

An Improved Template Matching Algorithm for LIGO

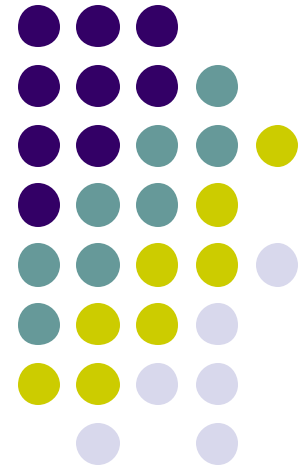


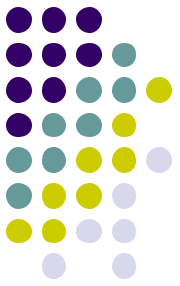
Akash Kansagra

Massachusetts Institute of Technology

Drs. Szabolcs Márka and John Zweizig

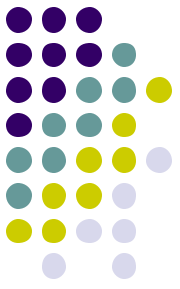
California Institute of Technology





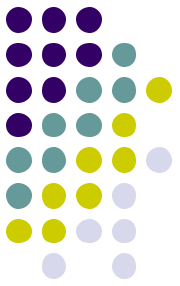
Introduction

- “LIGO will open an entirely new window into the Universe.”
- We have the essential hardware, but lack a reliable method by which to find gravitational radiation signatures in the data collected by the interferometers.
- Template matching needs to be improved.



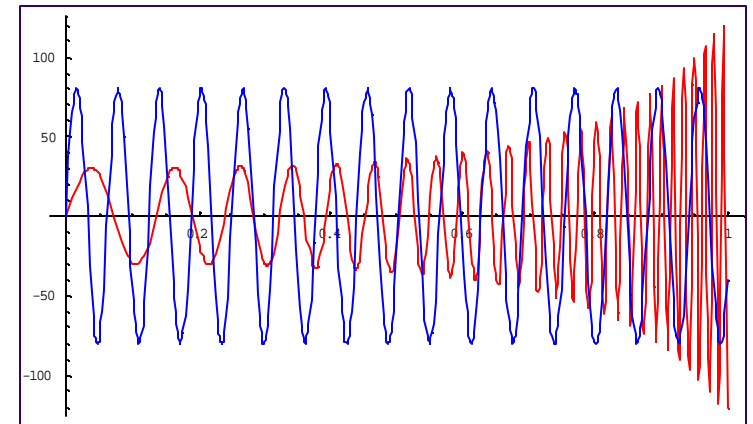
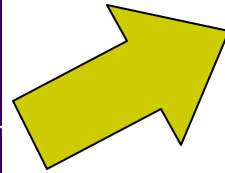
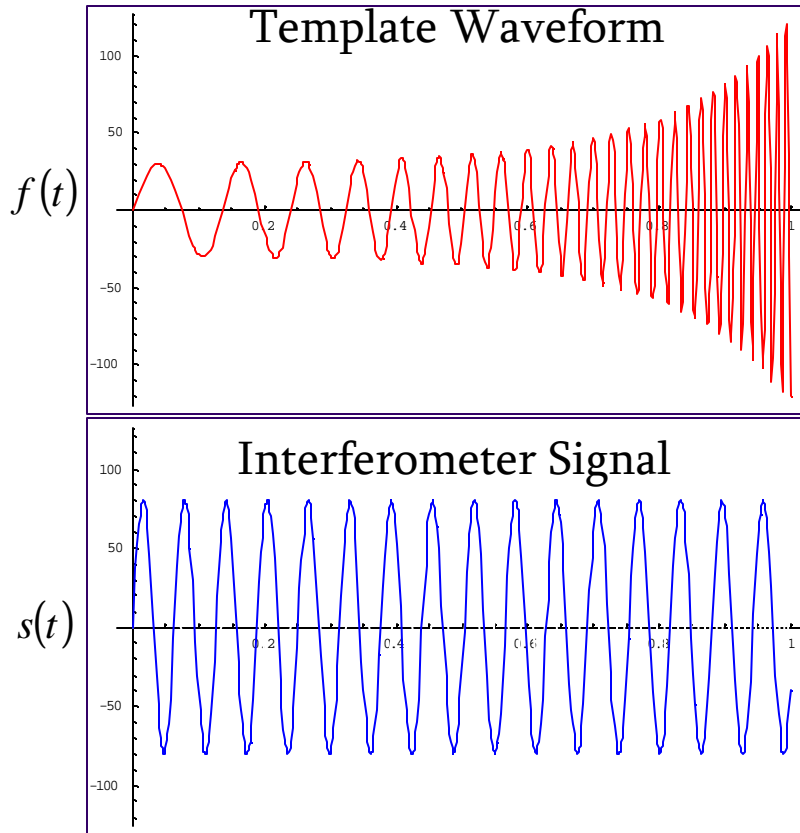
Template Matching

