What is a Sensible Architecture
for an Antelope-Powered Web Service?

HTTP

Frameworks

Messaging

Services

Distributed NoSQL

Ingest

Native

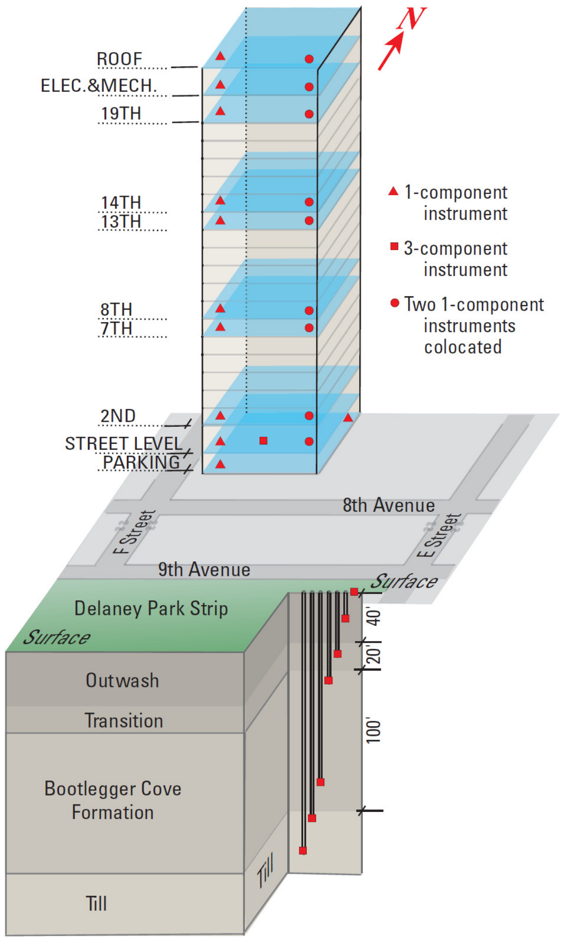
React

To the web

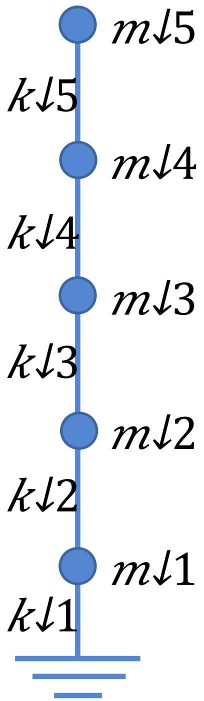
To the sensor network

Structural Health Monitoring

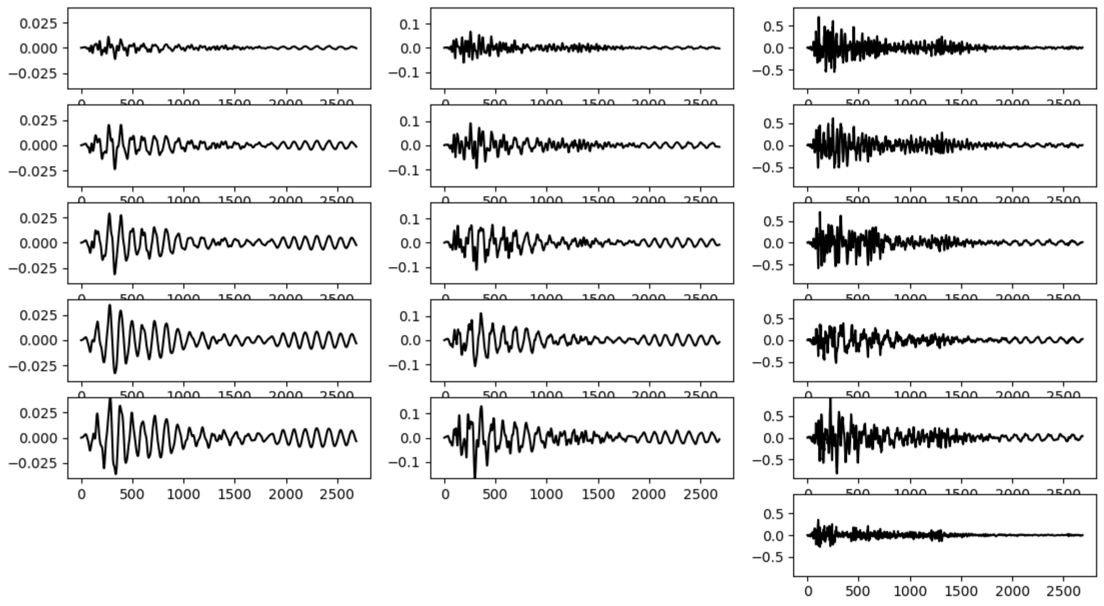
Instrumentation



Model



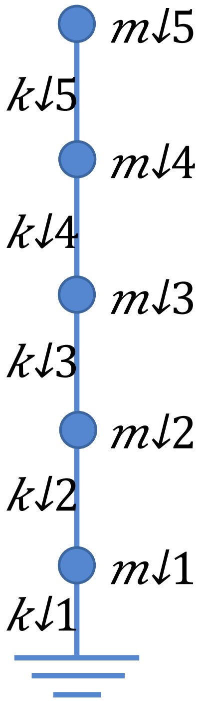
Signal



Atwood Building, Anchorage

