Smart Fence: Decentralized Sequential Hypothesis Testing for Perimeter Security

Fabien Chraim and Kristofer Pister

University of Califronia, Berkeley CA 94720, USA,
Electrical Engineering and Computer Sciences
{chraim,pister}@eecs.berkeley.edu

Abstract. This paper presents a practical solution for the fence-line intrusion detection problem. The MEMS based sensor platform is introduced along with its WirelessHART networking capabilities. A Decentralized Sequential Hypothesis Testing algorithm is studied at the theoretical level before being applied to the problem at hand. Results from several deployments show 100% detection rates within one second of the intrusion. The system is presented as a viable replacement for much more expensive and less performant security systems.

Keywords: Smart Fence, WSN, MEMS, WirelessHART, Hypothesis Testing.

1 Introduction

Since humans first transitioned from nomadic tribes to permanent settlements, we have had to find solutions to the essential problems associated with property: how to mark what is ours and how to protect it. In short, perimeter security has been an ever-evolving priority in human history. At their most basic, fences delineate land and serve as a marker of private property. Factor in a few technological innovations, and they can serve as a deterrent to intruders, or even an alarm system. However, even the strongest barriers have their weaknesses. Fences can be climbed over, dug under, and even cut through.

Many security systems were implemented in an attempt to reinforce the fence structure [5]. The most common fence line intrusion detection systems are taut wire, fiber optic, strain-sensitive cable, electric field and capacitive systems. However, each of those approaches comes with its own set of challenges, rendering it undesirable. For example, taut wire setups are expensive, complicated to install, and require regular tensioning of the fence sections. Fiber optic fence sensors, which use light variations within the communication medium to detect movements, are suceptible to weather turbulences. False alarms can also result from poor fence installations. This is also the case of strain-sensitive cables, which can also suffer from Electro-Magnetic interference. This type on interference does not affect electric field and capacitive installations. However, those systems suffer from suceptibility to weather and animal movement, as well as the

M. Zuniga and G. Dini (Eds): S-Cube 2013, LNICST 122, pp. 65–78, 2013.

gradual growth of vegetation along the fence line. All of the described security schemes require laying at least one cable around the facility to be monitored. This becomes quite expensive as the perimeter grows, and also leads to issues with maintenance.

Our approach to the problem of fence intrusions in perimeter security will be one of Wireless Sensor Networks (WSN) combined with cheap Microelectromechanical systems (MEMS) sensors. Essentially, we will set up a low-power wireless network along a fence line, equip each fence section with vibration and inertial sensors and run detection algorithms to detect intrusions. The goal is to detect all manner of intrusions (e.g. kicking, climbing, leaning, rattling) with zero false negatives, while limiting false positives to less than 1/mile/month. This idea is not novel as can be seen in [6], [7] and [8]. The difference between this work and that undertaken by Yousefi et al. is our more advanced hardware platform and greater emphasis on the network architecture, the system, and its scalability. Furthermore, our solution is a hybrid between distributed and centralized detection. Finally, we try to move away from the unecessary complexity that often characterizes of academic studies. The simplicity of our algorithm allows us to reach superior performance with, we conjecture, a much lower energy consumption. Concerning the study undertaken by Wittenburg et al., the authors do not present any reason for choosing only accelerometers to perform their sensing, and fail to explain the less-than-intuitive placement of their module on the fence poles. The results presented in their paper are not satisfactory, as they fail to bring down the rate of false negatives, something that is of great importance for security systems. Finally, though one can appreciate the exercise of training and classification, it is apparent in [8] that such an approach is of no use in fence-line inertial sensing. The main reason of course is that signals captured at the fence are chaotic in nature and vary widely from one climber to the other, making the extraction of common features a very difficult task. The nature of this problem however hints at a different statistical tool we employed in our detection scheme: sequential decision problems.

In this paper, we start by studying the theoretical problem of sequential hypothesis testing in section 2, both for the centralized and decentralized case. We then describe the experimental setup both in terms or hardware and software and network architecture in section 3. Applying the theory to a practical algorithm is shown in section 4. Section 5 illustrates the results we obtained following several deployments, before concluding in section 6.