- Some astrophysical objects in the LIGO band are understood well enough that the GWs they emit can be predicted numerically.
- The results of these simulations (templates) are stored in databases.
- The basic function of a template matching algorithm is to see if any of these template waveforms exist in the data.
 - A variety of methods exist to complete this task.



Current Matching Algorithm

- Computes the inner product of the signal and template



Inner product

$$s \bullet f = \int s(t) f(t) dt \neq 0$$

**CANDIDATE
IDENTIFIED**

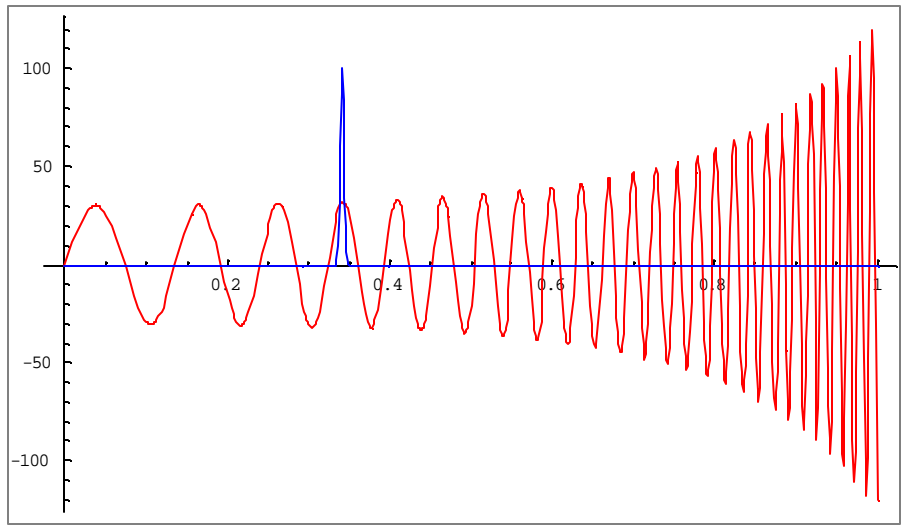
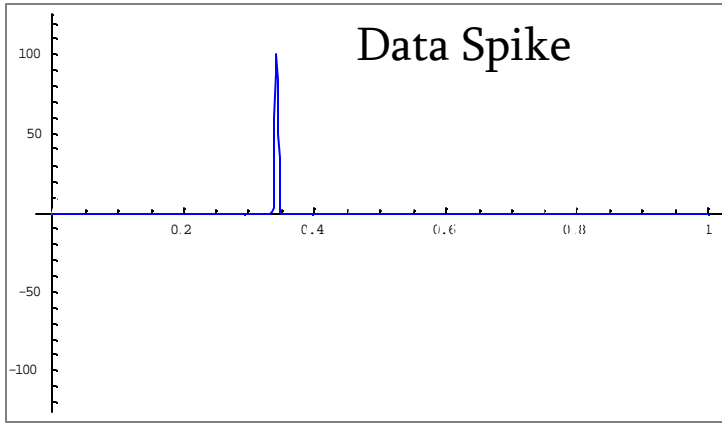
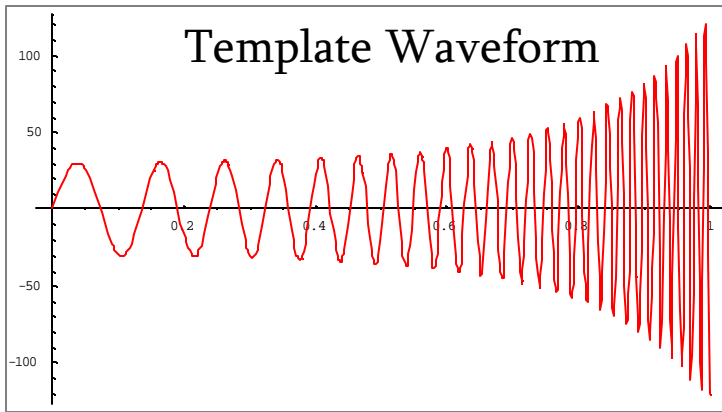


