A Data-driven Synchronization Technique for Cyber-Physical Systems

Terrell R. Bennett
University of Texas at Dallas
tbennett@utdallas.edu

Nicholas Gans
University of Texas at Dallas
ngans@utdallas.edu

Roozbeh Jafari
University of Texas at Dallas
rjafari@utdallas.edu

ABSTRACT

In this paper, we present a method to synchronize data from multiple sensors in a cyber-physical system without any software or hardware modifications to the sensors. This method allows for synchronization of low-power embedded systems in heterogeneous sensor networks, regardless of accuracy of individual sensor clocks by using the events in the physical world to drive the synchronization in the cyber world. We propose two methods to select portions of sensor data streams to drive the synchronization: one leveraging the notion of known templates and the other using an information theoretic approach. Using the events as well as cues from the delay models, we determine alignment points between the data streams. These alignment points are used to synchronize the data. This novel approach is based solely on the sensor data for synchronization, and it can be applied post-deployment on systems of heterogeneous sensors that are not well designed and lack effective synchronization. Experiments show an average accuracy improvement from ~12000ppm to ~2400ppm for a template based-method and from ~12000 to ~277ppm and ~445ppm for information theoretic methods when comparing the synchronized (corrected) clock data to an ideal clock source.

Categories and Subject Descriptors

C.3 [**Special-purpose and application-based systems**]

General Terms

Algorithms, Reliability, Experimentation.

Keywords

Cyber-physical systems, synchronization, alignment, sensor networks, data-driven.

1. INTRODUCTION

The number of electronic sensors is rapidly growing, and we will soon be dealing with trillions of sensors and actuators deployed around the world in a global cyber-physical system [1]. Data synchronization between sensors in such a network is critical. Synchronization aids in data fusion [2, 3], estimation [4], and other designed tasks in cyber-physical systems. Additionally, synchronized data may create opportunities for designs and concepts that unsynchronized data cannot support.

Due to inaccurate clocks, clock drifts, delays in time stamping, and wireless transmission delays, data between sensors may be unsynchronized and misaligned. For example, a clock source with ±2000 parts per million (ppm) error would drift by approximately 120ms for every minute of operation. Furthermore, highly accurate clocks can still drift or be misaligned from other clocks in the system. Many current synchronization methods are based on communication between/among the sensors. Sensors may communicate with a central server to synchronize, but adding the necessary communication hardware and software can be expensive. If the sensors can communicate with the cloud or some other aggregation node, the communication delays and dropped data packets could add to the synchronization issues. In many cases, such sensor communication will not be possible.

In cyber-physical systems, sensors and actuators provide a crucial link between certain physical phenomena and the cyber-world (*i.e.*, computation and communication). As the number of sensors in the environment increases, the system is likely to be heterogeneous with a variety of clock generation methods, sampling rates, and communication protocols, which may be outside of the user's control. Therefore, one cannot expect to add hardware or software to each sensor in the system to support synchronization. Additionally, time stamping data upon reception by some central node or in the cloud ignores possible transmission delays and assumes immediate and accurate time stamping of received data. We require a novel technique that operates solely on the sensor data streams to synchronize the sensor data without modification to the sensor software or hardware. This synchronization tool could run in the cloud or at some aggregator node so that any sensor in the network can be synchronized without sacrificing the embedded nature and low power designs of the sensors.

In this paper, we present a method to align data from multiple sensors without any additional software or hardware on the sensors. Physical events measured by the sensors can provide a common, direct link between the data and the timing. This is the foundation of our proposed alignment and synchronization algorithm. This method takes data from a (possibly heterogeneous) sensor network, then uses measurements of physical events to find and align distinguishable critical points in the data sets to improve synchronization of the total data stream.

The remainder of this paper is organized as follows. Section 2 reviews related works on clock synchronization and alignment techniques. Section 3 provides background information on oscillators and sensor clock types. Section 4 explains our methods and algorithms for sensor synchronization and alignment. Section 5 covers the experiments and gives an analysis of the results obtained. Finally, Section 6 presents conclusions and future work.

2. RELATED WORKS

Techniques for sensor synchronization can be broadly divided into several categories: hardware based, communication based, and software based synchronization techniques. Some methods require combinations of these general techniques for the synchronization of sensor data.

Hardware techniques generally add components to the sensors to improve the accuracy and reduce drift of the clock. Clock and data

SWEC'15, April 13, 2015, Seattle, Washington, USA.

