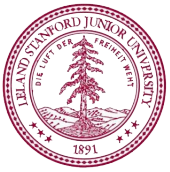

STANFORD
UNIVERSITY

DEPT. OF AERONAUTICS AND ASTRONAUTICS, AA 222 SPRING 2016

Pre-Seismic Earthquake Detection Via Optimized Distributed Sensor Network: A San Andreas Fault Case Study

Final Project

Emily Briere, Jan Peeters Salazar

Due: June 03, 2016

"I know that we'll be able to reach a point in time – maybe only a few years into the future – where we'll be able to issue public alerts stating,

'Stresses at a particular fault seem to be building up deep in the earth's crust and there is an increased chance of an earthquake within the next few days.'

Reaching this point will have an impact. It will save lives.... It will mark a complete turn-around from the present state of affairs, where nobody, in particular no mainstream seismologist, can forecast the approach of a seismically dangerous situation." – F. Freund, SETI Institute, 2015.

1 Introduction

1.1 The Problem

For centuries, earthquakes have puzzled the human race with their magnificent power and quiet approach. Recall the devastating magnitude 9 earthquake that hit Japan in 2011 releasing the mechanical energy equivalent to 2,000,000 Hiroshima atomic bombs, and consider the magnitude 8-9 earthquake anticipated at the San Andreas fault any day now [1]. Despite being one of the largest scale natural disasters, even the most formidable quakes remain largely unpredictable by conventional seismological community, with uncertainty margins surpassing 30 years [1]. Yet in the last twenty years, curious phenomena has led some researchers to posit that maybe these earthquakes do not enter without a trace. In 2003, Dr. Shitov documented an order of magnitude increase in diagnosed neurological disorders in the weeks prior to the 7.5 magnitude Chuya earthquake [1]. In 2008, a group of Chinese doctors found that 3-4 days prior to the magnitude 8.0 Wenchuan earthquake, their lab mice became timid and experienced a substantial shift in circadian rhythm and neurological electro-magnetic spectrum. Even luminous "earthquake light" effects have been noticed [4]. Only recently have these curious events been correlated to an increase in Ultra-Low Frequency Emissions that are emitted from deep beneath Earth's surface where the pre-seismic stresses are developing.

1.2 Prior Work

Dr. Freidmann Freund of the SETI Institute has championed a promising earthquake indicator: the indisputable, non-seismic signals generated as stresses build up in the Earth's crust days to months before the rocks rupture. Despite skepticism from his peers for his multi-disciplinary approach [1], Freund has demonstrated both in the lab and in the field the accuracy of his claims. Indeed, rocks act like batteries under stress, producing massive currents (thousands - millions of Amperes) within Earth's crust that are great enough to be measured from the surface [1]. Freund's lab experiments linearly extrapolated predict a build-up of $1-2x10^9 A/km^3$! Further, when this current changes from a quasi-dc mode to an ac-mode, low-frequency pre-seismic magnetic pulses are produced [2], which explain the documented ULF activity [3]. Freud has worked with Tom Bleier of Quakefinder to install around a hundred air conductivity sensors to detect ultralow frequency (ULF) waves along the San Andreas Fault, and in Peru, Greece, and Taiwan in 2014. With such a distributed network they saw overwhelming increases in the air conductivity prior to any moderate earthquake activities nearby [1,8].However such a distribution of sensors has yet to be optimized with respect to alert time and cost of sensor implementation.

1.3 Project Scope

The focus of this paper is in developing an optimization tool that, given a pre-seismic surface current accumulation function, identifies the optimal spatial sensor network distribution such that T-EQ warning time is maximized for a given number of sensors. Research on this topic, although relatively novel, has ranged the last decade in development of sophisticated semiconductor earthquake fault models. Thus it would be unwise to consider the development of a fully accurate electromagnetic earthquake model within the scope of this project. For the purposes of the work here, we have derived from literature a simplified earthquake model that follows the major trends highlighted. We will use this model as a tool to develop an optimization code that, given any earthquake current density function, can return an optimal distribution of sensors. This allows us to focus on the optimization aspect of the problem, while allowing the possibility that our optimization algorithm could be run on more sophisticated models in the future.