2 Sequential Hypothesis Testing

Sequential analysis is the branch of statistics that deals with decision making as the samples are being collected. It differs from classical hypothesis testing in that conclusions are reached more quickly, often before the end of the experiment. Consequently, it is ideal for detection, signal processing, clinical trials and other applications. In general, sequential decision problems involve one or more sensors and a fusion center where the final decision is made. In the centralized setting, all

of the information received by the sensors is made available at the fusion center. However, in the decentralized case, the sensors themselves are part of the decision process and relay partial information rather than all of their observations [1,3,4]. The following two sub-sections go into some detail concerning both approaches. In this part of the paper, we will be following the study done in [2].

Let us now formally define the problem we want to solve.
Depicted in figure 1, we see K sensors observing a stochastic process $\{\xi_t^i\}$ with prior P^i. The assumption we make here is based on the study in [2], and it states that the observed processes are independent. Furthermore, we consider that there are two possible hypotheses in the system at hand

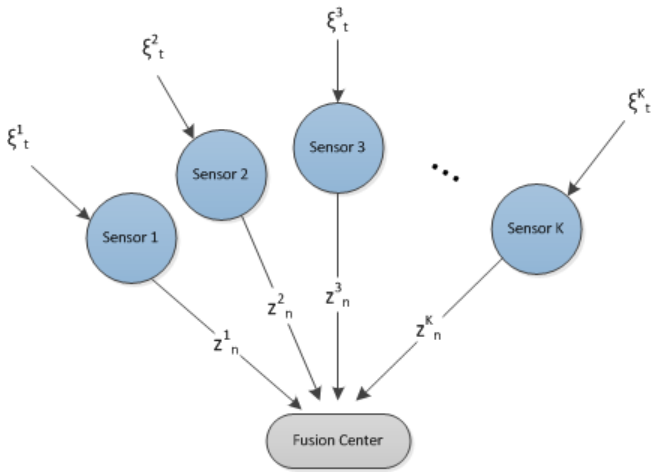

Fig. 1. A system of K sensors and a fusion center

$$H_0 \text{ with probability } P = P_0 \text{ and } H1 \text{ with } P = P_1 \tag{1}$$

Additionally, $P_j = \prod_{i=1}^{K} P_j^i$ for $j = 0, 1$. The next step would be to define the local log-likelihood ratio that takes place at each of the sensors.

$$u_t^i = log\frac{dP_1^i}{dP_0^i}(\xi_t^i) \text{ with } u_0^i = 0 \tag{2}$$

Independence between the observed processes allows us to write the following log-likelihood ratio that takes place at the fusion center,

$$u_t = log\frac{dP_1}{dP_0}(\xi_t^i) = \sum_{i=1}^{K} u_t^i \text{ for } 0 \leq t < \infty \tag{3}$$

The aim is to optimally define the pair (T, d_T) where T is the stopping time and d_T is the decision, which takes values 0 or 1, depending on which hypothesis was picked. We shall attempt this both in the centralized and decentralized cases.

2.1 Centralized Sequential Hypothesis Testing

It has been shown by Wald and Wolfowitz [1] that the Sequential Probability Rratio Test (SPRT) is optimal in solving the centralized case of the following problem.

Given, the type-I and type-II probability levels $\alpha, \beta > 0$ such that $\alpha + \beta < 1$, we want to find $(\mathcal{T}, d_T)$ such that

$$E_j[\mathcal{T}] = inf E_j[T], j = 0, 1 \tag{4}$$

The SPRT mentioned above is defined in this case as,

$$\mathcal{T} = inf\ t > 0 : u_t \notin (-A, B) \tag{5}$$

$$d_{\mathcal{T}} = \begin{cases} 1 \text{ if } u_{\mathcal{T}} \geq B \\ 0 \text{ if } u_{\mathcal{T}} \leq -A \end{cases} \tag{6}$$

$$A = log(\frac{1-\alpha}{\beta}), B = log(\frac{1-\beta}{\alpha}) \tag{7}$$

Hence, the procedure for applying this test is to obtain the observations from the sensors, apply the global log-likelihood ratio, add it to the previous value and verify whether this sum left the open interval $(-A, B)$ or not. The first time at which we leave this interval will be time $\mathcal{T}$, and the associated decision $d_{\mathcal{T}}$ will follow the rule in (6). This test is optimal under our assumptions. We will see in the next sub-section how to generalize this concept to the decentralized case.