UCSB Geotechnical Array monitoring program

Equations of Motion and Objective Function



$$Mu(t)+Cu(t)+Ku(t)=-Mrg(t)$$

Put into state space form
Discretize

$$x(k+1)=Ax(k)+Bg(k)$$

$$y(k)=Cx(k)+Dg(k)$$

Unwind recursions

$$x(k)=\sum_{l=1}^{k} A^{l-1} B g(k-l)$$

$$y(k)=\sum_{l=1}^{k} CA^{l-1} B g(k-l)+Dg(k)$$

Model for accelerations in unknown physical parameters

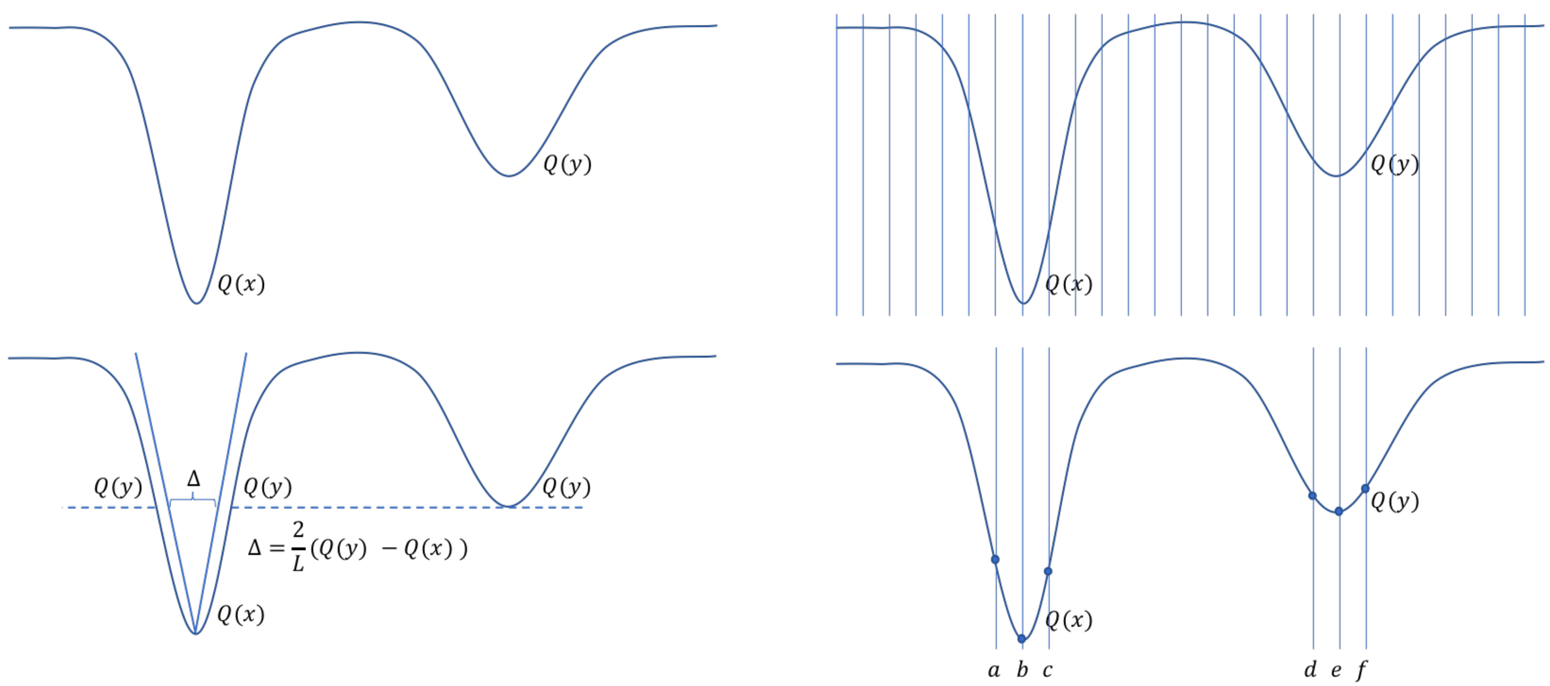
Objective: sum of squares of acceleration residuals at all DOFs