2 Earthquake Model Development

2.1 Model Background

In developing the simplified model, we thus want to capture the key temporal and spatial distributions of quake related currents in accordance to Dr. Freund's current investigations. Freund identified the following key temporal features experimentally:

1. Stress gradients increase significantly as time approaches the time of failure. Freund says that the increase is faster than exponential, but we will model it as exponential here [3]

2. Once a given stress level achieved, there is a nearly constant current flowing, yet as the load is removed the current drops quickly [3]

Further, we assumed the model to have the following spatial characteristics:

1. Current magnitude falls off exponentially with distance from point where primary concentrated stress is developing

2. Current is weaker at higher surface altitudes in comparison to lower surface altitudes (current is proportional to resistance), and therefore topology must be taken into account when extrapolating strength of air conductivity to likelihood of earthquake being triggered

2.2 The Earthquake Model

Following the Project Proposal, one of the most pressing issues regarding this optimization problem was avoiding getting trivial solutions for the grid location (e.g. uniformly spread, clustered around a single point). Hence, some additional characteristics were included in order to make the model resemble the actual behavior of the San Andreas Fault.

The most important inclusion is the geographical mapping of the San Andreas Fault. Following the work of Dr. David K. Lynch [10], a linear interpolation function was created in order to trace the shape of the fault across California. A map of the original fault line (3.1), the interpolated fault line function (3.2) and the corresponding Matlab implementation (3.3) are shown.

Going from a straight line to a curved one representing the fault meant that if the percentage of earthquake epicenters located within the analysis region was to be kept at 99.7% as originally planned, the aforementioned region was to be widened by around $150km$ (the approximate width of the fault). Hence, the analysis region will be now $1500km * 300km$. The distribution of the earthquakes still follows a uniform distribution in x; in y, it follows a $N(0,30)$ distribution from the Faultś mean line at that particular x-location. Various earthquake epicenter location instances are shown in Figure 3.4.

The terrain altitude in this Californian region was also very important, as current is a function of resistance, and in our model the "thickness" of the earth that the current is travelling through from the Earth's crust maps linearly to the current that would be measured at that sensor. We were able to import Governmental 'DTED' or elevation files which we spliced together to cover the entire region of interest. With this, we created a linear function to represent the resistance of the material. Knowing that most Earth quakes originate at the Earth's crust, 30,000 km below sea level, we can map the impact of higher regions on decreasing the current (V=IR) by a scale = $\frac{h-30{,}000}{hmax-30{,}000}$ where h is the height we are at in a given iteration and hmax is the highest point in our region.This scale factor multiplies our current intensity function.