2.2 Decentralized Sequential Hypothesis Testing

This decentralized problem is approached from the discrete time case. Though the continuous time case is studied in [2], it is not of major practical use. Rather it provides some intuition on treating its discrete time equivalent. As we have mentioned before, the difference between the centralized and decentralized cases is that, in the latter, the sensors make decisions before relaying some information back to the fusion center. Indeed, each sensor computes a local log-likelihood ratio,

$$l_n^i = log \frac{dF_1^i}{dF_0^i}(\xi_n^i) \tag{8}$$

Now, we set two thresholds $-\underline{\Delta}_i, \bar{\Delta}_i$,

$$-\underline{\Delta}_i = log \frac{P_1}{P_0}(\xi_n^i = 0), \bar{\Delta}_i = log \frac{P_1}{P_0}(\xi_n^i = 1) \tag{9}$$

This allows us to define an SPRT that occurs at each sensor as follows,

$$z_n^i = \begin{cases} 1 \text{ if } u^i_{\tau_n^i} \geq \bar{\Delta}_i \\ 0 \text{ if } u^i_{\tau_n^i} \leq -\underline{\Delta}_i \end{cases} \tag{10}$$

where τ_n^i is the local stopping time at the sensor and z_n^i is the information sent to the fusion center at that time. In order for the fusion center to compute its log-likelihood ratio based on the z_n^i, we could envision that the following two values be precomputed and made available at that fusion center:

$$-\underline{\Lambda}_i = log\frac{P_1}{P_0}(z_n^i = 0), \bar{\Lambda}_i = log\frac{P_1}{P_0}(z_n^i = 1) \tag{11}$$

Now, the reason the pair $\underline{\Lambda}_i, \bar{\Lambda}_i$ is defined separately from the pair $\underline{\Delta}_i, \bar{\Delta}_i$ resides in the fact that discrete time sampling gives rise to an *overshoot effect*. One can think of this effect as an uncertainty in the exact time the local log-likelihood ratio crossed the open interval $(-\underline{\Delta}_i, \bar{\Delta}_i)$ for the first time.

In turn, the fusion center will use the information provided by the sensors to update its global log-likelihood ratio.

$$u_n = u_{n-1} + \begin{cases} \bar{\Lambda}_i & \text{if } z_n^i = 1 \\ -\underline{\Lambda}_i & \text{if } z_n^i = 0 \end{cases} \tag{12}$$

The main result of Fellouris and Moustakides in [2] is first to define a measure of the D-SPRT thresholds at the fusion center,

$$\tilde{A} \leq |log\beta|, \tilde{B} \leq |log\alpha| \tag{13}$$

then to show the following asymptotic optimality on the global log-likelihood ratio:

$$|E_0[u_{\tilde{\mathcal{T}}}] - E_0[u_{\mathcal{T}}]| \leq \frac{O(\theta)}{\Theta(\Delta)}|log\beta| + \Theta(\Delta) \tag{14}$$

$$|E_1[u_{\tilde{\mathcal{T}}}] - E_1[u_{\mathcal{T}}]| \leq \frac{O(\theta)}{\Theta(\Delta)}|log\beta| + \Theta(\Delta) \tag{15}$$

where θ is the maximum overshoot (formally defined in [2]), Δ is equivalent to any of the two $\underline{\Delta}_i, \bar{\Delta}_i$ assumed to be equal, and $\tilde{\mathcal{T}}$ is the stopping time for D-SPRT at the fusion center, while $\mathcal{T}$ is the optimal stopping time for the centralized case. In fact, one can readily see that because of the loss of information and loss in time resolution between the centralized and decentralized cases, the optimality is lost. Equations (13) and (14) represent the Kullback-Leibler divergence (which can be thought of as the relative entropy between an optimal distribution and a sub-optimal one) applied to the log-likelihood ratio. The authors in [2] go on to simulate and observe that the D-SPRT derived in this section is useful in most practical implementations. The reader is invited to study fig. 3 of [2]. The proof of the above result uses simple concepts in statistics and is within grasp of readers with some background in stochastic processes.

3 Experimental Setup

Now that the theory is well understood, we turn our focus to implementing a solution to the perimeter security problem at hand. Tackling fence monitoring

was not a straightforward task. In fact, fence models were not readily available, which meant that building any detection infrastructure needed thorough experimentation. We now present the hardware platform of choice along with the underlying software architecture.

