequency (approximate) part. Like shown in Figure 2, $A_i$ is the approximate of the signal and $D_i$ is the detail of the signal. The relation between $A_i$ and $D_i$ is expressed in Eq. (11).

$$A_i = A_{i+1} + D_{i+1}$$ (11)

The square errors of $D_i$, i.e. the energy of the detail part, represent the removed energy after decomposition at the level $i$. The energy of $D_i$ should be decreases as $i$ increases due to the narrower band-pass region. If the energy of $D_i$ increases suddenly at certain level, which means some dominated low frequency signal is filtered, then the decomposition level is determined at the previous level.
2.6 The Efficiency Analysis

In this section, the efficiency is discussed. In order to discuss the efficiency of the proposed algorithm, it is assumed that a convolution calculation is viewed as a computation unit. For traditional DWT, the computation unit of different level can be calculated by Eq. (12) which means the length after convolution.

$$s_i + f - 1$$  \hspace{1cm} (12)

And $s_i$ can be the length of the signal or coefficients at different level. The coefficient at different level can be calculated by Eq. (13). $s_i$ is the length of the coefficients of the previous level.

$$s_i + f - 2)/2$$  \hspace{1cm} (13)

If the length of the signal is $n$ and the length of the filter is $f$, then the convolution leads to a signal with length equal to $n + f - 1$. After down-sampling, the length of 1st level coefficients is equal to $(n + f - 2)/2$. In Table 3, the computation units of direct DWT are listed. As IDWT reconstruction, the procedure includes firstly up-sampling and then convolution. The computation unit can be calculated by Eq. (14) which means the length after convolution the up-sampling sequences, where $s_j$ is the length of the signal or coefficient at different level and the results are listed in Table 4.

$$2^j s_j - 1 + f - 1$$  \hspace{1cm} (14)

For the DWT decomposition of the proposed modified method, the computation unit of different level is the same and is equal to the corrected terms at the right side, $f/2$. For IDWT reconstruction, the computation unit is also calculated by Eq. (14) but $s_j$ keeps constant at the different level and usually $s_j$ is set to the length of the coefficients at the desired level. While application, two points are considered: 1. length of the moving-horizon, 2. selection of the filter.

Selection of the filter Different wavelets show different filter bands in the frequency domain but the differences are quite small. The mean square errors for the reconstruction signals, which have the same original signal with random noises of fixed standard deviation, from different level of different Daubechies wavelets are plotted in the figure 3. From the figure, it tells that the performances are almost the same except the first two wavelets. So it can select the wavelets freely after Daubechies 6, but it does matters if the efficiency is considered.

Length of the moving-horizon According to Table 3 and Table 4, assuming that four-level of DWT decomposition is chosen, then the total computation units of direct DWT is equal to $(15n + 65f - 114)/4$.

For the proposed method, the total computation units is $(15n + 97f - 162)/8$ ($s_i$ in Eq. (14) is equal the length of the 4th level coefficients). Then the efficiencies of different Daubechies wavelets with corresponding lengths in the Tables 2 are calculated and the results are listed in Table 5. If the computation units are normalized, then the efficiencies are shown as Figure 4. It can see from the figure that at certain length there exist the best benefits.
Table 5 Efficiencies of different length of the moving horizon respect to different Daubechies wavelets

<table>
<thead>
<tr>
<th></th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>db02</td>
<td>0.5738</td>
<td>0.6703</td>
<td>0.7483</td>
<td>0.8007</td>
</tr>
<tr>
<td>db04</td>
<td>0.5062</td>
<td>0.6036</td>
<td>0.6959</td>
<td>0.7663</td>
</tr>
<tr>
<td>db06</td>
<td>0.4653</td>
<td>0.5573</td>
<td>0.6545</td>
<td>0.7363</td>
</tr>
<tr>
<td>db08</td>
<td>0.438</td>
<td>0.5232</td>
<td>0.6209</td>
<td>0.71</td>
</tr>
<tr>
<td>db10</td>
<td>0.4184</td>
<td>0.4971</td>
<td>0.5931</td>
<td>0.6866</td>
</tr>
<tr>
<td>db12</td>
<td>0.4037</td>
<td>0.4764</td>
<td>0.5697</td>
<td>0.6658</td>
</tr>
<tr>
<td>db14</td>
<td>0.3923</td>
<td>0.4597</td>
<td>0.5498</td>
<td>0.6471</td>
</tr>
</tbody>
</table>

Fig. 4. Efficiencies with normalized computation units

By integrating differential-algebraic (DAE) equations of a linear dynamic system between some time $t_0$ and $t_n$, we can get the following algebraic equations (i.e. Eqs. (15) and (16)). $h$ is the vector of state variables, $f$ is the vector of non-state variables, $A$ and $C$ are constant matrices from the algebraic part of the DAE.