Figure 2.1: Dr. Lynch fault Line representation

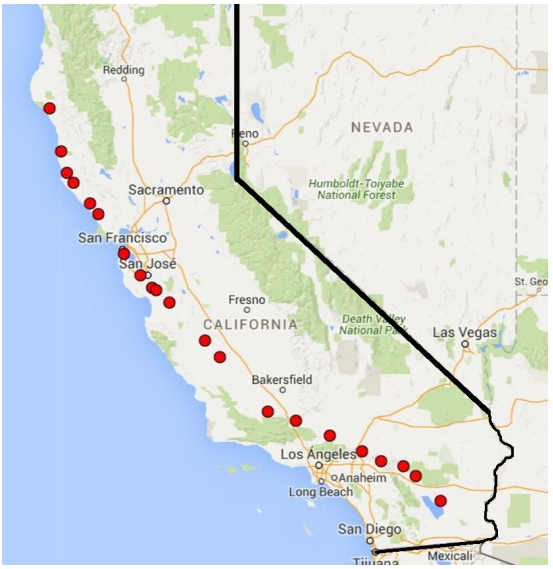

Figure 2.2: The interpolated fault line function

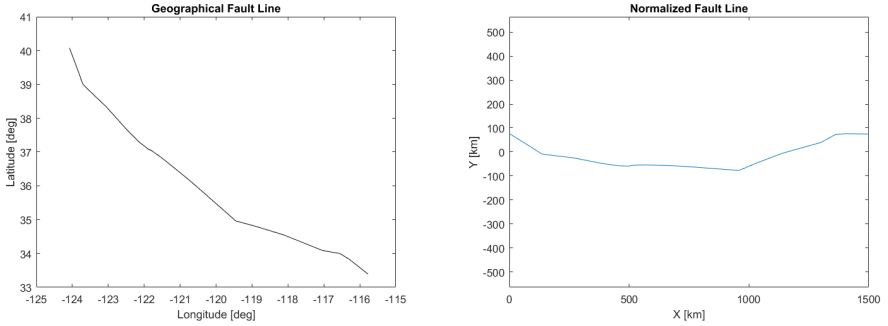

Figure 2.3: Interpolated fault line function on GPS and normalized coordinates

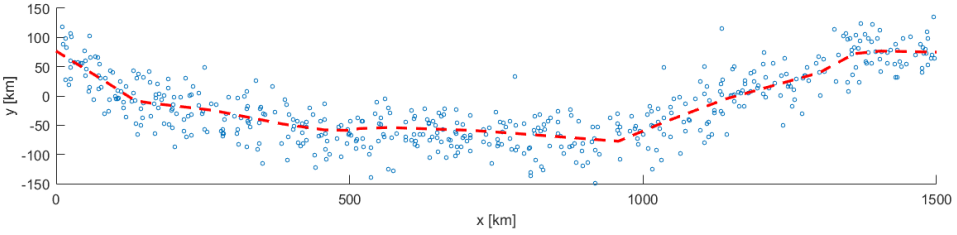

Figure 2.4: Updated Epicenter probability distribution

3 Optimizer Development

On the project proposal, the Quake Current Function (QCF) was first introduced:

$$I(x, y, t_{alert}) = f_{topo} e^{-a\sqrt{b(x-x_c)^2+d(y-y_c)^2}} e^{-t_{alert}/\tau} \tag{3.1}$$

Here, [x,y] is a given sensor location, [xc,yc] represents the epicenter, (a,b,c) are constants we iterated to achieve a function emulating literature results, and f_{topo} is the topological linear scaling previously mentioned. With this function as a basis, a fitness function was constructed

in order to provide it to the optimizer. At any space and time, the QCF will take a value between zero and one. $I = 1$ only occurs at the epicenter on $t_{alert} = 0$ (start of the earthquake). As the project proposal explained, it is desirable to have some warning time before the earthquake manifests itself. For that matter, an alarm will be triggered if the combined L-2 norm of all the measured intensities exceeds a pre-established threshold, in this case 0.5. However, the provided sensors are limited by its precision; hence, only sensors with readings above 0.1 will be considered for the LP norm. In summary, an alarm will be triggered when:

$$I_{LP}(X, Y, t_{alert}) = \sqrt{\sum (I(x_i, y_i, t_{alert}))^2} > 0.5 \tag{3.2}$$

For all

$$I(x_i, y_i, t_{alert}) > 0.1$$

Where X and Y represents the x-y locations of all the sensors. A sample run is shown below for an epicenter location at $(1200, -50)$. Inactive sensors are represented as black dots whereas active ($I > 0.1$) sensors are depicted as red crosses. As it can be seen below, at $t = -900$ not a single sensor shows an active reading, hence $I_{LP} = 0$. As we move forward in time, the sensors in the vicinity of the epicenter start to activate. For example, at $t = -800$, two locations read a combined intensity of $I_{LP} = 0.17216$. As time continues to march forward, I_{LP} increases, as the individually measured intensities and the number of active sensors increase (see $t = -600$). This continues until $t = t_{alarm} = -566$, when a combined LP intensity surpasses 0.5. At this point an alarm will be triggered.