3.1 Hardware

The hardware platform for this project is based on the GINA (general inertial and navigation assistant) [9] which is developed at UC Berkeley and is used in many research groups around the world. This sensor board comes with the MSP430f2618 microcontroller from Texas Instruments, and sensors listed in table 1. GINA is expandable with analog sensors and digital devices through its expansion pins. As can be seen in figure 2, a daughter card is mounted on those pins and sits on top of the GINA board. This daughter card is referred to as the WirelessHART Interface Module (WHIM) and carries the DN2510 radio by Dust Networks. As can be inferred by the board name, this radio is a WirelessHART compliant one at 2.4GHz. The GINA/WHIM combination consumes a few microwatts in sleep mode and, when it is running at full capacity, currents around 20mA were recorded at 3V. An IP-65 enclosure (shown in figure 3) completed the hardware solution fitting nicely in the diamond pattern of the chain-link fence.

Table 1. List of Gina components and their features

Type	**Manufacturer**	**Part Number**	**Features**
Microcontroller	Texas Instruments	MSP430F2618	16-bit, 16MHz, 116kB flash, 8kB RAM
3-axis accelerometer (sensitive)	STMicroelectronics	LIS344ALHTR	+/-2 Gs or +/-6 Gs, 1.8 kHz, 660 mV/G
3-axis accelerometer (large range)	Kionix	KXSD9-1026	+/-8 Gs, 2 kHz
3-axis gyroscope	Invensense	ITG3200	2000 degs/s
3-axis magnetometer	Honeywell	HMC5843	+/- 6 Oe, 116Hz
Temperature sensor	Texas Instruments	TMP20AIDRLT	+/-2.5 C, -55 C to 130 C

3.2 System Architecture

The fact that the hardware platform holds a WirelessHART compliant radio, alludes to the system architecture. As the sensors are placed along the fence line, they join the WirelessHART network formed by the gateway, and start reporting data there. Looking at figure 4, we can see that a computation element exists both at the sensor side and at the gateway as well. This means that any detection scheme can implement one of the following three models:

Fig. 2. The hardware platform: GINA/WHIM

Fig. 3. The IP-65 enclosure that houses the hardware platform and sits on the fence [courtesy of Hammond Inc]

– server-side computations only
– sensor-side computations only
– hybrid model with computation taking place at both ends

The first model is clearly not scalable, as all the nodes would be required to send all of their raw sensor data, thereby flooding the network. In the second case, an energy issue arises. Since the sensors are battery-operated, we need to be careful with resource usage. As such, the ideal solution would require some initial computation to be done on the sensors themselves, which then relay partial information to the gateway. The final decision is then generated at the server end, and an appropriate action performed.

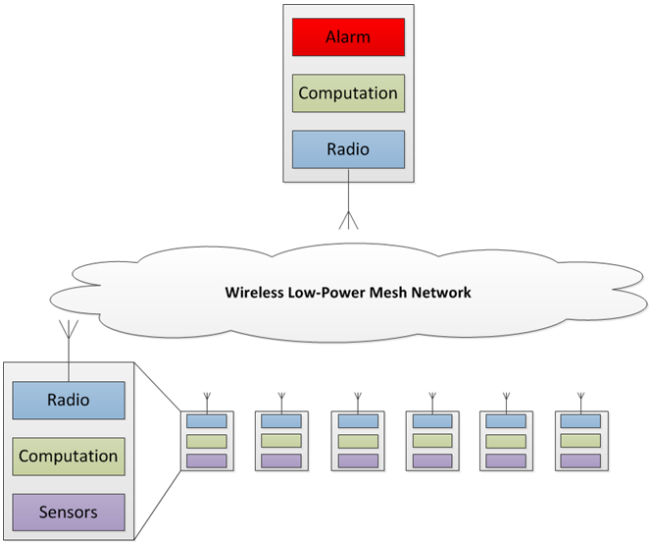

Fig. 4. System architecture for the Smart Fence. The detection algorithm can run only on the motes, or only on the server, or using a hybrid approach where some computations happen in the motes and their output is relayed to the server to make the final decision.

4 Detection Algorithm

In this section we explore the procedure we followed to develop the detection algorithm for fence monitoring.