SWEC'15, April 13-16, 2015, Seattle, WA, USA

DOI: http://dx.doi.org/10.1145/2756755.2756763

recovery circuits synchronize the system by generating special pulsed patterns in the data that provide information for synchronizing clocks [5, 6]. These circuits require modifications for the sender (*i.e.*, sensor) and receiver (*i.e.*, host PC) of the data to handle the encoding and decoding, respectively, of the synchronization pulses in the data. Buevich et al. use a hardware clock-rate adjustment circuit in addition to a power-line sensing circuit to reduce clock drift and keep sensor clocks synchronized to a global clock [7].

Many synchronization methods for wireless sensor networks are based on two-way communication between the sensors or two-way communication with a host. Tabata et al. present a scheme to synchronize clocks in adjacent nodes in a mesh network [8]. The sensors exchange messages to determine master and slave nodes for synchronization purposes. Deemster et al. use a Kalman filter to estimate when the sensor and the host are not synchronized [9]. This information is used to trigger a "synchronization action" (*i.e.* time request) from the host to the sensor. Round trip time and the remote time are then used to synchronize the system. The sensor enters a special timing mode and does not make measurements when handling the synchronization messages.

Other synchronization techniques use software that is not on the sensor to synchronize the data. Olson used a software synchronization technique based on finding low-latency measurements to estimate the delay between a sensor measurement and the time stamp on the host PC for data fusion in a mobile robot [10]. Zaman and Illingworth present a method to restrict clock drift based on events measured by odometry and vision sensors on a mobile robot [11]. This method assumes the clocks are generally closely aligned and further inhibits deviation by detecting the moment when the robot starts to move in both data streams. Bækgaard et al. present a method for synchronizing EEG data with eye tracking data [12]. The blinking signature on each modality is used for synchronization by aligning blink probability functions determined for each modality through cross correlation.

The hardware and communication synchronization methods require the sensor to have special hardware or software to enable the synchronization, which incur extra cost and energy to the system and in some cases, disrupt the measurements. These methods also require complete control over every sensor in the system to keep sensor synchronization. The software methods synchronized the sensors without modifying the sensors. The last two methods discussed improve synchronization by measuring concurrent events on multiple sensors.

Our proposed method can improve synchronization based on known, predetermined events as well as unknown events. Given known events, we generate templates to detect events on one sensor and align them with related events detected on a second sensor. Alternately, if no events are known ahead of time, our method can determine which events from one sensor can likely be used to align with unknown events on a second sensor that measures the same physical phenomena. The proposed method also works with no sensor hardware or software modifications.

3. BACKGROUND

There are a variety of clocking circuits used in sensors. The accuracy of these clocks typically vary from ±20ppm for a crystal oscillator to ±5000ppm or higher for digitally controlled oscillators (DCOs), voltage controlled oscillators (VCOs), and relaxation oscillators. Oscillators with higher accuracy generally add cost and power to sensor designs. To reduce cost and power concerns, sensors may not always be designed with the most accurate clock sources. Even sensors with high accuracy clock sources and similar sample rates will see a drift over time. Due to drift and inaccuracies of the clocks as well as transmission delays, all sensors in the system can be considered as sensors with possible poor synchronization and some jitter in the sampling rate.

Figure 1 details an experiment to measure delays between multiple clock sources on a single custom sensor. The sensor has three clock sources that were used to timestamp the data, a crystal oscillator, a DCO, and a relaxation oscillator. Please refer to Section 5 for a more detailed description of the sensor. The data was collected from the sensor that started fully charged and ran until the battery was depleted. The graph shows the difference between timestamps for the crystal oscillator vs. the DCO and between crystal oscillator and the relaxation oscillator. With no synchronization, the difference in the measurements is greater than 10 minutes for the DCO and the relaxation oscillator vs. the crystal after 22.5 hours of data collection. This amount of error would make combining data from sensors with these different clocking methods nearly impossible without some form of synchronization.

We note that for this particular sensor, there are two approximately linear segments that occur during the time that the data is captured. Essentially, the delay of the DCO vs the crystal for this sensor changes over time, but it is linear for long periods. The change in slope (*i.e.*, inflection point) between the segments can be seen around sample 6.2 million. A similar change can be seen between the crystal and the relaxation oscillator around sample 12 million. These changes in clock accuracy are likely due to the decrease in the battery voltage of the sensor over time.