3.1 Deriving an Optimal Sensor Network: Particle Swarm Optimization

Hence, the optimization problem is stated as follows: Maximize t_{alert} in $I_{LP}(X, Y, t_{alert})$ by varying X,Y such that $I_{LP} > 0.5$. For our constraints, there is a natural intuition that we want the sensors to cover the region in full, while bunching more toward the regions where the epicenter is likely to trigger (along the fault). We decided that we want at least one sensor *roughly* every 50 km, but near the fault we want each sensor to have at least 2 sensors both withing 8 km of it for quick triggering of Earthquake warnings. While we originally planned on creating a cost function to balance the cost of the network and the warning time it produced, Dr. Kochenderfer suggested that we rather cap the project at two million dollars. From our research, this conservatively maps to about 400 sensors [11].This means we will optimize the distribution of 300 sensors along the fault, and distribute the remaining 100 across the region to satisfy the 50 km constraint.

To implement this, we began with an experimental Matlab package for Constrained Particle Swarm Optimization that we downloaded from Mathworks [12]. This package has a strong visual aspect in that you can see the particles swarming, however it is not great with constraints (particularly nonlinear, as arises when you are constraining the distances between points). Thus, we split the fault up by 15 points, all of which have roughly equal distances on either side (this includes the ends). We used the individual current function as our fitness function rather than the norm as the norm does not map easily to ones individual fitness unless you are in a massive-hyperspace. We implemented interior point penalty functions to account for all of our constraints, only 'activating' the penalty if it was violated, and varying the weights of each constraint until we consistently got even spreads of points. Beginning at the left most point, we 'triggered' an Earthquake. For this particular earthquake, the points want to be as close to the epicenter as possible without violating the constraint. We lay down one point at a time so that we can properly check the constraints as we go. We are going to run 15 earthquake simulations at different epicenters along the fault, laying down our sensors using particle swarm as we go. For the first earthquake, we lay a sensor at the epicenter, and then the next where the particles swarm to (somewhere 8 km from the center). The next point then has to be roughly 8km from