$$x^{*}=\{M^{*},K^{*},C^{*}\}=\arg\min \frac{1}{2}\|f(x)\|_2^2$$

$$=\arg\min \frac{1}{2}\sum_{k=0}^{T-1}\sum_{i=1}^{n} f_i^2(x,$$

$$f_i(x,k)=y_i(x,k)-u_i(k)$$

Black Box Optimization



Randomized Black Box Coordinate Search

```
k=0
while not terminate:
    for i=shuffle(1…n):
        [a,c] = bracket_min(x_i)
        x_i = arg min Q(x_i)
    end for
    k=k+1
end while
return x
```

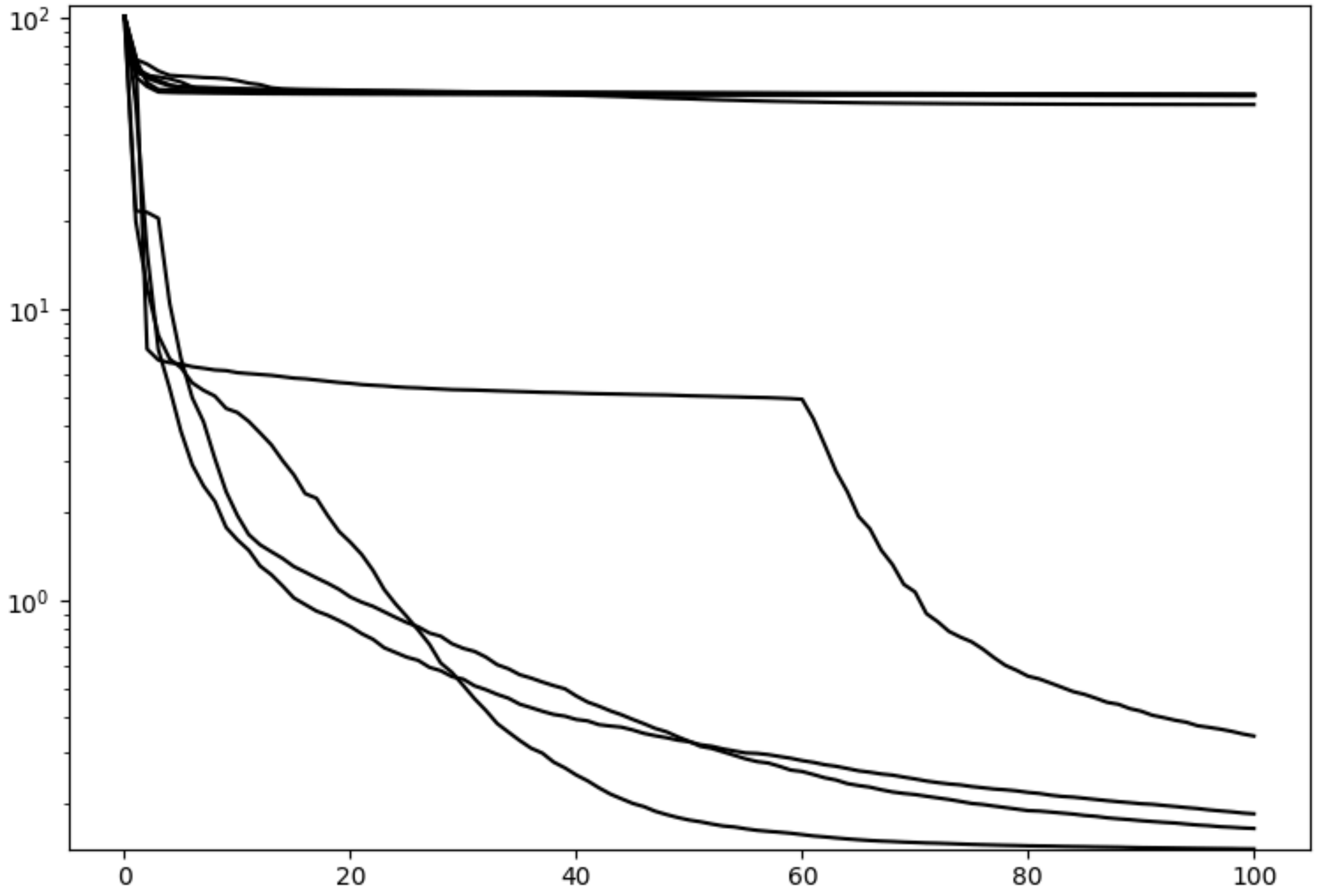
1-D Golden Section Search

Simulation Procedure

Choose parameters → Continuous time State space → Discrete time State space → Recursions → Acceleration model

