Lightweight Signal Processing in Sensor Node for Real-time Traffic Monitoring

Yuhe Zhang¹,², Xi Huang¹,² and Li Cui¹,

¹ Inst. of Computing Technology, Chinese Academy of Sciences, Beijing, China, 100080
² Graduate School of the Chinese Academy of Sciences, Beijing, China, 100049
Tel: +86-10-6260 0724, Fax: +86-10-6256 2701
E-mail: {zhangyuhe, huangxi, lcui}@ict.ac.cn

Abstract—Wireless sensor network may be used for real-time traffic monitoring. In this paper, we describe the design and implementation of a practical wireless traffic information collection node (TICoN). We propose a specific lightweight adaptive signal processing algorithm based on matched filter, namely AEMFA, for traffic monitoring. We provide solutions to embed this signal processing algorithm in resource-limited sensor node, to reduce power consumption. We also propose a traffic volume detection algorithm, namely TVDA, based on AEMFA. Experimental results show that AEMFA algorithm performs well in sensor node. The algorithm is efficient both in traffic volume detection with an accuracy of 90%, and in energy saving up to 80%.

I. INTRODUCTION

With the rapid development of modern cities and fast increasing amount of vehicles, ITS has become crucial in many megalopolises. ITS is comprised of different elements [1], such as arterial management, freeway management, information management and so on. These elements are based on the road condition represented mainly by the traffic volume, vehicle velocity and lane occupancy. Greenshield’s quadratic model [2], as shown in Fig.1, indicates the relationship between traffic volume and velocity, which can further be used to estimate the state of road. Therefore, to obtain the information of traffic volume, velocity and time occupancy in real-time is essential for ITS.

The traffic monitoring system based on WSN technology has great advantages compared with the traditional systems [3], like deployment convenience, detection accuracy, and overall cost. As shown in Fig.2, the WSN-based system is comprised of wireless traffic information collection nodes (TICoN), wireless data aggregation node (DAN) and remote traffic monitoring center (RTMC). TICoN nodes collect and process real-time information in a distributed and coordinated fashion. They transmit the results to the DAN node. The DAN node uses the information to control the traffic light or transfers the information to the RTMC.

Fig. 1. Greenshields’s Quadratic Model, 1935

This work is supported in part by the National Basic Research Program of China (2006CB303000), National High Technology R&D Program of China (2006AA01Z215), NSFC general project (60572060).
little storage resource and computing resource are needed. It’s simple, but the cutoff of the spectrum may lead to the loss of some useful information. Therefore, it is not suitable for detecting very tiny signals.

In this paper, we propose a lightweight signal processing algorithms based on matched filter [7]. Matched filter is the optimal linear filter, which maximizing the signal-to-noise ratio (SNR) at the judgment point. It has been commonly used in radar and many other fields, like DSSS system in communication, watermark recognition, underwater acoustic signal processing, and nondestructive testing and so on. To the best of our knowledge, there hasn’t been research work for matched filter in resource-limited WSN node. We developed an adaptive embedded matched filter algorithm, namely AEMFA, for the real-time collection of traffic information.

The main contributions of this work are:
- We design and implement a traffic monitoring node based on WSN technology, which may obtain the information of road in real-time.
- We provided the matched filter for traffic monitoring and proposed a novel solution for the choice of reference signal introducing the curve-fitting method.
- We designed a lightweight embedded signal processing algorithm, namely AEMFA, to meet the requirement of real-time traffic monitoring. We proposed a TVDA algorithm for more accurate traffic volume detection.
- We provided the performance analysis of the algorithms in power efficiency and in detection accuracy with data collected from on road experiments.

This paper is organized as follows: The second section presents the hardware design of TICoN node. In the third section, we show the detailed studies of the adaptive embedded matched filter, namely AEMFA, and a further study of traffic volume detection. Experimental results are presented and analyzed in the fourth section. Finally, we conclude our studies and discuss the future work.