4.1 Preliminary Testing

Some of the primary questions to answer are the following: Which of the sensor data generated by GINA is relevant? How fast should we sample our sensors? To

come up with a solution, we strapped three of the platforms we developped on three contiguous fence sections. All of the sensors were then sampled at 300Hz and the raw data transmitted to a gateway, as we ran controlled shaking, kicking and climbing tests. A singular value decomposition of the highly dimensional sensor data enabled us to single out three axes of interest, as can be seen in figure 5. Not surprisingly, it turned out that most of the information is contained in the z-axis of the accelerometer (pointing out of the plane of the fence), and the x-axis and y-axis of the gyroscope (in the plane of the fence). The result is intuitive, and the reader is invited to imagine a chain-link fence vibrating under the influence of, for example, shaking. Clearly, the accelerometer will see accelerations dominantly in the z-direction, while the angular rates observed by the gyroscope will mainly be along the plane of the fence itself. The next step in the analysis was to determine the sampling frequency. A quick look in the Fourier domain revealed that most events of interest happened below 35Hz, justifying our sampling frequency of 70Hz. An additional tradeoff can be observed here. Increasing the sampling frequency obviously yields better results, in terms of capturing all of the information contained in the the waveform. However, the added data generated by this oversampling has to be processed either at the sensor or server side. This means that energy will be spent either on the onboard microcontroller or during transmission.
The preliminary testing phase also showed that strong gusts of wind were problematic at the fence line. As a matter of fact, the accelerations recorded during those events were comparable to those recorded after a person gently shook the fence. For this reason, we made the decision to apply Lebesgue sampling and set a threshold under which, the entire platform enters sleep mode.

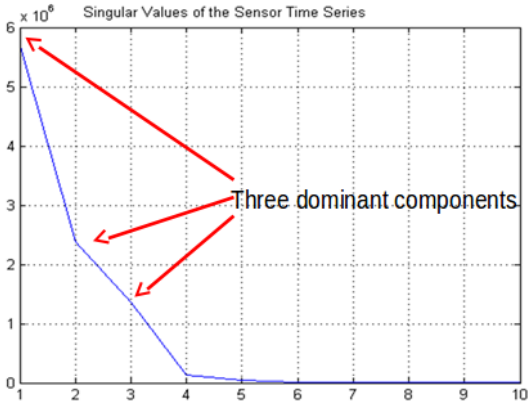

Fig. 5. Singular value decomposition of the highly dimensional sensor data. This plot shows that only three axes contain the majority of the data. Namely, the z-axis of the accelerometer into the fence and the x-axis and y-axis of the gyroscope in the plane of the fence.

4.2 Applying D-SPRT to Fence Monitoring

The process of applying D-SPRT in our case consists of selecting the probability priors both at the sensor side and server side. This was done by running controlled tests along the fence line. Similar to the previous section, intrusion traces were compared with "background noise" traces. With enough repetitions, the probabilities could be calculated. Those statistical values were then augmented with assumptions concerning the number of intrusions per time period. It is of note that this process does not need repetition and the thresholds could be applied to any fence section regardless of the environment or construction. Figure 6 shows actual log-likelihood ratios recorded at the sensors during experiments we performed on fences. As can be seen in the right-most figure, everytime the upper limit of the interval is crossed, a value of H1 is transmitted to the server and the ratio gets reset. The middle graph shows the case where the sensor records activity but deems it regular noise. A value of H0 gets transmitted and the ratio is reset as well. The left-most figure shows the case where the sensor starts recording data, but the traces end before the interval is left and no information is transmitted to the server. This corresponds to very small disturbances for example. Figure 7 shows the progression of the log-likelihood ratio over time for a climb signal and for a kick stimulus. It can be readily seen that the climb signal generates several H1 packets sent to the server side, while the kick signal only generates a couple before the ratio decays and the sensor stops recording. At the server side, a decision is made based on the number of H1 and H0 packets received, the reception time, and the relative proximity of the reporting sensors. With this last piece, the system was ready for deployments.

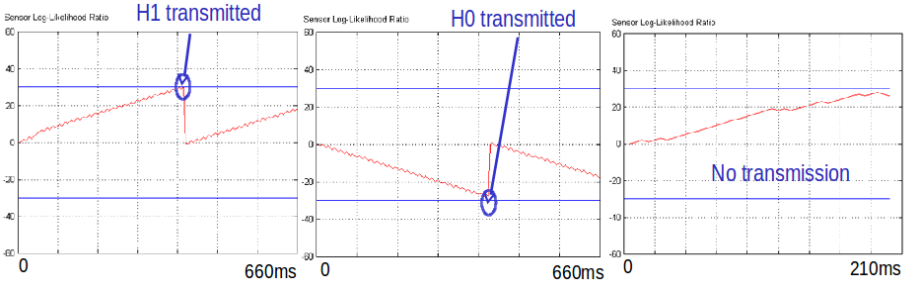

Fig. 6. Sensor log-likelihood ratios for various disturbances, in the short run. When the ratio crosses the upper limit, an H1 is transmitted to the server. This is then added to the global log-likelihood ratio that can trigger the alarm or ignore the sensor output.