Given an event detected in one sensor data stream, these models of delay (*i.e.*, delay profiles) can be used to estimate a range on the data stream of another sensor using a different clock source where the related event can be located. In other words, understanding the clock types on the sensors used can influence the frequency and amount of synchronization (or data alignment) needed in the system, as well as reduce the amount of data necessary to search for critical data points. Furthermore, reducing the size of the search will ideally improve accuracy in matching and synchronization.

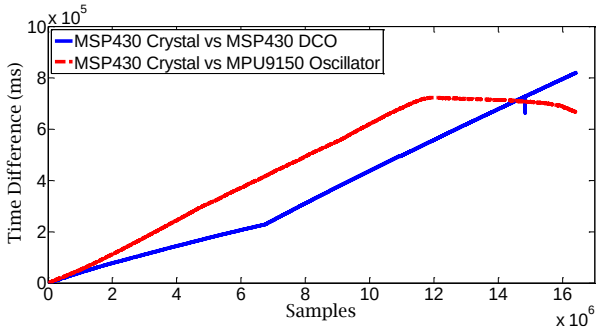

Figure 1: Clock source differences for over 22.5 hours.

4. ALIGNMENT AND SYNCHRONIZATION FORMULATION

Our proposed alignment method leverages events in the physical world to improve sensor synchronization. Specifically, we search the data streams for evidence of physical events that are monitored by multiple sensors. These events can then be used to align the sensor timestamps. We select one of the sensors in the system and treat this as our "world clock" (*i.e.*, the ideal sensor). Any sensor can be selected without loss of generality, but the most accurate sensor (if known) would generally be preferred.

4.1 Alignment Concept Formulation

Consider two sensors, $S1$ and $S2$, that are measuring a physical phenomenon where $S1$ is the ideal sensor. Sensors $S1$ and $S2$ collect sets of data points x_i and y_i, respectively, where $i \in \{1 \dots n\}$ is the time index and n is the current time. The sensors sample the data with sampling periods of $T_1 + \epsilon_{1i}$ for $S1$ and $T_2 + \epsilon_{2i}$ for $S2$ where T_k is the expected sampling period and ϵ_{ki} represents the jitter in the sampling period for $k \in \{1,2\}$. Each data point is time stamped by a clock source on the sensor generating the following data streams

$$\boldsymbol{X} = \{(x_i, t_i)\}, \quad \boldsymbol{Y} = \{(y_j, t_j')\}, \quad i,j \in \mathbb{N} \tag{1}$$

where t_i is the timestamp from the clock source on $S1$ and t_j' is the timestamp form the clock source on $S2$. With these concepts in place, we define alignment points.

Definition: An alignment point is a measurement (*e.g.*, x_i) of a physical phenomenon or event in a sensor data stream that can be accurately distinguished and directly related to a measurement (*e.g.*, y_j) of the same event in the data stream of another sensor.

For example, a sensor measuring the movement of a light switch and a light sensor would have measurements in their respective data streams related to the events of turning on and turning off the lights.

We denote the correct correspondence between the shared measurements of an event on two sensors as

$$x_i \equiv y_j \text{ where } i \neq j. \tag{2}$$

We determine the times related to the data points selected to generate a set of alignment points denoted as

$$\boldsymbol{A} = \{\{x_i, t_i\}, \{y_j, t_j'\}\}. \tag{3}$$

We select a subset of the alignment points, then use the timing data from one sensor clock source to synchronize the other sensor's clock. For a given alignment point from $\boldsymbol{A}$, we set $t_j' \leftarrow t_i$, intuitively, replacing the less accurate clock or time with the more accurate time. After all times in $\boldsymbol{A}$ are adjusted, the clock data, t' (*i.e.* $S2$ time), is linearly resampled between the alignment points to create a new time series, $\hat{t}$. The new time series is combined with the data points from $\boldsymbol{Y}$ to create a synchronized data stream

$$\boldsymbol{Y_s} = \{(y_j, \hat{t}_j)\}. \tag{4}$$

As shown in Figure 1 and discussed in Section 3, the delays of one clock oscillator vs. another are often approximately piecewise linear over long collection times. However, the clock jitter may make the delay nonlinear over shorter segments. Additionally, other sensors and clock sources likely will not exhibit the same behavior as the clocks on the custom sensor. Without greater knowledge of the delay profiles, linear resampling gives a valid estimation of the timing data between alignment points. Other resampling schemes could be used if more appropriate.