II. DESIGN AND IMPLEMENTATION OF SENSOR NODE

The TICoN node consists of four modules as shown in Fig. 3 (a), which are signal collecting module, signal processing module, wireless communication module and a power supply module. Fig. 3 (b) shows a photo of the node made in-house [8] which is composed of sensor and communication boards.

There is sensor, signal pre-processing unit, DC/DC converting unit and set / reset circuit on the sensor board. The resolution of the magnetic sensor (HMC1021Z, from Honeywell) is 85μGauss, with sensitivity of 1mV/V/Gauss [9]. There is a Wheatstone bridge within the sensor, and each leg of the bridge is a resistive strip made of nickel-iron thin film deposited on a silicon wafer. When the outside magnetic field changes, the resistance of each leg will change, which induces a change in the output voltage. The principle of vehicle detection is according to the distortion of magnetic field when a vehicle approaches.

However, the distortion of magnetic field aroused by the vehicle is very small. According to [10], it’s about 1mGauss when a vehicle passes at the distance of 7 meters. As a consequence, the output of sensor is 3μV or even lower. This tiny signal is difficult to be processed so that we need to amplify the signal before A/D conversion. We select INA128, a low-noise amplifier (from Bubb-Brown), as the pre-amplifier and a common INA126 as another amplifier on the next level. We choose MAX680 (from MAXIM) which generates ±6V from a 3V battery as power supply.

Another key part on the sensor board is the set / reset circuit. Any strong disturbance (10 Gauss or above) will change the sensor’s characteristics by upsetting or flipping the polarity of film magnetization [9]. The set / reset circuit is designed to solve this problem. When the sensor is disturbed by a strong magnetic field, a set and reset pulse will be applied to the sensor through the set / reset unit. The pulse lasts for 2μs with a current of 0.5A.

The communication board EZ210 is also made in-house [8], which is compatible with TinyOS [11]. There are microprocessor and communication chips on this board. The microprocessor runs at a low clock rate of 7.3728M typically.

III. LIGHTWEIGHT SIGNAL PROCESSING ALGORITHM

As already mentioned, since the TICoN nodes are fixed by the roadside for deployment convenience, they are easily to be interrupted. We define the magnetic field introduced by vehicles as signal fields, and all the other magnetic fields rather than the signal field as an interruption. These interferences are mainly electromagnetic interference. The response signal of vehicle decreases with its distance to the sensor node. If this small signal is disturbed by the electromagnetic interference, the information of this vehicle may be missed. The information missing is called false negative detection, which will reduce the accuracy of traffic volume.

Besides, for the detection of vehicle velocity, a pair of synchronized nodes is used, like Nodes A and B shown in Fig. 2. They are fixed apart along the road and response to the passing vehicles independently. The key issue is to determine the interval between the two response signals, which is difficult when the signal is badly interrupted.

Therefore, efficient signal processing methods are needed to filter the interference. The feature of electromagnetic interference is its wide range spectrum, which may be recognized as White Gaussian noise. Some commonly used filtering methods, such as moving average and weight moving average methods, are not suitable for processing tiny signals as in this case. And for velocity detection, where the accurate time for peak / valley of the response signal is required, the common methods are not suitable, either. Here we designed an adaptive embedded matched filter algorithm, namely AEMFA, to process the original signal in a more efficient way.
A. Matched Filter

Matched filter [7] is the optimal detector for an object. It makes the useful signal \( s(t) \) strengthen, whilst restraining the additive noise \( n(t) \). The function of matched filter, that is, \( h(t) \), can be expressed as \( s(t-t) \). Matched filter can be recognized as the cross-correlation of original signal \( (s(t)+n(t)) \) and \( s(t) \), as Equation 1 shows. After the correlation, \( \text{corr}(s(t), s(t)) \) reaches the peak value at the judgment point, whilst \( \text{corr}(s(t), n(t)) \) keeps the lowest. The maximum SNR may be achieved at this point.