$$\int_{t_0}^{t_n} \frac{dh}{dt} dt = A^{*} \int_{t_0}^{t_n} f dt$$

(15)

$$C^{*} \int_{t_0}^{t_n} f dt = 0$$

(16)

Let $Z_1 = \int_{t_0}^{t_n} \frac{dh}{dt} dt$, $Z_2 = \int_{t_0}^{t_n} f dt$, and we can rearrange the integrations and get the following matrix form in Eq. (17).

$$\begin{bmatrix} A & -I \\ C & 0 \end{bmatrix} \begin{bmatrix} Z_2 \\ Z_1 \end{bmatrix} = 0$$

(17)

It shows that the result is an algebraic constraint. In the following, the reconciliation procedure is to incorporate the integrating part by Simpson’s rule for numerical integration. Then the algebraic equations can be expressed as Eqs. (18) and (19).

$$Z_1 = Q_1^{*} F$$

(18)

$$Z_1 = Q_2^{*} H$$

(19)

Define new variable $H$ and $F$ which represent the collections of all measurements of all instruments during the integrating time interval $t_0$ to $t_n$. Finally, we obtain the algebraic constraint equation (Eq. 20) represented by $H$ and $F$ from the original DAE.

$$[A - I] \begin{bmatrix} Q_1 \\ 0 \end{bmatrix} + [0 \ Q_2] \begin{bmatrix} F \\ H \end{bmatrix} = 0$$

(20)

With the equality constraint, the reconciliation problem can be solved. As on-line reconciliation, one reconciled point is obtained in one moving-horizon.

Fig. 5 A four-tank system

4. EXAMPLE

A four-tank system shown in Figure 5 is illustrated as an example. The differential algebraic equation of this example is showed in Eq. (21). There are two main flows $f_5$, $f_6$ split into two branches apiece. The four branches, $f_1$, $f_2$, $f_3$, $f_4$ flow into four tanks respectively. Each tank has flow, $q_1$, $q_2$, $q_3$, $q_4$ out of it. The flow out of tank 3 is fed into tank 1 and the one out of tank 4 is fed into tank 2. Parameters of the process are listed in Table 6. From the analysis in section 2, wavelets “Daubechies 6” is selected and the moving-horizon is 74. The results are shown in Figures (6)-(8).

$$A \frac{dh_3}{dt} = -f_1 + q_1 + f_5$$

$$A \frac{dh_4}{dt} = -f_2 + q_2 + f_6$$

$$A \frac{dh_1}{dt} = -f_3 + q_3 + f_4$$

$$f_5 = f_1 + f_4$$

$$f_6 = f_2 + f_3$$

(21)

Fig. 6 Reconciled result of $f_1$
In this research, the wavelet filtering is used to apply in the on-line dynamic data reconciliation. In the theory, the shift property of the DWT is the basic idea to establish the method whose main purpose is to save the DWT computation efforts. A modified method is actually being applied which can save more computation efforts. The efficiencies of different Daubechies wavelets are discussed in order to decide the best performance match-up. The reconciliation performances are good.

![Fig. 7 Reconciled result of $h_i$](image)

**Table 6 Parameters of the four-tank system**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>State/Parameter</th>
<th>Value</th>
<th>Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h_0$</td>
<td>Nominal levels</td>
<td>[20.4; 20.4; 11.5; 11.5] cm</td>
<td></td>
</tr>
<tr>
<td>$a_i$</td>
<td>Area of the drain</td>
<td>[3; 3; 2; 2] cm$^2$</td>
<td></td>
</tr>
<tr>
<td>$A_{ri}$</td>
<td>Area of tank $i$</td>
<td>1000 cm$^2$</td>
<td></td>
</tr>
<tr>
<td>$f_i$</td>
<td>Flow into the tank $i$</td>
<td>[0.3; 0.3; 0.3; 0.3; 0.6; 0.6] cm$^3$/S</td>
<td></td>
</tr>
<tr>
<td>$T_i$</td>
<td>Time constants</td>
<td>[68; 68; 76.5; 76.5] S</td>
<td></td>
</tr>
<tr>
<td>$g$</td>
<td>Gravitation constant</td>
<td>981 cm/S$^2$</td>
<td></td>
</tr>
<tr>
<td>$\sigma_f$</td>
<td>Standard deviation of flow</td>
<td>0.015 cm$^3$/S</td>
<td></td>
</tr>
<tr>
<td>$\sigma_h$</td>
<td>Standard deviation of level</td>
<td>0.6 cm</td>
<td></td>
</tr>
</tbody>
</table>

![Fig. 8 Reconciled result of $q_i$](image)

**REFERENCE**


