Preliminary study of low-cost GPS receivers for time synchronization of wireless sensor networks

R. Kim,*a T. Nagayama, b H. Jo, a B. F. Spencer Jr. a

aUniversity of Illinois Urbana Champaign, 205N Mathews Ave., Urbana, IL USA 61802;
 bUniversity of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan 113-0033

ABSTRACT

Growing public concern regarding the health of the aging civil infrastructure has spurred research in structural health monitoring (SHM). Recent advances in wireless smart sensor (WSS) technology has significantly lowered the cost of SHM systems and resulted in WSS being successfully implemented at full-scale. However, assuring accurate time-synchronized WSS nodes in a network is still a challenging problem. Generally, WSS synchronization is realized by communicating a sensors’ CPU clock information over the network. However, such a synchronization approach becomes more challenging as the network size increases. Reliable communication is not easily achieved due to longer communication distance, larger numbers of sensors, and complexity of a distributed sensor network. Moreover, CPU clocks may not be sufficiently reliable for accurate time-synchronization due to substantial tolerance errors in crystal and/or temperature effects. In this study, the use of low-cost GPS receivers for time synchronizing WSSs is explored to resolve these issues. GPS sensors offer the potential to provide high-accuracy synchronization --- nano-second level even with low-cost GPS receivers. The GPS-assisted time synchronization approach overcomes network communication limitations to realize time-synchronization in large-scale networks of WSS.

Keywords: Low-cost GPS, Time synchronization, Wireless sensors, Structural Health Monitoring

1. INTRODUCTION

Detecting damage, determining load bearing capacity, and assessing remaining life are primary goals of structural health monitoring (SHM). Advanced MEMS technology greatly reduced the cost of sensors, and introduced various types of sensors to realize SHM networks. For several decades, wired sensor applications in SHM were widely used. However, wired sensors have proven to be quite expensive for SHM of civil infrastructure, primarily due to the cost of cabling and installation. Moreover, centralized data collection intrinsic wired sensor networks to results in a single point of failure for the SHM system.

Alternatively, wireless smart sensors (WSS) offer the potential to significantly reduce the cost of SHM systems by eliminating the need for cables and facilitating installation. Moreover, WSS have the ability to perform on-board computation. Rather than having enormous amounts of raw data collected centrally, users can configure the nodes to collect only data of significance/interest. As a result, significant research interest has been directed to SHM using WSS.

One of the issues that is critical to successful use of WSS for SHM is time synchronization. To determine an appropriate time synchronization strategy, the specific characteristics of sensors and the level of time synchronization precision must be taken into consideration. The flooding time synchronization protocol (FTSP) is implemented in the Illinois Structural Health Monitoring Project (ISHMP) Service Toolsuite using MEMSIC’s Imote2 processor. This implementation of the FTSP provides time synchronization precision of approximately 100 μsec. However, the assumption that all of the sensor nodes can communicate with each other limits application in complex, large-scale WSSN (e.g., for large civil infrastructure).

This paper studies the possibility of utilizing the global positioning system (GPS) pulse per second (PPS) signal to trigger and synchronize networks of WSS. First, a possible application of a low-cost GPS receiver to control the wired sensor network is examined with high sampling-rate data acquisition. Then, implementation of high-accuracy time synchronization strategies using low-cost GPS sensors is investigated for wireless smart sensor networks (WSSN).
2. BACKGROUND

2.1 Time Synchronization Algorithms for Wireless Smart Sensors

The need for a common clock for the nodes in a distributed system, such as wireless smart sensor network (WSSN), led to the development of various time synchronization protocols. Such protocols each focus on different network needs, for example: 1) accuracy of the time synchronization, 2) efficiencies in power usage, and 3) efficiency in required memory. The Network Time Protocol (NTP) uses an external time standard to statistically analyze round-trip message times [1]. NTP provides accurate, secure, and robust time synchronization but requires a high power demand. The Reference Based Synchronization (RBS) algorithm broadcasts a reference message to multiple receivers in the network [2]. RBS uses a proactive technique, which seeks the information only when it is needed, and thus reduces energy consumptions. The Sichitiu and Veerarittiphan method [3] uses two algorithms called mini-sync and tiny-sync. This method provides tight, deterministic synchronization with low storage demand. Building a common time line within the WSSN is critical, yet challenging. Among the various methods, depending on the needs of the WSS purpose, a suitable protocol needs to be established.