\[
( s(t) + n(t) ) \ast h(t) = \text{corr}(s(t), s(t)) + \text{corr}(n(t), s(t)) \tag{1}
\]

B. Reference Signal for Matched Filter

The primary issue for matched filter is the choice of reference signal \( s(t) \). The criterion in choosing \( s(t) \) is to get the least correlation of noise. Fig. 4(a) shows a typical original signal pattern collected by the sensor node within its response time. The magnetic field downward, which is the response signal, is caused by a passing vehicle. Instead of using the original response signal, we adopted the curve-fitting technology to get the reference signal. Three kinds of function are employed. They are the polynomial function, the addition of sine and proportion function, and the Gaussian function. Fig. 4(b) shows a typical original signal pattern collected by the sensor node within its response time. The magnetic field downward, which is the response signal, is caused by a passing vehicle. Instead of using the original response signal, we adopted the curve-fitting technology to get the reference signal. Three kinds of function are employed. They are the polynomial function, the addition of sine and proportion function, and the Gaussian function. Fig. 4(b) shows a typical original signal pattern collected by the sensor node within its response time. The magnetic field downward, which is the response signal, is caused by a passing vehicle. Instead of using the original response signal, we adopted the curve-fitting technology to get the reference signal. Three kinds of function are employed. They are the polynomial function, the addition of sine and proportion function, and the Gaussian function. Fit-

We express the Gaussian function using Equation 2, where \( a_i \) represents the pattern of the response signal. This function is symmetry to Y-axis. We select \( f_G(x) \) (i=1~N) as the reference signal given \( x_i \) evenly increasing from \((-N^*SR/2) \) to \((-N^*SR/2) \). Here, \( SR \) represents the sample rate of the TICoN node. The reference signal is represented as \( \text{ref}(t:N) \).

\[
f_G(x) = e^{-\frac{x^2}{2}} \tag{2}
\]

C. Adaptive Embedded Matched Filter Algorithm (AEMFA)

There are two technical challenges to embed the matched filter algorithm in the node. Firstly, we must take the limitation of processing and storage into account. The microprocessor is 8-bit and it only possess integer calculator physically. The float calculation is implemented by software, which is time-consuming. Secondly, the power consumption must be low for the practical use. We introduce the solutions to these challenges in two phases. They are the embedded matched filter for limited resource and the adaptive processing to meet the power requirement.

C.1. Phase 1: Embedded Matched Filter (EMF)

Before the design of embedded matched filter, we analyze the operations probably used and the corresponding execution time. The method for the measurement is introduced in /A. As Table 1 listed, it will take the microprocessor dozens or hundreds or even thousands of clock cycles to finish a single operation. The microprocessor is always run at a low clock rate, which is 7.3728M for TICoN node.

<table>
<thead>
<tr>
<th>Calculation</th>
<th>Clock Cycles</th>
<th>Calculation</th>
<th>Clock Cycles</th>
</tr>
</thead>
<tbody>
<tr>
<td>16 bit Integer Addition</td>
<td>6</td>
<td>Float Addition</td>
<td>153</td>
</tr>
<tr>
<td>16 bit Integer Multiplication</td>
<td>14</td>
<td>Float Multiplication</td>
<td>367</td>
</tr>
<tr>
<td>32 bit Integer Addition</td>
<td>12</td>
<td>Float Division</td>
<td>502</td>
</tr>
<tr>
<td>32 bit Integer Multiplication</td>
<td>60</td>
<td>Float Square Root</td>
<td>1769</td>
</tr>
<tr>
<td>32 bit Integer -&gt; Float</td>
<td>227</td>
<td>Float -&gt; 32 bit Integer</td>
<td>134</td>
</tr>
<tr>
<td>32 bit Assignment</td>
<td>17</td>
<td>16 bit Assignment</td>
<td>9</td>
</tr>
</tbody>
</table>

The requirement of the embedded signal processing algorithm is that it must complete in one sample cycle, which is 10 ms for our node. The original operations of matched filter, as Equation 3 shows, are mainly float-point. There are totally \((8*N-4) \) times float-point addition, \((4*N) \) times float-point multiplication, \((N+5) \) times float-point division, \( N \) times float-point square root calculation, \((2*N-1) \) times transformation from integer to float-point and \((N-1) \) times integer addition.