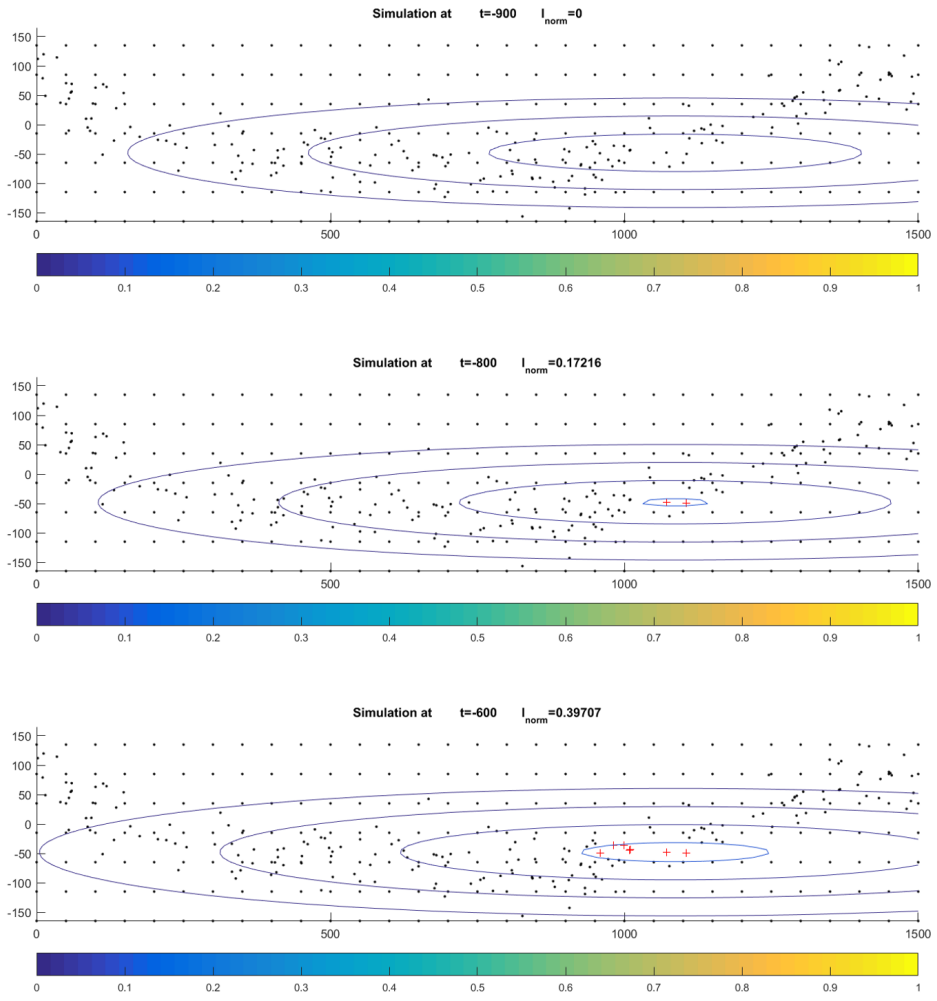

Figure 3.1: Current density function - temporal evolution with measured LP norm

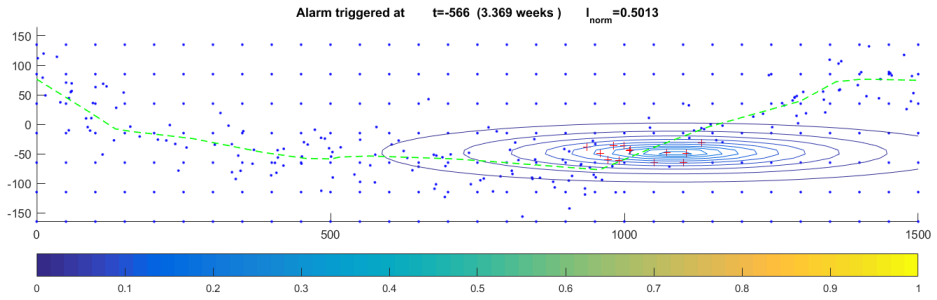

Figure 3.2: Current density function at $t = t_{warning}$, first time step when $I_{norm} > 0.5$

both of those two points, and is thus more limited. Our penalties weren't terribly high, so these were fairly soft constraints to guide the optimization. We continue this until we have 20 sensors lain around the first epicenter. Thus we get a roughly 90kmx20km ellipse of more or less evenly distributed points around the first epicenter. For some reason, the particle swarm happens to align this ellipse with the x and y axis, so we rotate it to match the slope of the fault line. For the

next epicenter, we still have the previous sensors lain on the grid, and the new sensors aren't allowed to be within 8 km of those sensors (soft constraint). And so on, and so forth until all 300 sensors are lain optimally along the fault line. The remaining 96 points are then distributed uniformly, with 6 points each vertically spanning from the 16 points that lie between/around the 15 fault epicenters that we chose. Because our topology data could not be expressed as an explicit function, it could not be used in the generation of the initial optimum selection of points. Rather, these points are later corrected by choosing the point within a 5 km radius with the lowest topology and thus highest current scaling factor. The result is the distribution shown below:

4 Results and Discussion

We looked at three different methods here: a uniform distribution (4.1), a combination of uniform and across-fault random distribution (4.2), and an optimized distribution (4.3) to ascertain which gave the best warning time. See here that the uniform distribution performed the worst (warning of 2.8 weeks) and the average response time from the random distribution (3.1 weeks) is more or less equivalent to the average warning time of our optimized network (3.15 weeks), although our network shows a very slight improvement (0.05 weeks). This demonstrates that in the future, we want to explore more substantial constraints. The topology actually only had a very slight affect on the results as the difference in height above surface level is quite small compared to the depth (Earth's Crust) at which this current is developing.

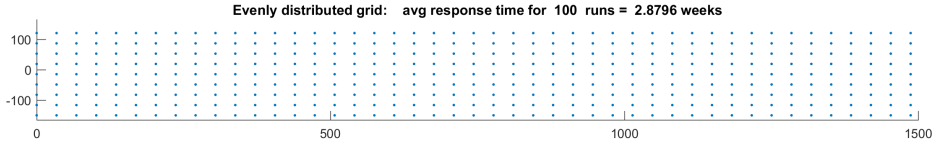

Figure 4.1: Sensor distribution performance for $N = 400$ sensors on uniform distribution

If we want to pursue advancement of this, we want to take into account a more sophisticated current function, as well as a more strict set of constraints (such as perhaps the ability to put sensors on the water). We want to be able to differentiate more substantially between the different types of distributions to get a more substantial affect. We would hope to have better constraints to draw you away from non-trivial results (we forbade our optimized sensor distribution from including sensors on the water, but the uniformly distributed sensors on each grid are allowed on the water and we would correct this in the future).