2.2 FTSP (Flooding Time Synchronization Protocol) for ISHMP Application

FTSP broadcasts a radio message that contains sender’s global time stamp. On receiving the message, the receiver also records the local time stamp and calculates the difference between sender’s global time and receiver’s local clock. Clock delay is estimated by calculating the differences. FTSP time-stamping effectively calculates deterministic delay and also reduces jitter from non-deterministic delays. However, FTSP using delayed time is sufficient for synchronizing the network only if the local clocks have exactly the same and constant clock rate. The clock rate drift in a WSS can differ up to 40 μsec/sec. Maroti et al. [5] introduced a method to determine the pairwise clock drift offline by applying linear regression to the multiple time stamps recorded. Nagayama [6] implemented FTSP on the Imote2, estimating the time synchronization error, message time delay, and individual clock drifts. To estimate time delay, the global time sent, identified as $t_{send}$, local time that receiver obtains the message, $t_{receive}$, and finally delays (interrupt handling, encoding, decoding delays etc.) are identified in $t_{delay}$. Then, the offset time $t_{offset}$ for delayed time is defined as $t_{offset} = t_{receive} - t_{send} - t_{delay}$. Using the FTSP application on the Imote2, the time synchronization error showed zero mean, with small variance and upper bound of 80 μsec (see Figure 1). This implementation of FTSP is adopted for time synchronization in the ISHMP Services Toolsuite.

![Time synchronization error estimation](image)

Figure 1 Time synchronization error estimation.

2.3 Importance of Highly Accurate Time Synchronization in Structural Health Monitoring (SHM)

Regardless of the efforts on time synchronization in WSS, time synchronization protocols cannot escape from several challenges; (i) Fail in communication while sending and receiving time stamp packets, (ii) time delay and drift error in time synchronization are those examples. Communication failure cause loss of data and time delay and drift error cause data corruption. However, in SHM even several msec of time synchronization error can lead to the significant confusion in analyzing a structure because many damage detection algorithms examine changes in natural frequencies and mode
shapes before and after damage. For example, when 2 msec of time delay was introduced in a four-story building, a significant change in phase was observed. Moreover, as shown in Figure 2, frequency increases the phase error increased linearly.

Nagayama et al (2007) reported the other issues on time synchronization errors such that sampling rate drifts also significantly distorts the phase information and data loss adds noise in PSD [7].

3. PROTOTYPE SYSTEM

3.1 Clock Accuracy of a Low-cost GPS Receiver

A GPS receiver provides a timing pulse, one PPS signal. When a GPS receiver is locked with satellites, the PPS signal outputs globally synchronized Coordinated Universal Time (UTC) information. Coarse/Acquisition code, which contains the PPS information, is continuously broadcasted at 1.023 MHz from the satellites and is generally assumed to have approximately a 1% error, results in a GPS receiver time error of 10 nanoseconds or less. Ten nanosecond clock errors cause about 2.93-m error ($3 \times 10^8$ m/s / $1.023 \times 10^9$ Hz x 0.01 = 2.93m) in location estimation; however, it is still highly accurate for the time-synchronization of WSSN, considering civil structures’ fundamental frequencies typically lie below 10Hz [8]. In this section, the feasibility of low-cost GPS receiver for time-synchronization is explored, particularly using two low-cost GPS modules [9] shown in Figure 3, and the accuracy of the GPS receivers are examined. The modules are set so that the voltage of the 1PPS signal changes to 3V when UTC clock ticks and changes back to 0V after 0.1 seconds.

Two GPS receivers were placed outdoor and the PPS pulses are collected at 112400 Hz for 60 seconds using VibPilot DAQ [10] when receivers were in locked status. Figure 4 shows a typical set of pulse signal with a built in low pass filter. This figure indicates that individual GPS receivers are in near perfect time synchronized status when they are in locked position. A promising result can be drawn here that high-accuracy clock synchronization can be obtained with a low-cost GPS receiver.
4. PROTOTYPE SYSTEMS

4.1 Network Control Using Pulse Signal

One way to avoid building a large WSSN of low efficiency is to divide the WSSN into several sub-networks. However, dividing network may bring another challenge of time-synchronizing individual sub-networks. Alternatively, GPS PPS signal can be used to create an event to trigger the time synchronization of the sub-networks.

Figure 5 shows a LABVIEW prototype program to implement the concept of PPS signals triggering sensing events. UTC time corresponding to the start of sensing, the number of samples, and sampling rate are specified prior to execution. Once the program starts, a port is opened for NMEA GPS signals and the received signal is decoded for UTC time. If the received UTC time becomes larger than the specified value, the next PPS rise event triggers recording acceleration for the given number of samples at the sampling rate. Two sets of EVT – 5T GPS receivers [11], two tri-axial accelerometers (SD2422), and two data acquisition systems (NI-DAQ usb 6009) with the sampling rate of 1000 Hz and sampling time of 8 sec are used to verify the algorithm. The accelerometers were installed on a table, which was excited by tapping.