Assuming \( N \) equals to 160, it will take the microprocessor 118 ms to finish one matched filter, which is clearly unsuitable for real-time detection.

\[
corr(\text{ref}, \text{signal}) = \frac{1}{N} \sum_{i=1}^{N} \bigg( \text{ref}(i) - \mu(\text{ref}) \bigg) \bigg( \text{signal}(i) - \mu(\text{signal}) \bigg)
\]

\[
E(\text{ref}) = \frac{1}{N} \sum_{i=1}^{N} \text{ref}(i), \quad \text{var}(\text{ref}) = \frac{1}{N} \sum_{i=1}^{N} \left( \text{ref}(i) - E(\text{ref}) \right)^2
\]

\[
E(\text{signal}) = \frac{1}{N} \sum_{i=1}^{N} \text{signal}(i), \quad \text{var}(\text{signal}) = \frac{1}{N} \sum_{i=1}^{N} \left( \text{signal}(i) - E(\text{signal}) \right)^2
\]

To reduce the amount of calculation, we introduced several improvements through the following ways.

Firstly, we eliminated the duplicate calculation and keep the useful history results in storage. For example, the calculation for \( E(\text{signal}) \) is duplicated between adjacent periods, so that we may store the sum of \( \text{signal}(1:N) \) as \( \text{lastSumOfSignal} \) and \( \text{signal}(I) \) as \( \text{lastSignal} \) temporarily. When a new data comes, the sum can be computed as in Equation 4. As to the node, although the storage resource is still limited, it is not as critical as for the computing resource.

\[
\sum_{i=1}^{N} \text{signal}(i) = \text{signal}(1) + \text{signal}(2) + \ldots + \text{signal}(N) = \text{lastSumOfSignal} + \text{signal}(N) - \text{lastSignal}
\]

Secondly, an algorithm [12] is used for the square root calculation. This algorithm can compute the invert square root with several multiplication, addition and shift operations.

Thirdly, we make the best use of the symmetry feature of the reference signal. It can be found that the value of \( \text{ref}(i) \) and \( \text{ref}(N-i) \) are equivalent. So that Equation 5 is introduced. This operation will cut down \((N/2) \) integer multiplication.

Fig. 4. Original Signal Fitting using Gaussian Function

(a) Original Signal

(b) Fitting Result

Fig. 4. Original Signal Fitting using Gaussian Function
\[ \sum_{i=1}^{N} r[i] = \sum_{i=1}^{N} r[i] \cdot (\text{signal}(i) + \text{signal}(N-i)) \]  

Finally, we introduce a circular array instead of a common array for the original signal. As Equation 3 shows, the correlation happens between a constant reference array and a variable signal array. \text{signal}(1:N) is comprised of the new sample data and its former (N-1) data. Thus to update the signal array, it needs N-times operations of read-write memory, which are time-consuming. Our circular array has two position indicators, one of which indicates the head position of the circular array, namely \text{headPosOfSignal}, while the other one indicates the tail position, namely \text{tailPosOfSignal}. The relationship between them is shown in Equation 6. The indicators move circularly from 1 to N as a new data comes.

\[ \text{tailPosOfSignal} = \left( \text{headPosOfSignal} + (N-1) \right) \mod N \]

In summary, there are totally 5 steps for the embedded matched filter running in real-time.

- **Step 1**: store the newly sampled data in the circular array.
- **Step 2**: update \( E(\text{signal}) \), \( \text{var}(\text{signal}) \) and history data.
- **Step 3**: compute the inverse square root of \( \text{var}(\text{signal}) \).
- **Step 4**: calculate the sum of \( (\text{signal}(i) * r[i]) \)
- **Step 5**: calculate the final matched filter result, namely \( \text{MRResult} \), using the above results.