Understanding the delays over long data streams can assist in determining the optimal alignment points for processing large amounts of data and ensuring minimal processing time. If the relative drift between two clocks is known (at least approximately) and piecewise linear over a certain time window, the amount of processing can be reduced by selecting alignment points from $\boldsymbol{A}$ on either side of the inflection points when the slope of the linear fit changes. Because only two points are necessary to determine a line, distinguishable points closest to the inflection points from the delay profile would likely give the best possible alignment with minimal processing. If the drift is not constant or nonlinear, more alignment points can be used to improve the synchronization.

A limitation of this technique is the possible matching of the wrong data points in data stream 1 and data stream 2. Minimizing the search range based on knowledge of the possible delays helps to reduce this risk. Highly repetitive/periodic signals could still cause an issue, but single point errors do not propagate any synchronization error beyond the next alignment point. Essentially, the algorithm recovers from any errors at the next alignment point. For long-term collections, it is expected that there will be more time of inactivity, which will also mitigate the risk.

4.2 Template-Based Alignment Point Selection

If previous synchronized data sets for known physical events are available, we can use templates and pattern matching algorithms to extract alignment points from the data streams. Figure 2 shows an example template for two sensor data streams. The sensors represent the accelerometer data from sensors attached to the thigh of a subject (sensor 1) and a chair (sensor 2) when the subject does a stand-to-sit motion. This two-signal template represents the relationship between the two properly synchronized data streams.

We use dynamic time warping (DTW) [13] to find the locations in the sensor 1 data stream that match the sensor 1 template. The locations in data stream 1 that match the template provide a set of data points for the sensor 1 data stream that indicate a known event. For all data points found in the first data stream, the corresponding data points in the second sensor data stream must be found. We use a registration technique based on mutual information for image registration [14] to compare the templates and both sensor data streams to find these points.

Mutual information is defined as

$$I(X;Y) = \sum p(x,y) \log_2 \frac{p(x,y)}{p(x)p(y)} \tag{5}$$

where $p(x)$ and $p(y)$ are the probability distributions (estimated by histograms) of signal x and signal y, respectively, and $p(x,y)$ is the joint probability for signal x and signal y.

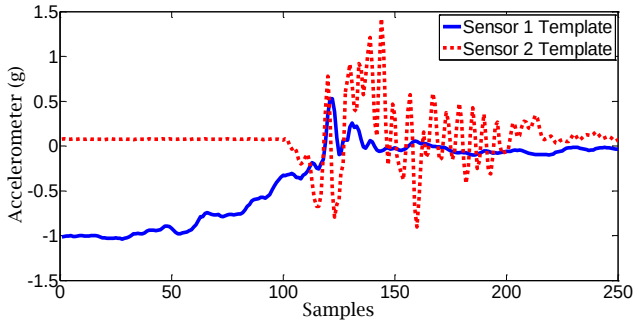

Figure 2: Sensor 1 (thigh) & Sensor 2 (chair) templates

The templates are treated like a 2-by-*N* image where *N* is the length of the templates. To determine the matching points on the sensor 2 data stream, a search range is selected based on the index of the data point from sensor 1. A sliding window centered around this point on data stream 2 is searched, as illustrated in Figure 3. The alignment point for the data streams is determined by the highest mutual information when comparing the template image against the 2-by-*N* images from data stream 1 and a sliding window on data stream 2.

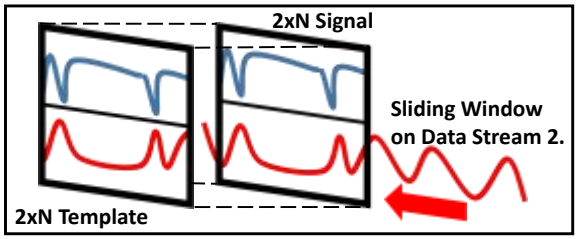

Figure 3: Matching of 2xN template vs 2xN signal

4.3 Entropy Based Alignment Point Selection

In many cases, we do not have the full knowledge or understanding of the cyber-physical system. Therefore, we are unable to make a set of templates with ideal timing. In this scenario, possible critical data points on data stream 1 can be found through statistical or information theoretical approaches.