![LABVIEW diagram and from panel.](image)

Collected data is further investigated to evaluate the time synchronization of two systems (Figure 6). Given that the excitation input were identical, the phase of transfer functions can be used for the validation of the time synchronization. When the response of the first accelerometer is \( x(t) \), second accelerometer \( y(t) \) will be \( y(t) = x(t - \tau) \), where \( \tau \) is a constant time synchronization error. Then, Fourier transform of \( x(t) \) and \( y(t) \) will be as follows:

\[
X(\omega) = \int x(t)e^{-j\omega t}dt
\]

(4)
Let $t - \tau = t'$, then we can rewrite equation (5)

$$Y(\omega) = \int x(t - \tau)e^{-i\omega t}dt$$

(5)

Using $X(\omega)$ and $Y(\omega)$, we can obtain an expression for cross spectral density and its phase.

$$P_{XY} = E[X^*(\omega)Y(\omega)] = E[X^*(\omega)X(\omega)e^{-i\omega\tau}]$$

(7)

$$\angle P_{XY} = \angle E[e^{-i\omega\tau}]$$

(8)

Equation (8) indicates that the phase angle of the cross spectral density is only function of time delay, which is the time synchronization error ($\tau$).

$$\tan \alpha \approx \frac{\angle \text{Phase}}{\omega} = \frac{\tau}{\omega}$$

(9)
4.2 Enhancing Time Synchronization Error Using Low Sampling Rate

In order to use low sampling rate and still to have accurate time synchronization, an idea to filter the PPS signal is proposed. Fundamental concept is shown in Figure 8. With the filter that soothes rectangular PPS pulse, we can have more than 1 intermediate point between 0 V and 3 V. Also with the known design of the filter, the filter delay can be calculated to capture the point when the pulse arises at the original PPS.

Here, 1st order step response function is used as the smoothing filter (Figure 9 (a) and equation (10)).

\[
X(t) = 1 - \exp(-\omega(t - \tau)). \tag{10}
\]

\[
1 - X(t) = \exp(-\omega(t - \tau)). \tag{11}
\]

Where, \(\omega\) is filter design property and \(\tau\) is filter delay. An advantage of using the 1st order filter is that by rewriting equation 10, linear relationship in equation (12) can be obtained by taking logarithm in both side of equation (11). Thus, for the intermediate points, by taking logarithm relationship and by linear fitting those points, \(\omega \cdot \tau\) can be obtained as shown in equation (12) (see Figure 9 (b)),

\[
- \log(1 - X(t)) = \log(\exp(-\omega(t - \tau))) = \omega(t - \tau). \tag{12}
\]
Using Matlab® simulink, a simulation model was ran to validate the 1st order filter approach. When PPS signal was collected with VibPilot, signal to noise ratio (SNR) is as small as $2.01 \times 10^{-4}$ Figure 10.

In the simulation model, SNR of 0.05 and $2.01 \times 10^{-4}$ were used. The purpose of introducing noisier signal to the simulation was to generalize the 1st order filter method. 1 PPS signals with known filter delay ($\tau_{\text{exact}}$) was generated 100–500 times, and by linear fitting the intermediate points using equation (12), $\tau_{\text{simulation}}$ and errors to $\tau_{\text{exact}}$ are calculated. For the noise level 0.05 case, 1st order filter approach gives $\tau_{\text{simulation}}$ error within 40 micro seconds. Using ‘difitool’ in Matlab®, Gaussian distribution was found with mean ($\mu = 4.11$) and standard deviation ($\sigma = 11.54$). For $6.01 \times 10^{-4}$ case, $\tau_{\text{simulation}}$ was always identical to the $\tau_{\text{exact}}$. (Figure 11) The result implies that the 1st order filter approach is valid to calculate the filter delay with the given noise level of the GPS PPS signal.
5. CONCLUSION

Many SHM applications deal with large structures that are composed of low fundamental frequencies. Monitoring the structures requires accurate system identifications such as natural frequencies, damping ratios, and mode shapes. In order to satisfy those requirements using WSSN, accurate time synchronization is essential. Large errors in time synchronization affect modal analysis, in particular, mode shape estimation. Current time synchronization protocols have reasonable time synchronization error, but message transfer from the base station to all of the remote nodes is needed and thus requires WSSN that covers large areas. Dividing a SHM network into several sub-networks also require time synchronization of multiple base stations.

Alternative approach of time synchronizing WSSN using GPS receiver and PPS signal is proposed in this study. When GPS receivers are locked with satellites, PPS signals show nanosecond-level time-synchronization error with multiple GPS receivers. An algorithm that uses both UTC information and PPS signal to create events that triggers two networks in synchronized manner has been verified. The networks are well synchronized under the limitation due to the sampling rate. However, capturing the exact PPS requires WSSN use a high sampling rate, which necessitates large power consumption. An approach using a filtered PPS is presented. The 1st order filter show linear relationship within the smoothed filter and also show agreeable estimation of filter delay with the noise level of the PPS signals.

In this paper, feasibility of using PPS signal on controlling WSSN is verified. Implementing the applications to meet with current wireless sensors will be needed for the further work.
REFERENCES