$x=\{M,K,C\}$ → A_c, B_c, C, D → A, B, C, D → $y(k)=\sum_{l=1}^{k} CA^{l-1} Bg(k-l)+Dg(k)$

Performance of Solutions over Multiple Runs



Local Nonlinear Search

The approach works well, but can be slow. Can we improve efficiency? Yes:

- GPU versions of random coordinate search?
- Improved algorithms based on local models*

$Q(x) \approx \tilde{Q}(x) = a_0 + a^T x + 1/2\, x^T H x$ Local quadratic model at iteration k

$\tilde{Q}(x_i) = Q(x_i) \quad \forall\, x_i \in X$ Interpolate against nearby simulations to find model (linear system, simplex or random)

$x_{k+1} = \arg\min \tilde{Q}(x_k)$ Minimize model

$C(x_{k+1}) = (x_{k+1} - x_k)^T (x_{k+1} - x_k) - \Delta^2 = 0$

at distance Δ from current iterate

Lagrange multipliers gives a solution

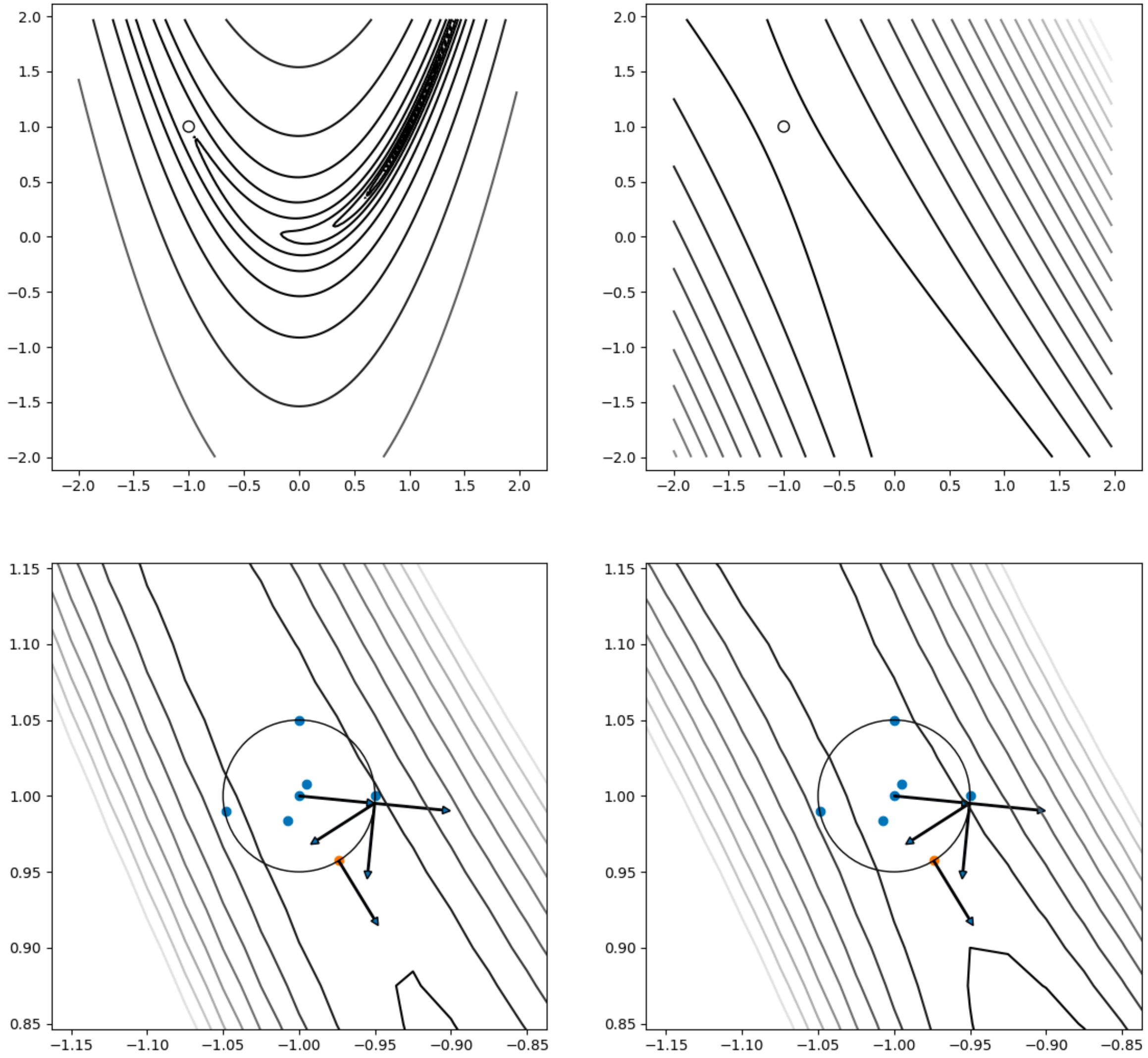
$x_{k+1} = -\sum_{i=1}^{n} (u_i^T \nabla Q(x_k)/\lambda_i - \lambda) u$