Entropy is a measurement of the randomness of a signal [15] and is calculated as

$$H(x) = -\sum p(x)\log_2 p(x) \qquad (6)$$

where $p(x)$ is the probability of a given x in the signal's distribution based on a histogram. Using a sliding window across data stream 1, the entropy is calculated for each window. The highest entropy values give segments of the data stream with the most information (in Shannon's sense) or most "interesting" distributions. The entropy peaks are used to select data points in the first stream that will be used for matching.

Figure 4 shows a data stream and the related entropy calculation. The segments of the signal with the highest entropy match the most distinct features of the signal, which is expected. Because we do not know exactly what these data streams represent, we assume that the physical event that creates an "interesting" event on one sensor will also cause a similar event on the other sensors measuring the same phenomena. For example, if one sensor measure vibrations on a bridge while another measures the load on the bridge, a large truck driving over the bridge will cause a significant event in the data streams of both sensors and therefore corresponding high entropy measurements.

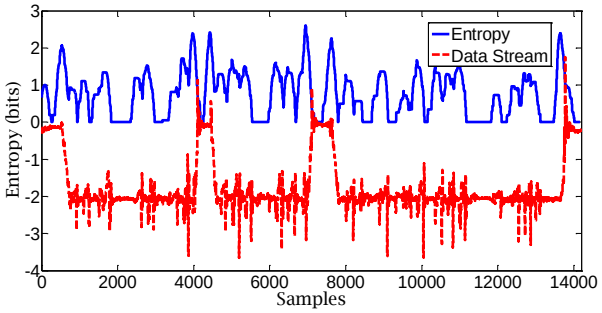

Figure 4: A data stream and the entropy calculated.

Once the data points on the first data stream are selected, the corresponding data points on the second data stream must be determined. This is again done through mutual information as described in (5). In this scenario, the mutual information is calculated on the first data stream segment against a sliding window on the second data stream within a range based on the index from data stream 1 and the delay profile. The peak of the mutual information calculation is selected as the corresponding data point for the second data stream. The two data points and times are combined to make the alignment point.

4.4 Alignment Point Subset Selection

Each alignment point determined by the template or entropy based methods could be used for synchronization of the signals, but this may be computationally expensive depending on the length of the signals and the number of alignment points available. We want to select a subset of alignment points (*i.e.* $\boldsymbol{S} \subseteq \boldsymbol{A}$) for synchronization. Consider that each alignment point $a_i \in \boldsymbol{A}$, has a position in the signal, p_i, and quality score, q_i, which indicates how well the alignment point can accomplish the alignment. Based on these characteristics, we define two methods for selecting $\boldsymbol{S}$.

4.4.1 Regional Peak Selection

The first method for determining $\boldsymbol{S}$ is called Regional Peak Selection. Based on the quality score and a region, R, we select the subset of alignment points. The highest quality score is selected, and other alignment points within the region around this peak are dominated (*i.e.* not selected). All a_i are ordered by decreasing q_i and the following $\boldsymbol{S}$ is determined by

$$a_i \in \mathbf{S} \; iff \; \max(q_i), \; |p_i - p_j| > R \text{ and } i > j. \qquad (7)$$

This method prioritizes the quality of the alignment point, but also works to spread the alignment points across the signal. Because of the emphasis on quality, there could be larger regions of the signal with no alignment points if the value of R is too high.

4.4.2 Binary Search Selection

The second method of selecting $\boldsymbol{S}$ is called Binary Search Selection. This method is based on bisecting the signal to select the positions for the alignment points. The number of bisections, which directly drive the number of alignment points, is selected by choosing a level L. The number of alignment points that this method will select is equal to $2^L - 1$ as each segment of the signal from the prior level is bisected at the current level. The positions based on the level, l_k, are used to determine $\boldsymbol{S}$ and is based on

$$a_i \in \mathbf{S} \; iff \; |p_i - l_k| < |p_j - l_k|, \forall \; l_k. \qquad (8)$$

The Binary Search Selection method prioritizes evenly spacing alignment points throughout the signal. It should be noted that some alignment points may be selected multiple times if they are closest to multiple level positions, l_k. If that occurs, duplicates will be removed

5. Experiments and Results

In the experiments, we use a custom sensor board with a Texas Instruments MSP430 microcontroller, and an InvenSense MPU9150 inertial measurement unit (IMU) collecting data at 200Hz. The sensor uses the MSP430 crystal oscillator (±20ppm), the MSP430 DCO (±5,000ppm), and the MPU9150 relaxation oscillator (±12,000ppm) to timestamp the data. Using MATLAB, the alignment algorithms are applied to the sensor data for one sensor using the crystal oscillator timestamp, and the second sensor using the MPU9150 oscillator timestamp.