So What's Wrong?

- Identifies far more candidate signals than it should; unrealistic by orders of magnitude.
- We don't really know what “realistic” rates should be, but at the current rate of detection, you can reasonably expect to encounter a BH-BH merger on your flight home.
- So, most candidates are false positives (but not all).
 - A major contributor to false identification is the existence of momentary data spikes.



False Identification in Action

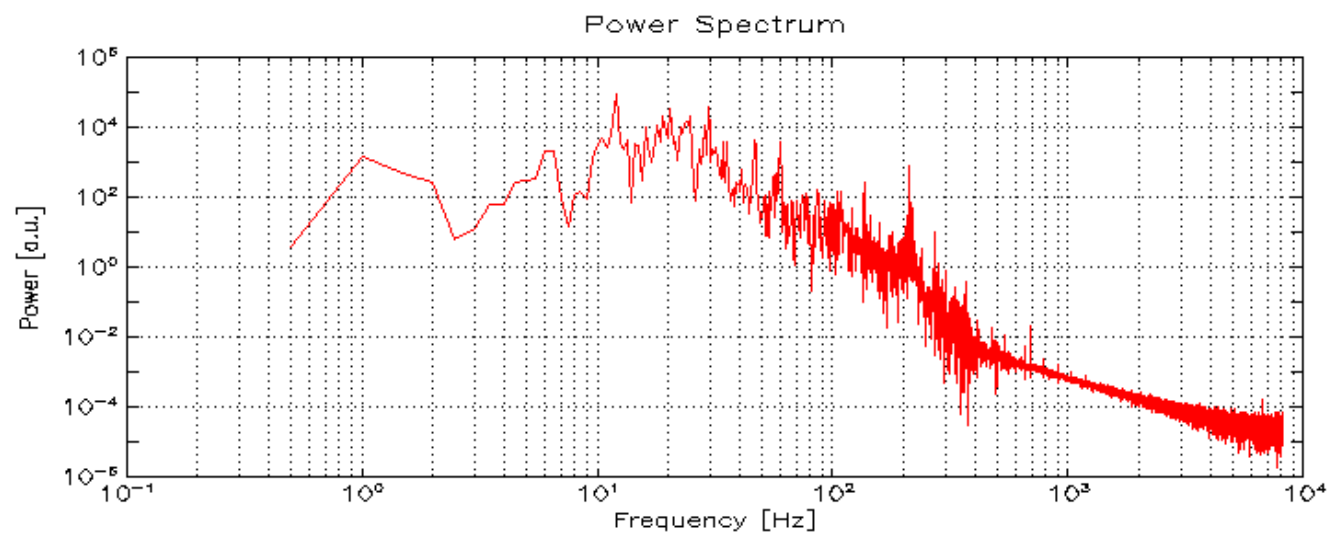
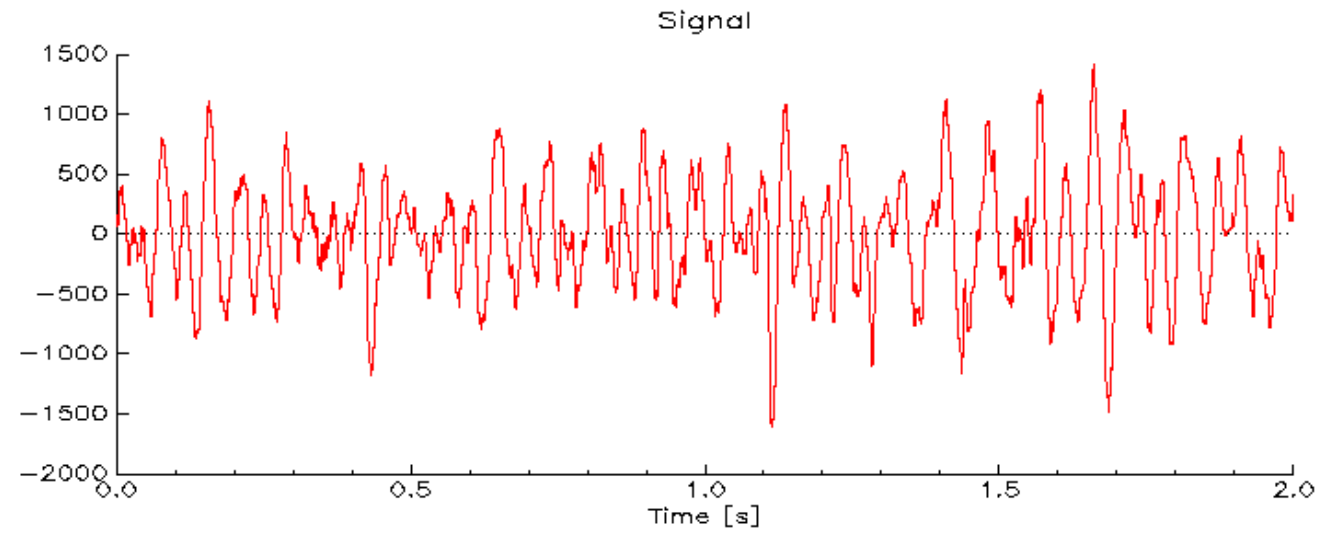