with Eigendecomposition $H = U \Lambda U^T$

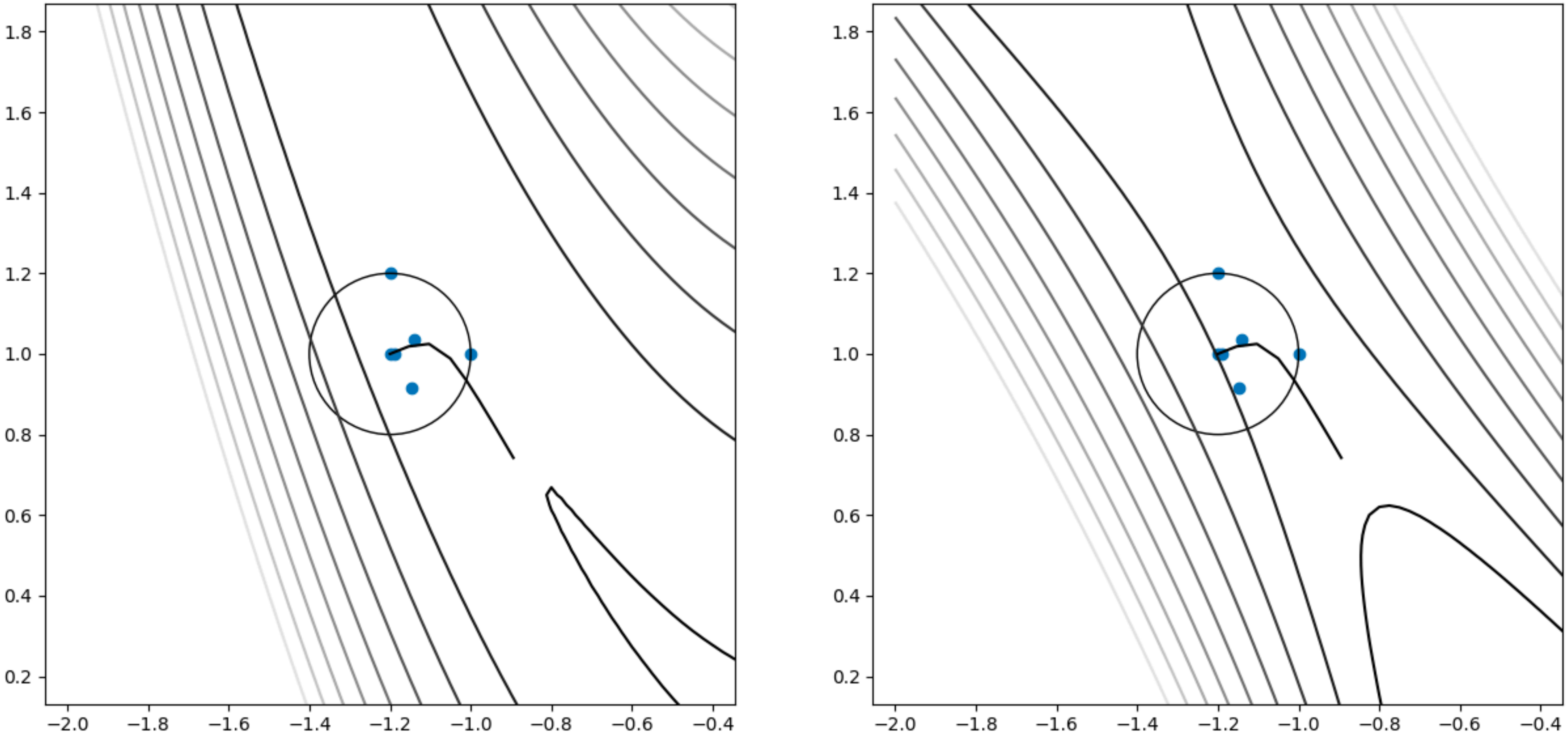
and λ given by $\sum_{i=1}^{n} (u_i^T \nabla Q(x_k)/\lambda$

- A different solution direction arises as we grow Δ.
- This is a 1-D nonlinear search direction.
- Apply grid search and golden section search as before on this nonlinear direction.
- Only searching in 1 NL dimension instead of n.
- Using local info produces good descent.
- ~100x faster.
- Provably convergent.