### C.2. Phase 2: Adaptive Signal Processing

Here the motivation is to improve the embedded matched filter to meet the requirement of power consumption. As we mentioned in phase 1, the node can carry out the embedded matched filter in real-time. However, if it is executed whenever a sample data emerges, the continuous power consumption of computation will be at a high level. If we decrease or stop the processing task when there is no useful signal, it will improve the overall power management.

A threshold is set for vehicle detection, namely \( VThr \). If the \( \text{MFResult} \) is higher than \( VThr \), it means a vehicle passes. From the experimental results, it is noticed that the part of \( \text{MFResult} \) which is higher than \( VThr \) lasts for 0.2s at least. Therefore, an adaptive matched filter is designed to adapt the changing signal. As Algorithm 1 described, a timer is set for preliminary decision to avoid unnecessary processing. The signal processing period, namely \( \text{TimeThr} \), is set to 0.1s so that it won’t miss the peak of a response signal. For preprocessing, we only carry out step 1 and 2 of phase 1 to prepare the essential variables for the matched filter. When \( \text{MFResult} \) is higher than \( VThr \), the signal processing continues until the peak is confirmed. Then, a packet containing the peak time of the response signal is sent out wirelessly, which can be used both for traffic volume and velocity detection.

### D. Traffic Volume Detection Algorithm (TVDA)

Traffic volume is the total number of vehicles passed per unit time. It is derived from counting vehicles. As the response time of response signal varies with the type and velocity of vehicles, sometimes, we need to use multiple matched filters to achieve the best detection. We use wide and narrow to express the length of response time. The criterion of choosing reference signals for traffic volume detection is that the correlation values of the widest reference signal and the widest signal, the narrowest reference signal and the narrowest signal, and the two adjacent reference signals, are all larger than \( VThr \). Experimental data showed that most of the response time is between 0.7s and 4s. We thus adopted 2 Gaussian functions as reference signals, that is, the 0.85s and the 3.4s respectively. The procedure of TVDA is described in Algorithm II.

#### Algorithm I. AEMFA

1. initialize \text{TimeThr}, \text{VThr}, \text{IsAVehicle}, \text{PeakMode}, \text{start Timer}
2. sample and A/D convert and currentTime ++
3. if \( \text{TimeThr} \) or \( \text{PeakMode} == \text{True} \) then
4. \( \text{MFResult} \) and set \( \text{Timer} = 0 \)
5. if \( \text{MFResult} > \text{VThr} \) \& \& \( \text{IsAVehicle} == \text{True} \) and \( \text{PeakMode} == \text{False} \) then
6. \( \text{IsAVehicle} = \text{True}, \text{PeakMode} = \text{True}, \text{lastResult} = \text{MFResult} \)
7. else if \( \text{MFResult} < \text{VThr} \) \& \& \( \text{IsAVehicle} == \text{True} \) then
8. \( \text{IsAVehicle} = \text{False} \)
9. else if \( \text{PeakMode} == \text{True} \) \& \& \( \text{lastResult} = \text{MFResult} \) then
10. \( \text{lastResult} = \text{MFResult} \)
11. else if \( \text{PeakMode} == \text{True} \) \& \& \( \text{lastResult} = \text{MFResult} \) then
12. \( \text{PeakMode} = \text{False}, \text{peakTime} = \text{currentTime} - 1, \text{Report peakTime to the DAN node} \)
13. end if
14. else
15. Preprocess; \text{Timer} ++;
16. end if