$$s \bullet f = \int s(t)f(t)dt \neq 0$$

Identified as a candidate, but it obviously isn't.

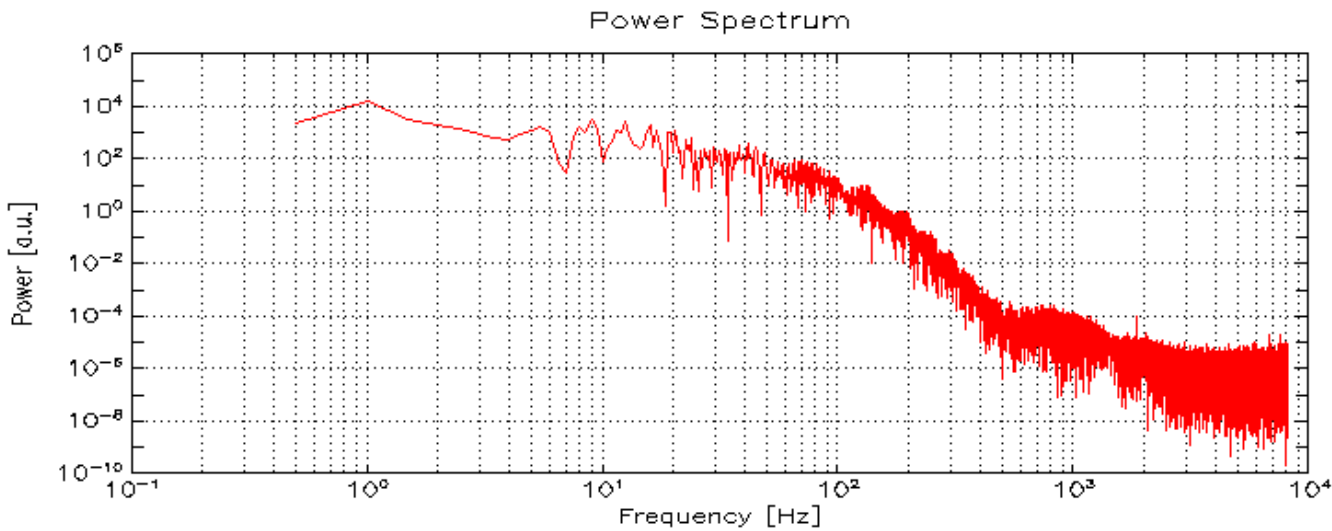
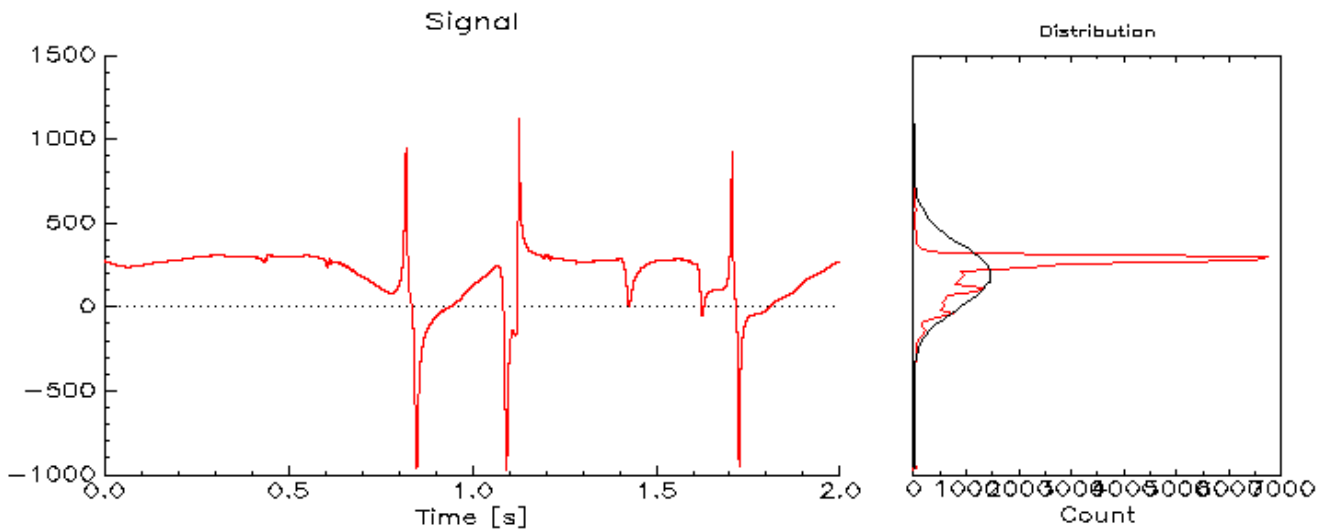
The spike or spikes become the main contribution to the inner product, which is nonzero as a result.

The only requirement is that the spike dominate the noise, which is usually the case.



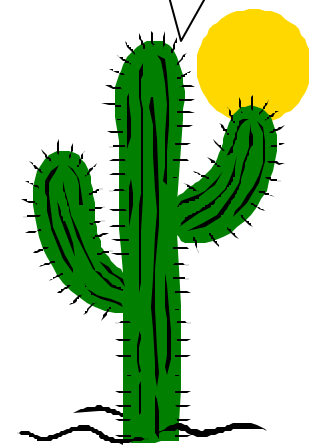
H1:LSC-AS_Q + 0712814307

How we want our signals to look...

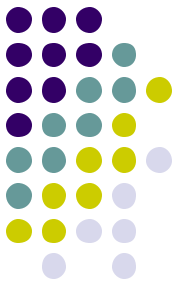


H1:LSC-AS_Q + 0712814909

Look at all those spikes!

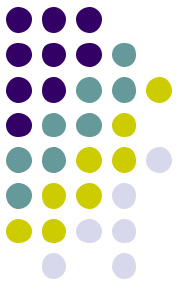


How our signals sometimes look.