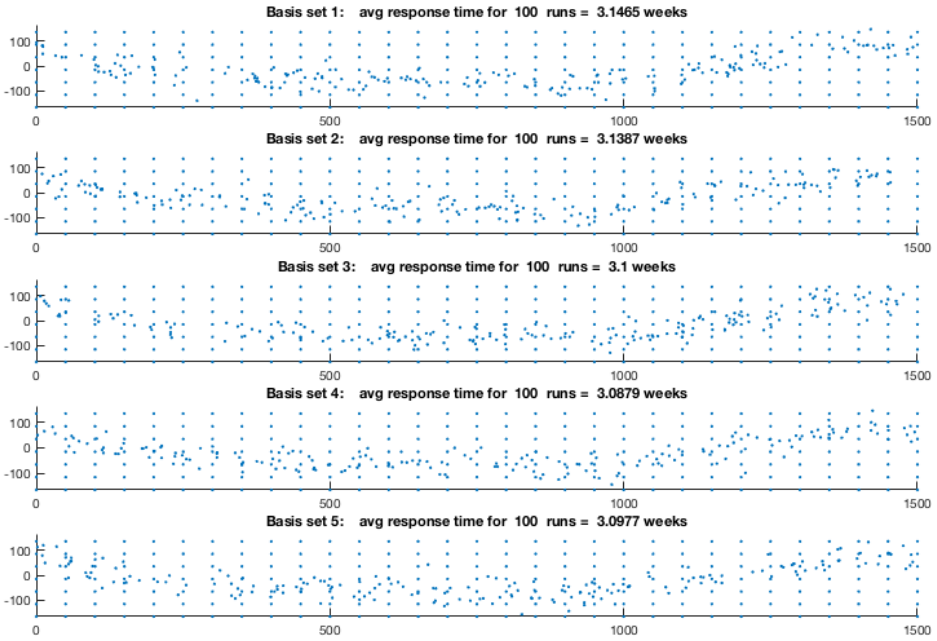

Figure 4.2: Sensor distribution performance for $N = 400$ sensors on different grid+normal distribution

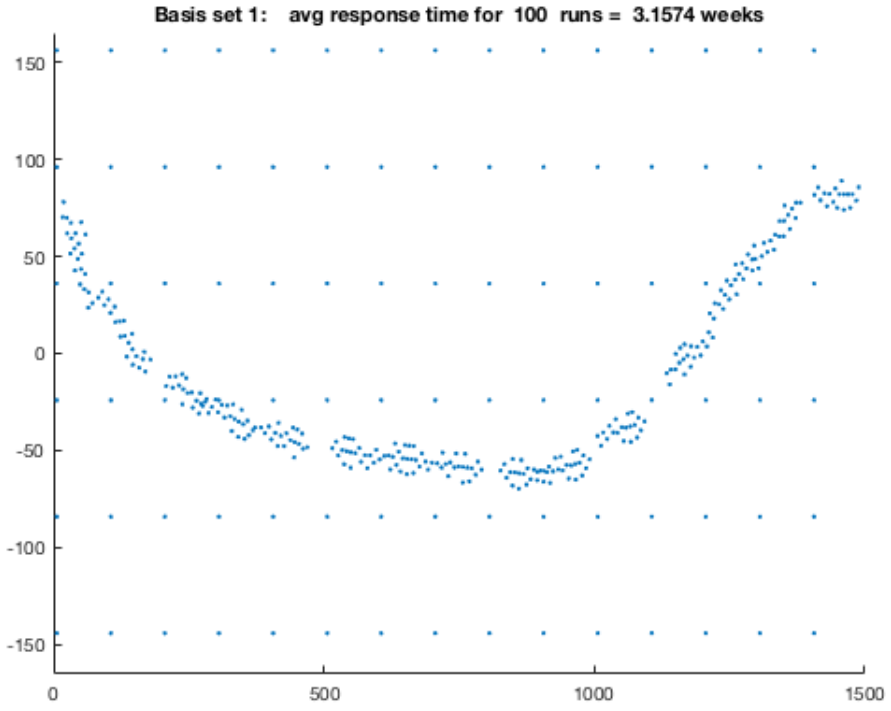

Figure 4.3: Sensor distribution performance for $N = 400$ sensors on optimized distribution