5.1 Metrics

The alignment algorithms are evaluated using the following metrics

$$E_{nTot} = \frac{\sum_i^n |t_i - \hat{t}_i|}{n} \qquad (9)$$

$$ppm = \frac{\Delta t}{T} \times 1{,}000{,}000 \quad (10)$$

where E_{nTot} is a measure of how much the synchronized data differs for all data points through the data point *n*, and ppm is a standard measure of accuracy for clocks. E_{nTot} takes into account jitter between the synchronized data timing and the ideal data timing while ppm measures how much drift the sensor has at a specific point relative to how long the sensor has been running. The end points of the clocks are set equal when calculating E_{nTot} to normalize the metric between synchronized and unsynchronized data. We calculate the ppm using a sliding window of 6000 samples (*i.e.*, ~30s) for each point in the data stream. We use the crystal oscillator on the second sensor as our gold standard. The accuracy of the crystals ($\pm 20ppm$) ensures that the drift between the two sensor crystals will be small (~ 2.4ms max for 60 seconds). Two sets of experiments were designed to test the alignment and synchronization algorithms.

5.2 Template Based Experiments

The template based experiments were conducted with two sensors and known templates as discussed in section 4.2. In the first experiment, Sensor 1 is attached to the right thigh of a human subject; and Sensor 2 is rigidly attached to the arm of an office chair. The first template represents a stand-to-sit motion on the X-axis of the accelerometer on the thigh sensor while the second template represents the vertical displacement of the chair measured by the Z-axis of the accelerometer in the chair sensor.

In the second template-based experiment, Sensor 1 is a wrist-worn sensor and Sensor 2 is mounted on an office door. The first template is the X-axis gyroscope on the wrist worn sensor and is used to detect when the door knob is turned. The second template from the sensor on the door uses the X-axis of the gyroscope to detect when the door is opened or closed.

The final template based experiment uses the first sensor on a record-player turntable and the second sensor on the tone arm of the turntable. The Sensor 1 template is the Z-axis gyroscope, which measures the rotation of the turntable. The second template is the X-axis gyroscope of Sensor 2 which represents the arm lifting.

5.3 Non-template Based Experiments

There were two non-template (*i.e.*, entropy) based experiments. In the first set of experiments, two IMU sensors were rigidly attached to each other (*i.e.*, stacked). In the second experiment, the two sensors were placed on the two arms of a chair. In these experiments, we do not have a template, rather we expect that events causing measurable signals on one sensor will also cause detectable signals on the other sensor. As explained in Section 4.3, entropy is used to select the data points in the first signal, that are then used to find the corresponding data points the second data stream. For these experiments, the same axis and modality (*i.e.*, accelerometer axis) of each sensor was used for signal processing. The window size for a segment was set to 200 samples (*i.e.* ~1s).

5.4 Results and Analysis

Table 1 shows the synchronization metrics for the non-template based and the template based experiments. We compare the subset selection algorithms against each other as well as the original data for the non-template based experiments. The Binary Search method was run with values of *L* from two to five showing an average improvement of ~12,000ppm to ~196ppm and ~359ppm for the two scenarios. The Region Peak method was run with values of *R* that generated the same number of alignment points as the Binary Search method for fair comparison. The Region peak method showed an average improvement from ~12,000ppm to ~209ppm and ~680ppm for the two scenarios. We used all alignment points for the template-based experiments because of the smaller number of alignment points (*i.e.*, 4 or less). The average improvement for the three template based experiments were from ~11,000ppm to ~5,000ppm, ~1800ppm, and ~200ppm.

Generally, the template-based solution has larger errors than the non-template-based solution except for the last template based experiment. The primary cause of this additional error is the human based movements in the first and second template based experiments. The subject may do the movement (*i.e.*, stand-to-sit and open door) at speeds that vary from the templates. Additionally, the two matching algorithms used for each set of alignment points increases the chances for error in matching and therefore synchronization. The first cause of error is mitigated in the turntable experiment because the machine performs the actions (*i.e.* turning the turn table and lifting the arm) at the same speed each time, which leads to more accurate synchronization.