#### Algorithm II. TVDA

1. initialize \text{IsAVehicle}
2. \( \text{MFResult}(1) = \text{Corr}[	ext{signal}(m-n+1:m), r[1:n]) \)
3. \( \text{MFResult}(2m) = \text{Corr}[	ext{signal}(m-n+1:m), r[1:n]) \)
4. if \( \text{IsAVehicle} == \text{False} \) then
5. if \( \text{MFResult}(m-1) < \text{VThr} \) \& \& \( \text{MFResult}(m) > \text{VThr} \) or \( \text{MFResult}(2m-1) < \text{VThr} \) \& \& \( \text{MFResult}(2m) > \text{VThr} \) then
6. \( \text{IsAVehicle} = \text{True}, \text{Report} \text{"There is a vehicle\"} \)
7. end if
8. else if \( \text{MFResult}(1) < \text{VThr} \) \& \& \( \text{MFResult}(m) < \text{VThr} \) or \( \text{MFResult}(2m-1) < \text{VThr} \) \& \& \( \text{MFResult}(2m) < \text{VThr} \) then
9. \( \text{IsAVehicle} = \text{False} \)
10. end if

By employing the TVDA algorithm, a local decision to judge whether there is a vehicle is made. If there is a vehicle passing, the node will send out a message. This TVDA algorithm is designed for traffic volume detection. However, the same idea can be expanded in velocity detection.

### IV. PERFORMANCE EVALUATION

We designed three kinds of experiments to evaluate the performance of the proposed algorithms. First, we evaluated the embedded matched filter in the metric of execution time. Then, we studied the total power consumption for three signal processing methods. Finally, we compared the accuracy of the TVDA algorithm and BPPA algorithm for traffic volume detection.

#### A. Execution Time of Embedded Matched Filter

The execution time for each step of EMF was measured and the result showed that the EMF could perform well in our sensor node.
We adopt Timer / Counter 1 of the microprocessor to evaluate the total time and the clock of MCU is chosen as the clock source. With initial value 0, Timer1 is started before each step and read after the step. Its value is recognized as the total clock cycles needed for this process. The atomic statement of TinyOS is used to avoid interruption. We also measured the execution time of sample. The measurement results with N equals to 160 is listed in Table 1

<table>
<thead>
<tr>
<th>Process</th>
<th>Clock Cycles</th>
<th>Execution Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>54</td>
<td>0.007</td>
</tr>
<tr>
<td>Step 2</td>
<td>2149</td>
<td>0.292</td>
</tr>
<tr>
<td>Step 3</td>
<td>12174</td>
<td>1.651</td>
</tr>
<tr>
<td>Step 4</td>
<td>2009</td>
<td>0.272</td>
</tr>
<tr>
<td>Step 5</td>
<td>1688</td>
<td>0.229</td>
</tr>
<tr>
<td>Embedded Matched Filter</td>
<td>18074</td>
<td>2.451</td>
</tr>
<tr>
<td>Sample</td>
<td>2370</td>
<td>0.322</td>
</tr>
</tbody>
</table>

From Table 1, we can see that the amount of time for sample and signal processing is 2.73 ms, which is far less than our sample period which is 10 ms. The time for preprocessing containing step 1 and 2 is 0.299 ms for AEMFA algorithm.

B. Power Consumption of AEMFA Algorithm

The proposed EMF and AEMFA methods were compared with the centralized processing scheme. The simulation result of power consumption showed that the AEMFA method had advantage in power-saving.

The TICoN node is operated at 2.7~5V, typically at 3V. The power consumption of the node alters according to different working states. Table III lists the current consumption for each component.

<table>
<thead>
<tr>
<th>Mode</th>
<th>Current (mA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td></td>
</tr>
<tr>
<td>Active</td>
<td>6.9</td>
</tr>
<tr>
<td>Idle</td>
<td>3.3</td>
</tr>
<tr>
<td>Power_Save</td>
<td>0.6</td>
</tr>
<tr>
<td>Radio</td>
<td></td>
</tr>
<tr>
<td>Listen</td>
<td>7.5</td>
</tr>
<tr>
<td>Transmit (-20dBm)</td>
<td>5.3</td>
</tr>
<tr>
<td>Transmit (0dBm)</td>
<td>8.3</td>
</tr>
<tr>
<td>Transmit (10dBm)</td>
<td>21.4</td>
</tr>
</tbody>
</table>

When the signal processing algorithm runs, the microprocessor is in active mode and other modules are shut down. The microprocessor is in Idle state when the node is in sample or transmission mode. The transmission power is fixed to 10dBm in our test, with a communication range of 200m outdoors. The data rate is set to 76.8 kbps using Manchester coding.