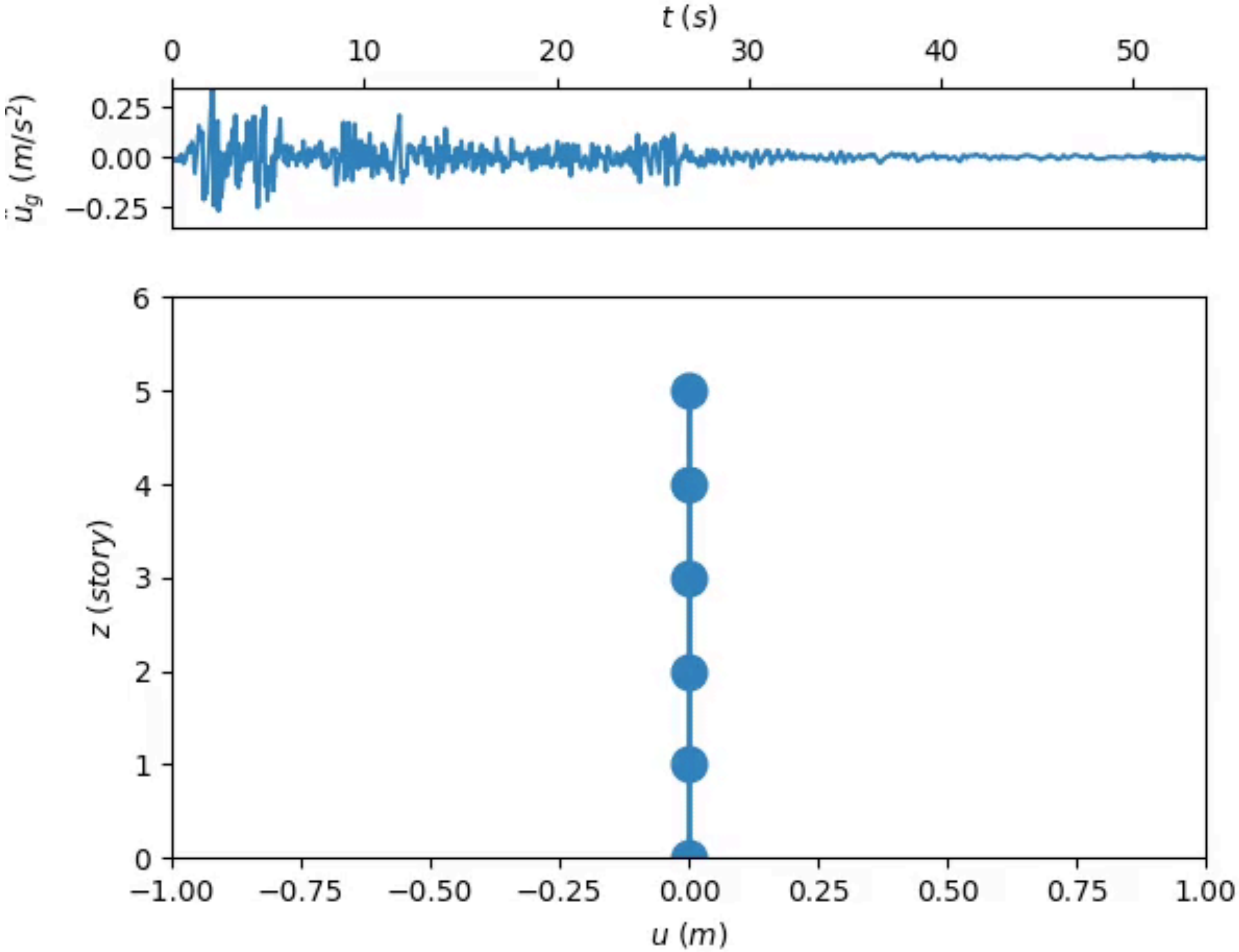
Local Nonlinear Search: Illustration



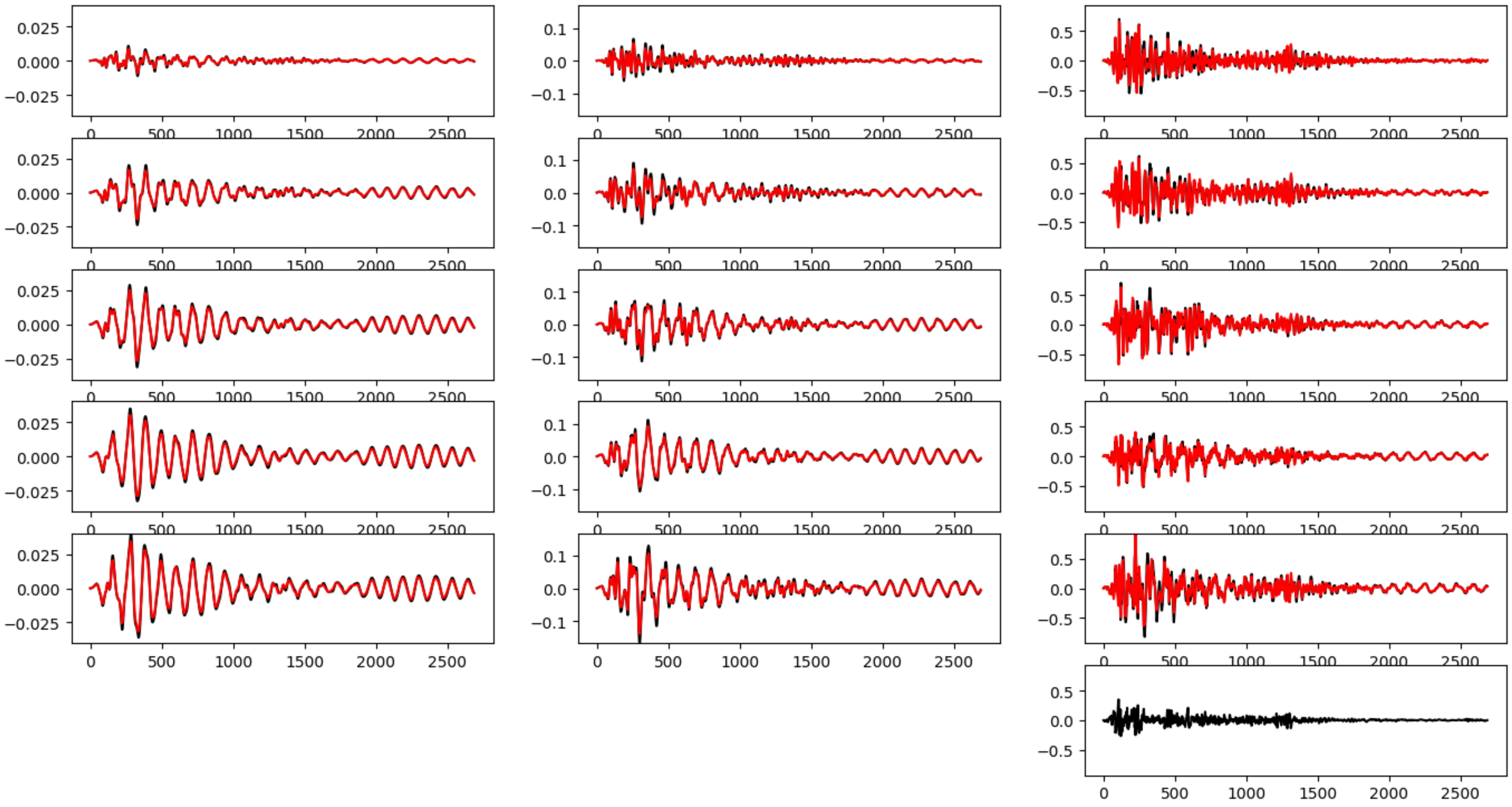