Figure 5 shows the error between the input clock and the gold standard clock as well as the error between the input clock and the synchronized clock based on the Binary Search method with $L = 2$. The beginning and end of the clock signals are made equal when generating this figure. Please note that the figure is illustrating the impact of the synchronization as the error between the synchronized clock and the gold standard clock (the area between red and blue lines) is lowering significantly. The error between the gold standard clock and the synchronized clock (after the alignment) corresponds to ~35ppm with an $E_{nTot} = 1.2$ms.

Figure 6 shows raw sensor data before (red dotted line) and after the alignment (green solid line) and synchronization algorithm for a stand-to-sit versus the chair. At this alignment point, the algorithm adjusts the original sensor clock by 279ms. Due to the factors mentioned for template based alignment, this is not a perfect alignment. There is still an error of about 38ms when compared to the ideal clock source.

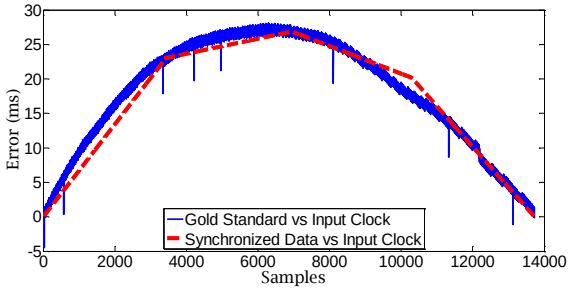

Figure 5: Input clock error vs gold standard and Binary Search Method ($L = 2$) for a stacked sensor experiment.

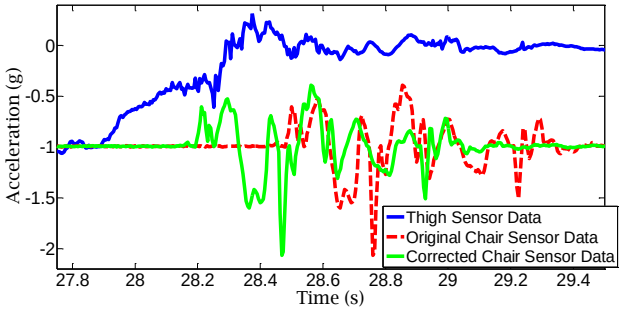

Figure 6: Comparison of chair vs thigh sensor alignment

Table 1: Metrics for Original and Synchronized Data

Template Based Method					
Experiment		**Thigh, Chair**	**Wrist, Door**	**Turn-table**	**Original**
E_{nTot} (ms)		72.8	51.6	22.9	13.3
ppm		5,131.1	1,862.2	199.9	11,907
Non-Template Binary Search Method					
Experiment		**Stacked**	**Two on Chair**	**Original**	
E_{nTot} (ms)	min	1.2	6.0	10.9	
	max	16.4	24.9	216.1	
	avg	5.9	11.4	70.2	
ppm	min	35.3	291.8	11,443	
	max	423.5	513.0	14,259	
	avg	195.8	359.2	12,224	
Non-Template Region Peak Method					
Experiment		**Stacked**	**Two on Chair**	**Original**	
E_{nTot} (ms)	min	2.9	10.4	10.9	
	max	27.4	33.2	216.1	
	avg	7.4	27.3	70.2	
ppm	min	107	215.7	11,443	
	max	529.1	838.9	14,259	
	avg	208.5	680.7	12,224	

Longer data captures amplify the effects of drift on the data, and therefore the alignment points will have a greater positive effect on the quality of synchronization. The Binary Search method shows the best accuracy for the non-template based synchronization. The average accuracy was ~277ppm for the two scenarios with average $E_{nTot} = 8.6$ms, which is less than two samples of error for a 200 Hz clock. The average E_{nTot} for template-based method is 49ms which is worse than the original E_{nTot} due to the high nonlinearity in the unsynchronized clocks. Using a larger number of alignment points would improve this metric. On the other hand these experiments had an average accuracy of ~2398ppm which is much better than the unsynchronized clock. Most of this error is due to the imprecise human based templates. For the experiment with a well-defined mechanical template, the accuracy is ~200ppm, which is in line with the best non-template based methods.

6. CONCLUSIONS AND FUTURE WORK

We presented an alignment and synchronization algorithm for cyber-physical systems that requires no modifications to the sensor software or the sensor hardware. The algorithm synchronizes sensor data timing based solely on the data streams of the sensors and a clock from one of the sensors. The algorithm can also determine the points in the data that are likely to give the best data points for alignment when no templates are available.