Alternative Techniques

- How do we avoid false identification due to data spikes?
 - Divide the waveform into several regions, and compute the inner product for each region.
 - Significant improvement over previous method, but still susceptible to multiple data spikes.
 - How orthogonal are templates?
 - Best approach: χ^2 fitting
 - Provides a rigorous method to match observations to templates.
 - Not susceptible to data spikes!
 - The downside: computationally challenging.



What Next, Einstein?

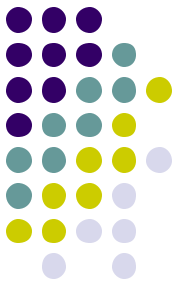
- EINSTEIN Identifies Non-terrestrial Signals That Exist In Noise
- Chooses template parameters (ex. inspiral masses and amplitude) that minimize χ^2 on a specified region of parameter space.
 - The algorithm actually minimizes the mean square deviation, which is equivalent to minimizing χ^2 .

$$c^2(\vec{z}) \sim \langle [s(t) - T(\vec{z}; t)]^2 \rangle$$



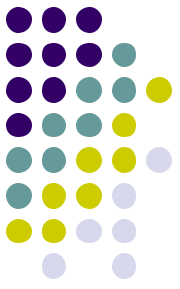
Modus Operandi

- User specifies lower bound, upper bound, and step size for each free parameter. A lattice of test points is thus generated.
- χ^2 is computed at each lattice point.
- Coordinates of the lowest χ^2 are returned.
- This is a brute force method!



Modus Operandi (continued)

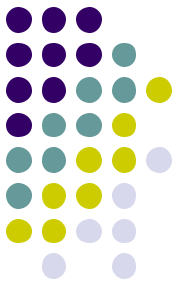
- The template corresponding to these coordinates is subtracted from the data.
- A residual distribution is generated.
- The user then has the option to remove data lying past three standard deviations. Any data spikes can be removed in this way.
- If data is removed, the algorithm iterates and refits the reduced data.



E tu, Brute?

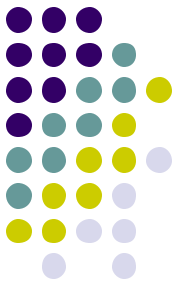
- Locate the global minimum on the χ^2 surface.
- The topology of the surface is highly complicated, even in just two dimensions. It is characterized by extremely narrow features and dramatic fluctuations.
 - Gradient search is out of the question.
 - Migrad and Simplex methods don't work either.
 - Most standard minimization algorithms converge to a meaningless local minimum. We absolutely require a globally convergent algorithm.





Managing Brute Force

- We resort to brute force only because it works.
- Fortunately, it is trivial to split up a lattice algorithm for a massively parallel system (Beowulf clusters).
- With no need for communication between nodes, execution speed will be almost perfectly proportional to the number of nodes.
- Also, we have given EINSTEIN the capability to function as a coincidence monitor. If set to operate in this mode, EINSTEIN will be triggered less often.

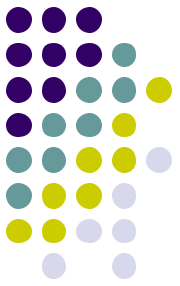


Other Issues

- Noise isn't necessarily Gaussian.
- Even if it is, the correlation time is quite large. You have to wait a while before the noise starts to look randomly distributed.
- As a result, EINSTEIN works best if long observation intervals are specified.
- The same issue arises when trying to match phase between LHO and LLO. Large correlation time makes phase matching a nontrivial task.



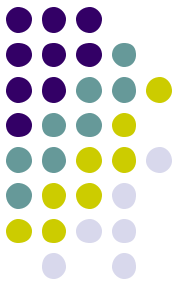
Selected Examples of Non-Gaussian Noise



Bullets, fires, and crashing SUVs, along with other, less curious sources, produce noise that does not obey Gaussian statistics very well. Recall the image with the data spike; it was very non-Gaussian.