5 Concluding Thoughts

Overall, we have found how difficult it is to design towards multiple constraints and fulfill them all at the same time. A lot of the time, when you put together your constraints, you are effectively deciding what you want your distribution to look like. Going forward, we probably wouldn't use Particle Swarm again. We liked that it featured an individual fitness along side a group-fitness, and were hoping that we could find a way to have the individual fitness be the current at a certain time while the group fitness would be the intensity norm of the group distribution at that time. The problem is, this method would only work if you had a crazy high-dimensional hyper-space that represented all the possible configurations of the sensors in the box and where there was in fact a true global minimum. Since in the end, we couldn't take advantage of such a space with our limited computing power, we found particle swarm to be quite limiting. Whereas we wanted the points to spread out to handle multiple earthquakes at a time, they would instead all swarm to one. Overall, it was a great experience. Hopefully we can continue this work as a AA228 project!

6 References

1. Freund, Friedmann. "Friedemann Freund - The Future of Forecasting Earthquakes." SETI Institute. SETI Institute, n.d. Web. 13 May 2016.

2. Scoville, John, John Spritzer, and Friedmann Freund. "A Distributed Magnetometer Network." ArXiv.org. Cornell University Library, 30 Sept. 2014. Web. 13 May 2016.

3. Scoville, John, Jaufray Sornette, and Friedmann Freund. "Paradox of Peroxy Defects and Positive Holes in Rocks Part II: Outflow of Electric Currents from Stressed Rocks." Journal of Asian Earth Sciences. Journal of Asian Earth Sciences, Feb. 2015. Web. 13 May 2016.

4. Scoville, John, Jorge Heraud, and Friedmann Freund. "Pre-earthquake Magnetic Pulses." ArXiv.org. Cornell University Library, 31 May 2014. Web. 13 May 2016.

5. Freund, Friedemann. "Earthquake Forewarning – A Multidisciplinary Challenge from the Ground up to Space." Acta Geophysica 61.4 (2013): 775-807. Springer. Web. 12 May 2016.

6. Freund, Friedmann. "Pre-earthquake Signals: Underlying Physical Processes." Journal of Asian Earth Sciences 41 (2010): 383-400. Pre-earthquake Signals: Underlying Physical Processes. 9 Apr. 2010. Web. 13 May 2016.

7. Freund, Friedemann. "Seeking out Earth's Warning Signals." Nature 473.7348 (2011): 452. Web. 12 May 2016.

8. Bleier, T. E., C. Dunson, S. Roth, J. Heraud, F. Freud, R. Dahlgren, N. Bryant, R. Bambery, and A. Lira. "Current Progress in Using Multiple Electromagnetic Indicators to Determine Location, Time, and Magnitude of Earthquakes in California and Peru (Invited)." American Geophysical Union, Fall Meeting (2010): n. pag. Web. 12 May 2016.

9. Freund, Friedmann. "How Can Stresses in the Earth's Crust Lead to Bursts of Electromagnetic Signals?" American Geophysical Union, Fall Meeting 2007. N.p., Dec. 2007. Web. 13 May 2016.

10. FLynch, David K. "The San Andreas Fault." San Andreas Fault Line. Geology.com, 01 Jan. 2005. Web. 04 June 2016.

11. Cutler, Bortnik, Dunson, Doering, and Bleier. "System Sciences CalMagNet âĂŞ an Array of Search Coil Magnetometers Monitoring Ultra Low Frequency Activity in California." Nat. Hazards Earth Syst. Sci., 8, 359âĂŞ368, 2008 (n.d.): n. pag. Nat. Hazards Earth Syst. Sci. Web.

12. "Constrained Particle Swarm Optimization - File Exchange - MATLAB Central." Mathworks. N.p., n.d. Web. 01 May 2016.