We plan to continue this work by improving estimating the delays of the system during processing to more efficiently align and select alignment points. Additionally, we will look at the interaction in synchronizing multiple sensors in a network and how to determine the optimal sensors for alignment.

7. ACKNOWLEDGEMENTS

This work was supported in part by the National Science Foundation, under grants CNS-1150079 and CNS-1012975, and the TerraSwarm, one of six centers of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding organizations.

8. REFERENCES

[1] E. Lee, J. Rabaey, B. Hartmann, J. Kubiatowicz, K. Pister, A. Sangiovanni-Vincentelli, S. Seshia, J. Wawrzynek, D. Wessel, T. Rosing, D. Blaauw, P. Dutta, K. Fu, C. Guestrin, B. Taskar, R. Jafari, D. Jones, V. Kumar, R. Mangharam, G. Pappas, R. Murray, and A. Rowe, "The swarm at the edge of the cloud," *Design Test, IEEE*, vol. 31, pp. 8–20, June 2014.

[2] J. Shen, D. Tick, and N. Gans, "Localization through fusion of discrete and continuous epipolar geometry with wheel and imu odometry," in *American Control Conference (ACC), 2011*, pp. 1292–1298, IEEE, 2011.

[3] Q. Yang and J. Sun, "An underwater autonomous robot based on multi-sensor data fusion," in *Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on*, vol. 2, pp. 9139–9143, IEEE, 2006.

[4] T. Bennett, R. Jafari, and N. Gans, "An extended kalman filter to estimate human gait parameters and walking distance," in *American Control Conference (ACC), 2013*, pp. 752–757, IEEE, 2013.

[5] H. Yu, Y. Li, L. Jiang, and Z. Ji, "Ultra-low-power adaptable ask clock and data recovery circuit for wireless implantable systems," *Electronics Letters*, vol. 49, no. 10, 2013.

[6] J.-W. Kwon, X. Jin, G.-C. Hwang, J.-H. Chun, and K.-W. Kwon, "A 3.0 gb/s clock data recovery circuits based on digital dll for clock-embedded display interface," in *ESSCIRC (ESSCIRC), 2012 Proceedings of the*, pp. 454–457, IEEE, 2012.

[7] M. Buevich, N. Rajagopal, and A. Rowe, "Hardware assisted clock synchronization for real-time sensor networks," in *Real-Time Systems Symposium (RTSS), 2013 IEEE 34th*, pp. 268–277, IEEE, 2013.

[8] K. Tabata, Y. Kishi, S. Konishi, and S. Nomoto, "A study on the autonomous network synchronization scheme for mesh wireless network," in *Personal, Indoor and Mobile Radio Communications, 2003. PIMRC 2003. 14th IEEE Proceedings on*, vol. 1, pp. 829–833, IEEE, 2003.

[9] E. Demeester, J. Philips, A. Huntemann, E. Vander Poorten, and H. Van Brussel, "Active, lifelong sensor synchronization: a kalman filtering approach," in *Advanced Robotics (ICAR), 2011 15th International Conference on*, pp. 501–506, IEEE, 2011.

[10] E. Olson, "A passive solution to the sensor synchronization problem," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pp. 1059–1064, IEEE, 2010.

[11] M. Zaman and J. Illingworth, "Interval-based time synchronisation of sensor data in a mobile robot," in *Intelligent Sensors, Sensor Networks and Information Processing Conference, 2004. Proceedings of the 2004*, pp. 463–468, IEEE, 2004.

[12] P. Bækgaard, M. K. Petersen, and J. E. Larsen, "In the twinkling of an eye: synchronization of eeg and eye tracking based on blink signatures," in *4th International Workshop on Cognitive Information Processing (CIP 2014)*.

[13] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series.," in *KDD workshop*, vol. 10, pp. 359–370, Seattle, WA, 1994.

[14] J. P. Pluim, J. A. Maintz, and M. A. Viergever, "Mutual-information-based registration of medical images: a survey," *Medical Imaging, IEEE Transactions on*, vol. 22, no. 8, pp. 986–1004, 2003.

[15] T. M. Cover and J. A. Thomas, "Entropy, relative entropy and mutual information," *Elements of Information Theory*, pp. 12–49