In the centralized scheme, the TICoN node sends out the original data every ten samples, and the packet length is 42 bytes. While for EMF and AEMFA, the node sends out a packet whenever it detects the peak time of response signal, and the packet is 24 bytes. Fig. 5 (a) shows the comparison results of power consumption as a function of traffic volume increasing. Fig. 5 (b) shows the power consumption at different states when the traffic volume is at 1000 per hour. These states are: sampling, preprocessing, matched filter, transmitting and sleeping.

From Fig. 5 (a) and (b), we can reach the following conclusions. The power consumption of the centralized scheme is independent of traffic volume because the node sends out original data periodically without signal processing. Communication takes about 90% of the total power in this scheme. The power consumption of the EMF almost equals to the centralized scheme and matched filter is the most time-consuming state as it is carried out for each original data. The most power-saving method is the AEMFA, whose power consumption reduced to about 22% of the centralized scheme and about 28% of the EMF.

C. Traffic Volume Detection

The proposed TVDA algorithm was compared with the original band-pass filter algorithm, namely BPFA, in traffic volume detection. The experimental results showed that the TVDA algorithm could reach higher detection accuracy.

We deployed the traffic monitoring system by a two-way double-lane road. TICoN nodes monitor two lanes simultaneously, as shown in Fig. 6 (a) and (b). After DAN node receives the original signal, it sends the information to the RTMC.
Two group’s experiments were carried out in parallel based on the two algorithms. Fig. 7 (a) and (b) show the results. To evaluate the accuracy of the system, the total number of vehicles actually passed during the test was measured manually at the same time.

The curves on the top parts in Fig. 7 (a) and (b) are original signals, the middle parts are the detection results using BPFA, while the lower parts are the detection results using TVDA. False negative detections are circled by solid loops, whilst false positive detections are circled by dotted loops. The total statistic results are listed in Table IV. It can be seen that TVDA achieves higher accuracy than BPFA.

![Traffic Monitoring Result by BPFA](image)

![Traffic Monitoring Result by TVDA](image)

![False Negative Detection](image) ![false positive detection](image)

(a) Group 1

![Traffic Monitoring Result by BPFA](image)

![Traffic Monitoring Result by TVDA](image)

![False Negative Detection](image) ![false positive detection](image)

(b) Group 2

Fig. 8. Comparison of BPFA and TVDA in Traffic Detection

<table>
<thead>
<tr>
<th></th>
<th>50</th>
<th>3</th>
<th>4</th>
<th>78.1%</th>
<th>87.5%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group 1</td>
<td>32</td>
<td>5</td>
<td>0</td>
<td>84.4%</td>
<td>100%</td>
</tr>
<tr>
<td>Group 2</td>
<td>32</td>
<td>7</td>
<td>4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

V. CONCLUSION AND FUTURE WORK

This paper presented a practical lightweight signal processing algorithm based on matched filter, namely AEMFA, embedded in WSN nodes for real-time traffic monitoring. The hardware design and implementation of the TICoN node were also introduced. Detailed performance examinations in the metrics of execution time and power consumption were carried out. Experimental results showed that the AEMFA algorithm performs well and saves energy. The traffic volume detection result indicated that TVDA algorithm achieves better detection accuracy compared with the BPFA algorithm.

In the future, we intend to further improve the detection accuracy of traffic volume and velocity detection using AEMFA algorithm. We will also investigate the cooperation of distributed sensor nodes in information processing.

ACKNOWLEDGMENT

The authors wish to acknowledge the helpful discussion with Transportation Research Center, Road and Transportation Lab of Beijing University of Technology.

REFERENCES