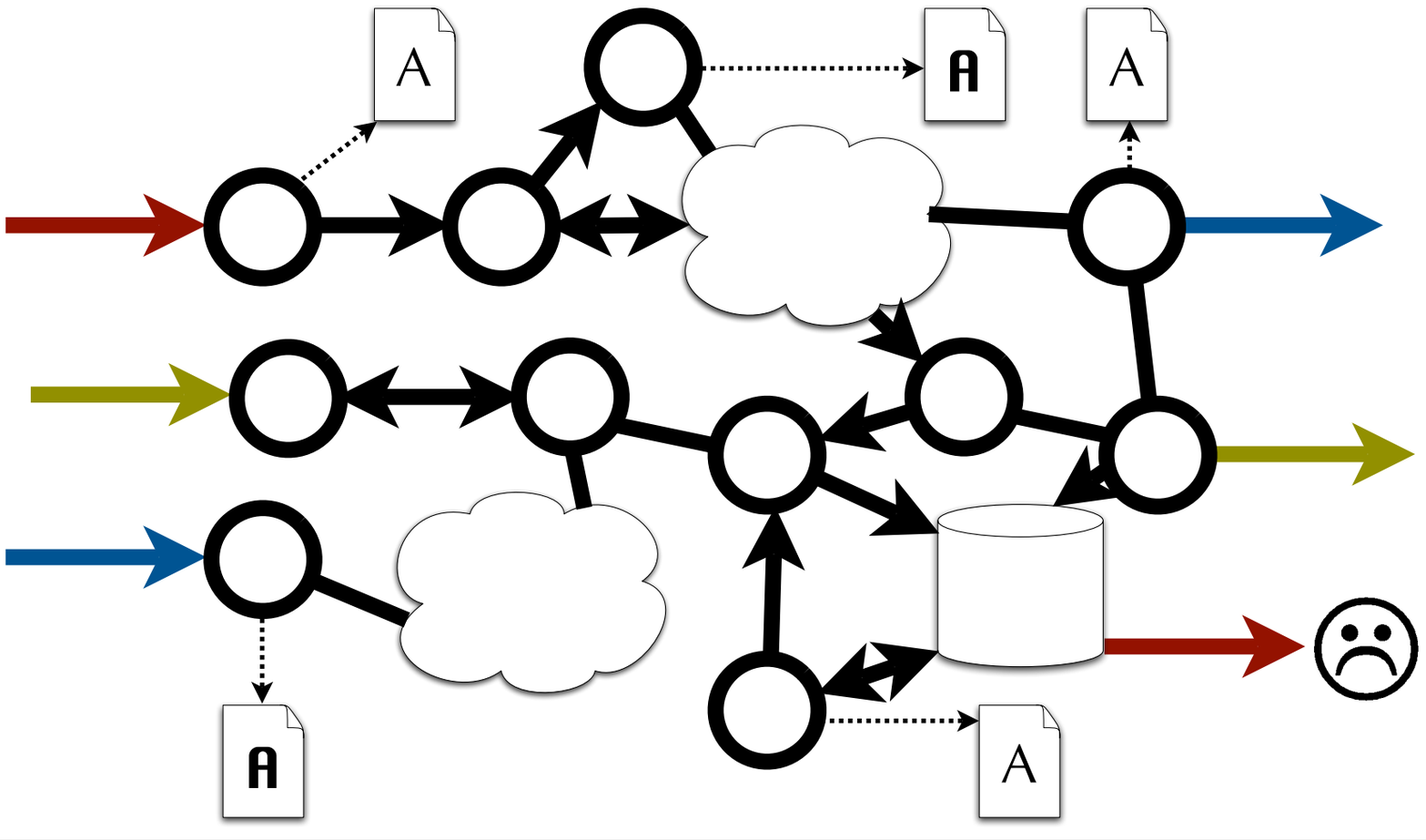
Using Influence to Understand Complex Systems

Adam Oliner and Alex Aiken
Department of Computer Science
Stanford University

A Typical System

- Complex interactions (loops, shared resources, etc.)
- Noisy and incomplete measurements