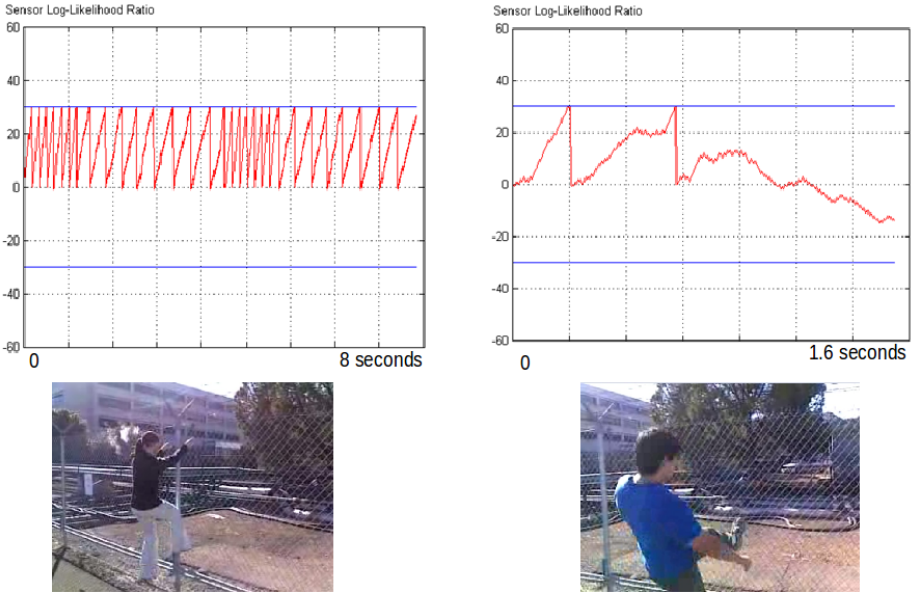

Fig. 7. Sensor log-likelihood ratios for a climb and kick signal, in the long run. In reaction to climbing the fence, the sensor reports a series of H1 consecutively as it seeing a lot of activity. For the kick stimulus however, the sensor may report a couple of H1 packets for example, before its log-likelihood ratio decreases.

5 Deployments and Test Results

Several deployments were made to put the Smart Fence system to the test. They varied in longevity, environmental conditions (e.g. wind, rain, time of the day) and fence construction. In all of them, different individuals were asked to climb the fence. Shown in figure 8 are the server side log-likelihood ratios for various climbers under varying conditions. Clearly we can see that the output of our dection algorithm is not the same in all of the time series. The main reason is, as stated before, not all climbing styles are equivalent. While some people tend to be aggressive, and generate a lot of activity at the fence line, others choose a calmer approach and climb more methodically. Independently of these variations however, our algorithm was able to detect intrusions every time we attempted one. The detection time was obviously not the same for all of them, but even with an interval of 660ms, we were able to see the algorithm reacting to the intrusion.

One long term deployment was also performed at the Chevron refinery in Richmond, California. During that test, four sensors were deployed along the North-East fence line of the Technology Center for a period of six weeks. Shown in figure 9 is the network manager and linux box running the detection algorithm overlooking the sensors from one of the offices in that building. Figure 10 is an aerial photograph of the deployment site. Some refinery employees were asked to disturb the fence line during the deployment period and record the time and date of that activity. At the end of the experiment, all of the "intrusions" were detected with no false alarms. Additionally, the sensors withstood rain, strong winds and direct sunlight.