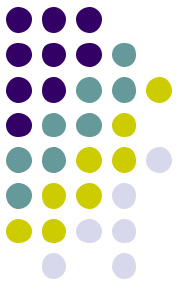


SUVs are known to introduce long correlation times.



Current State of EINSTEIN

- Working, but not fully functional.
 - More parameters need to be freed to accurately model sources (ex. inclination).
 - However, each additional lattice dimension drastically increases execution time.
- Still looking for globally convergent algorithms.
 - Evolutionary algorithms such as simulated annealing may hold promise.
 - Ultimately, however, they depend on random numbers to find the global minimum. They may not always be reliable on the particular surface we are working with, and the results may not be perfectly reproducible.
 - If they can be made to work, one can add parameters to the model without worrying about increased execution time.

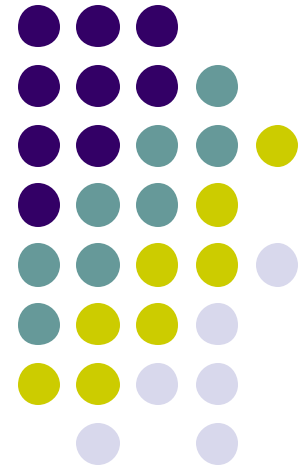


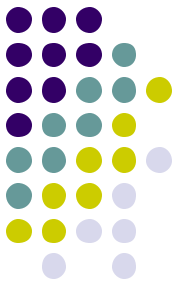
Simulated Annealing

- Boltzmann probability: $p(E) \sim e^{-E/kT} \Rightarrow p(\text{transition}) \sim e^{-\Delta E/kT}$
- χ^2 is treated as energy, and characteristic temperature is defined.
- Test point generally follows downward gradient, but is allowed to escape wells with the above defined probability.
- SA is essentially an iterative Monte Carlo algorithm. This is how nature works (consider statistics with on a PES).
- Algorithm iterates many times, and the temperature is gradually lowered. Typically, the point will settle in the global minimum.
- Annealing is used industrially to produce metals that are virtually perfect; SA attempts to mimic this behavior.
- SA is used extensively in the semiconductor industry to minimize interference between components.

Bonus Slides

Announcing Three Other Programs,
Products of GDS/DMT





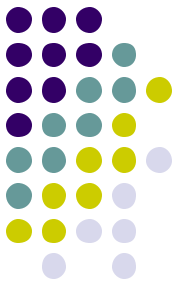
ALADDIN

- ALADDIN Looks At Data Distributions that Interferometerists Need
- GUI-driven program based on ROOT for viewing current and frame data. Provides a variety of graphics capabilities.
- I won't describe it in further detail since it is the work of Ryan Williams, a SURF student from LSU.
 - You may attend his seminar on the 20th at Caltech.
 - Or, you may read the documentation.
 - Or, you can try it out! Not yet stable, but it will soon be available in the CVS tree.



VIRGO dataDisplay

- Very powerful tool for reviewing frame files. Can generate virtually any type of plot from the common frame format.
- Similar to ALADDIN, but with added functionality within the context of frames (multiple frames, etc.). However, it cannot monitor data broadcasts.
- Originally written for VIRGO by Didier Verkindt and Damir Buskalic. Current version is 8.09 (very stable).
- Ported to LIGO by Benoit Mours and myself (mostly Benoit).

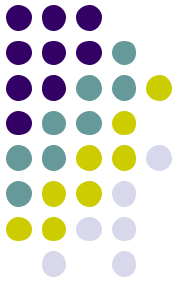


SIGMA

- Simple Graphical Monitoring Applet
- Can use current data or frame files.
- Similar to ALADDIN/VIRGO dataDisplay
 - Fewer plot types, but blazing fast!
 - Time series, power spectra, histograms, Gaussian fit
 - Generates very high resolution images. Based on GNU/plotutils (instead of ROOT).
- Stable release, already installed on DMT machines.
- Available through CVS repository
(GDS/gds/Monitors/sigma/)



Contact



Akash Kansagra
kansagra@mit.edu