Goal and Approach

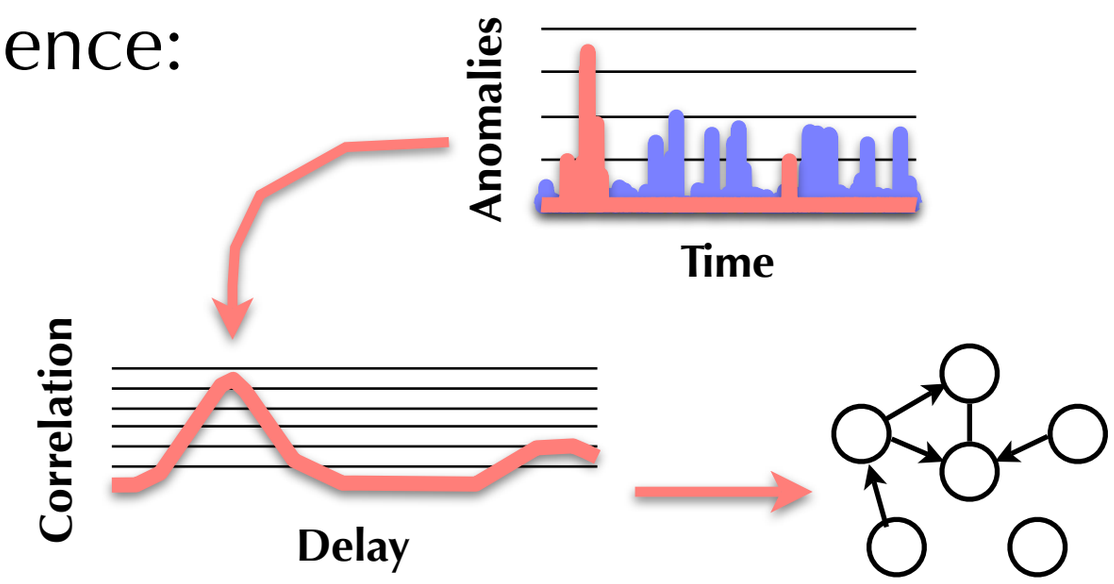
Infer which components and interactions were involved with the problem.

- We use **influence** to perform this inference
  - Statistical correlation that captures implicit interactions
  - Three steps to compute influence:

    (i) identify anomalies

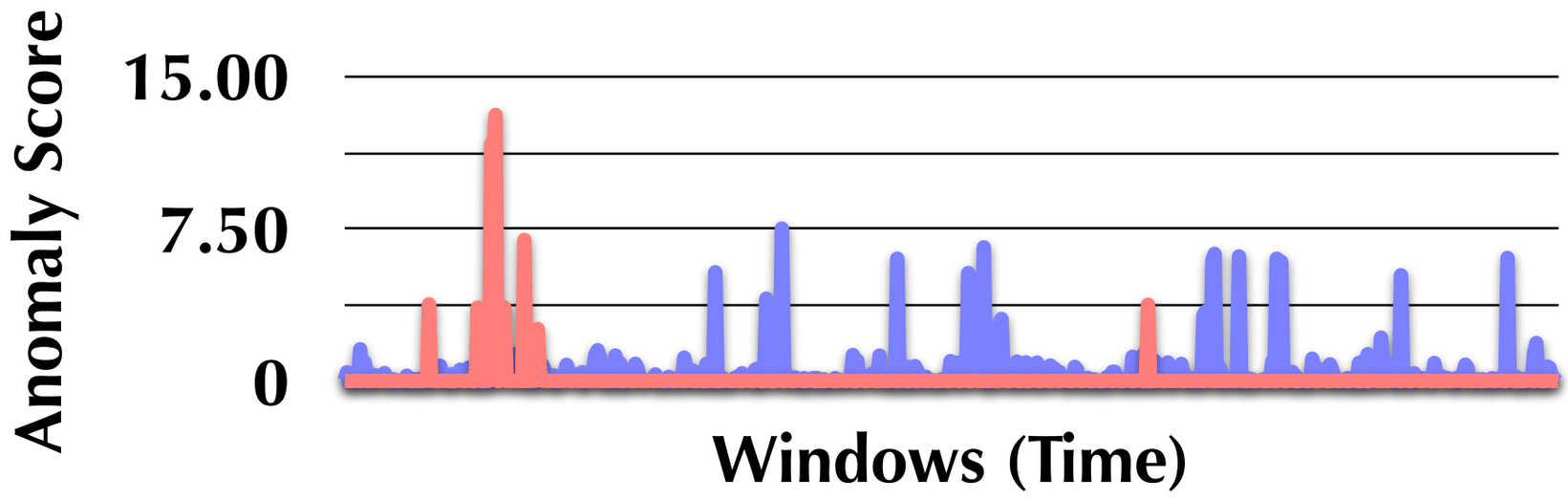
    (ii) correlate in time/space

    (iii) infer influence

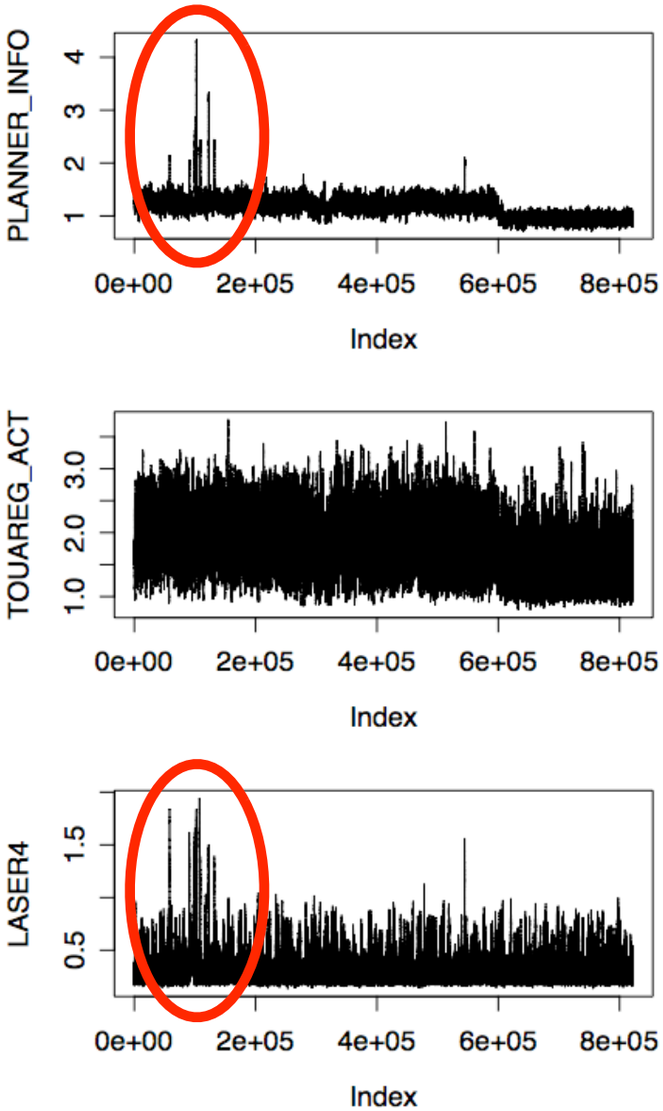


(i) Anomaly Signal

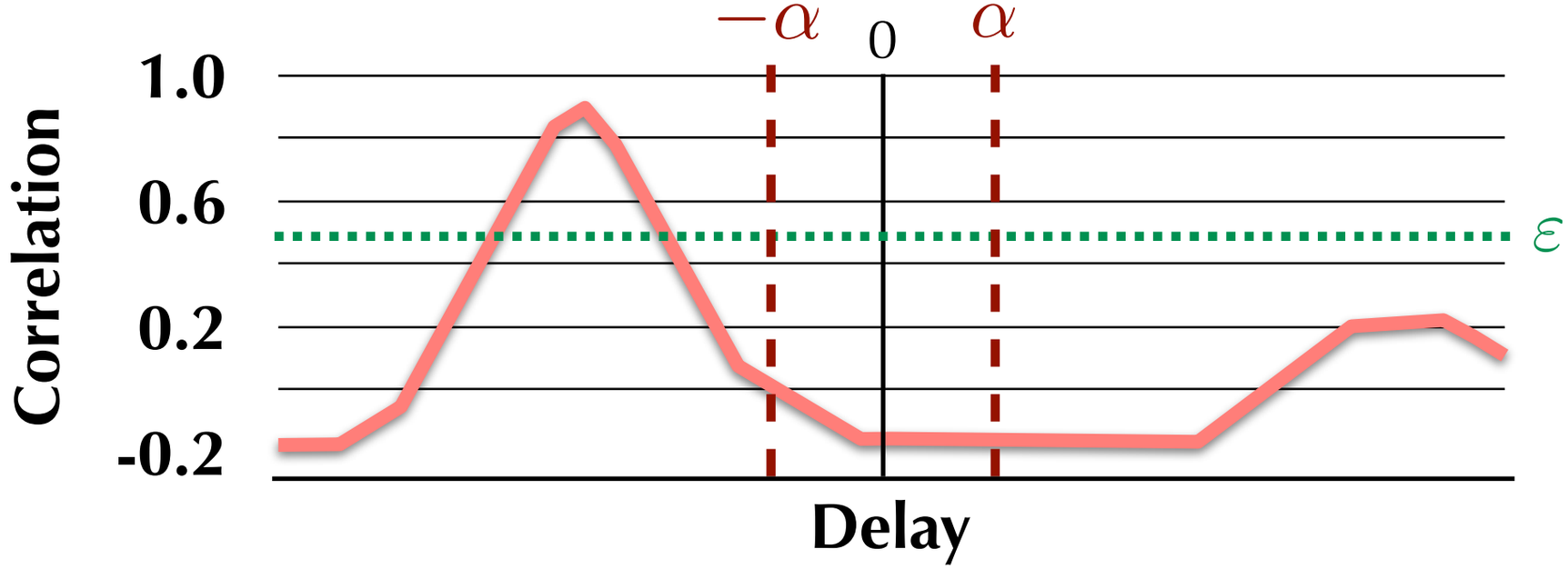
- Surprise over time: $\Lambda_j(t)$
- Component model
  - Compare heterogeneous components
  - Synthesize hypothetical behaviors