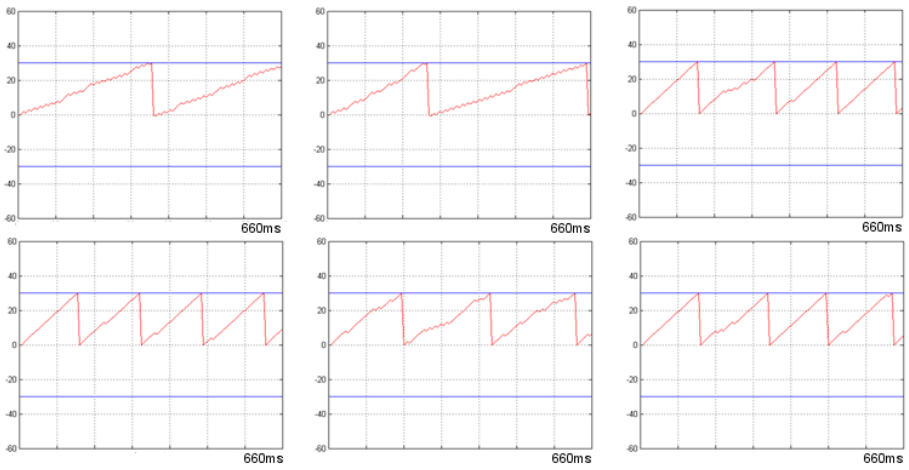

Fig. 8. Log-likelihood ratios for different climbers and conditions. This figure shows that even with different climbing styles, our detection algorithm reported the intrusion within a few milliseconds of its start.

Fig. 9. Long-term deployment setup at the Chevron-Richmond refinery. The result of this test was a detection rate of 100% with no false alarms. The sensors withstood strong winds and rainy weather.

Fig. 10. Aerial view of the deployment site [courtesy of Google Maps]

6 Conclusion

In this paper, we have demonstrated that the Smart Fence is a viable solution to the perimeter security problem of chain-link fences. Following our long-term deployment we observed 100% detection with no nuisances. Still, additional testing is needed in order to validate the detection algorithm and look for weak points. More intrusion methods should also be attempted to see if special cases need to be accounted for. One issue we did not address is the cost of the system. Since our custom-made hardware platform was built with commercial off-the-shelf parts, we can predict that, if produced in large quantities, the Smart Fence should directly compete with the solutions already available on the market. This still assumes that one sensor platform is needed per fence section. Nevertheless, one would certainly benefit from fusing different intrusion schemes along with our proposed solution. Perhaps a good practice would involve a human in the loop who, alerted by the Smart Fence system, would direct a CCTV camera to the corresponding fence section to check for the type of intrusion and its significance. In terms of future directions for this research effort, we observe an opportunity to make this security scheme even more robust by augmenting it with learning techniques to dynamically adjust the sensor thresholds based on collected statistical data.

Acknowledgements. We would like to thank Chevron ETC and Chevron ITC for funding and supporting this project.

References

1. Wald, A., Wolfowitz, J.: Optimum character of the sequential probability ratio test. Ann. Math. Statist. 19, 326–339 (1948)
2. Fellouris, G., Moustakides, G.: Decentralized Sequential Hypothesis Testing Using Asynchronous Communication. IEEE Transactions on Information Theory 57(1), 534–548 (2011), doi:10.1109/TIT.2010.2090249
3. Veeravalli, V.: Sequential decision fusion: theory and applications. In: Proceedings of the Workshop on Foundations Information/Decision Fusion with Engineering Applications (August 1996)
4. Tsitsiklis, J.: On threshold rules in decentralized detection. In: 1986 25th IEEE Conference on Decision and Control, vol. 25, pp. 232–236 (December 1986), doi:10.1109/CDC.1986.267213
5. Perimeter Security Sensor Technologies Handbook, Electronic Security Systems Engineering Division, North Charleson, South Carolina (1997)
6. Yousefi, A., Dibazar, A., Berger, T.: Intelligent fence intrusion detection system: detection of intentional fence breaching and recognition of fence climbing. In: IEEE Conference on Technologies for Homeland Security, May 12-13, pp. 620–625 (2008)
7. Yousefi, A., Dibazar, A., Berger, T.: Application of Non-homogenious HMM on Detecting Security Fence Breaching. In: Proceedings of the ICASSP (2010)
8. Wittenburg, G., Dziengel, N., Wartenburger, C., Schiller, J.: A system for distributed event detection in wireless sensor networks. In: Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks, pp. 94–104. ACM (2010)
9. Mehta, A., Pister, K.: WARPWING: A complete open source control platform for miniature robots. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2010), Taipei, Taiwan (October 2010)