Local Nonlinear Search: Illustration



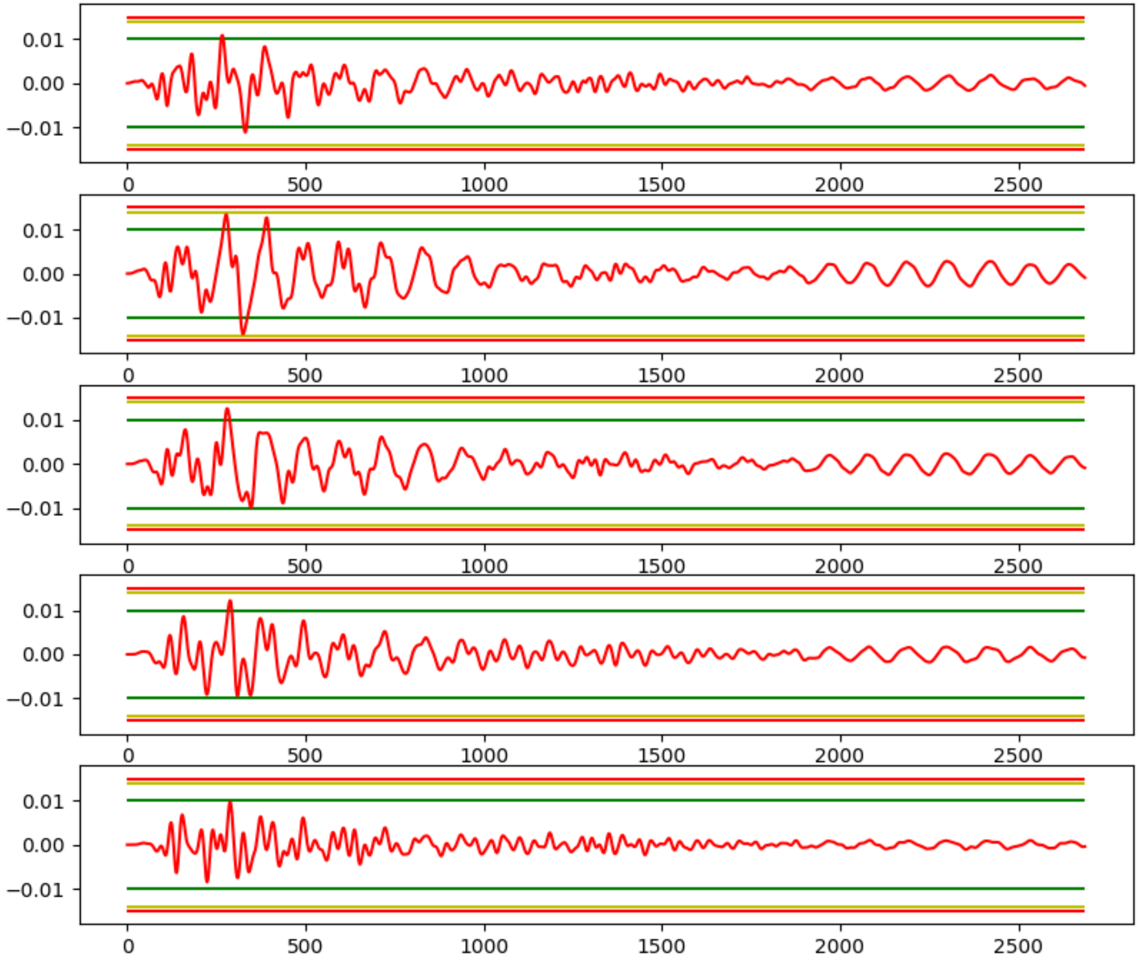
Example Dynamics

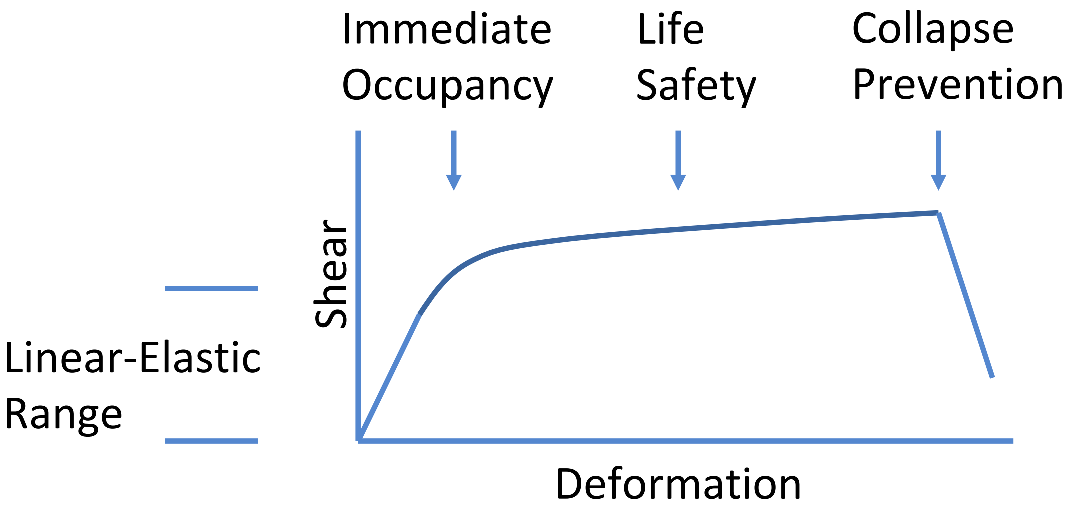


Dynamics and Predictions



Damage Estimation:
Interstory Drift





Damage Estimation:
Predictive Degradation