(ii) Cross-Correlation

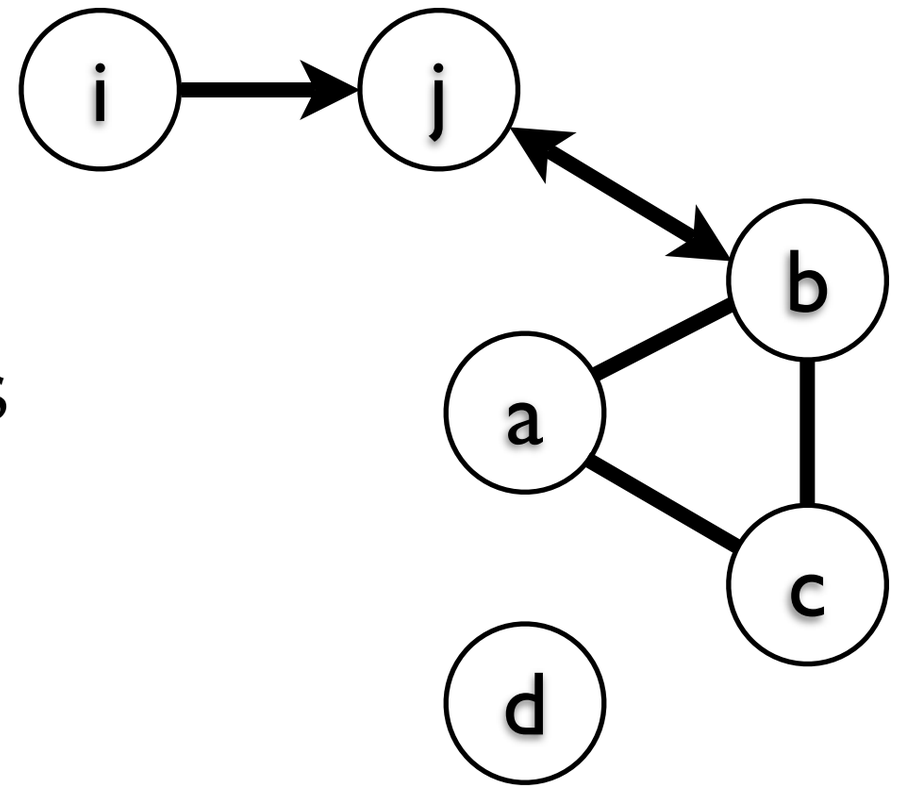


$$(\Lambda_i \star \Lambda_j)(t) \equiv \int_{-\infty}^{\infty} \frac{[\Lambda_i(\tau) - \mu_i][\Lambda_j(t+\tau) - \mu_j]}{\sigma_i \sigma_j} d\tau$$



(iii) Influences

- Encode as a Structure-of-Influence Graph **(SIG)**
  - Various directionality
  - Loops and cliques
  - Time-varying
- Captures implicit interactions
- Well-defined despite noisy or incomplete data



Stanley's Swerve Bug

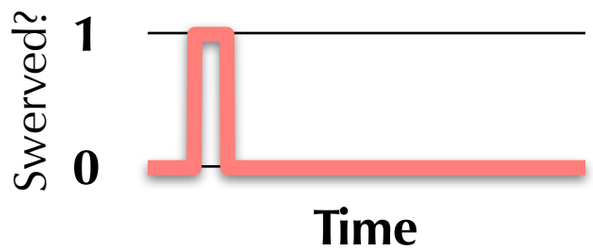
- Won 2005 DARPA Grand Challenge
- Tough bug:
  - Implicit timing dependency
  - Nondeterministic
  - Two month search
- 17 Swerving Incidents

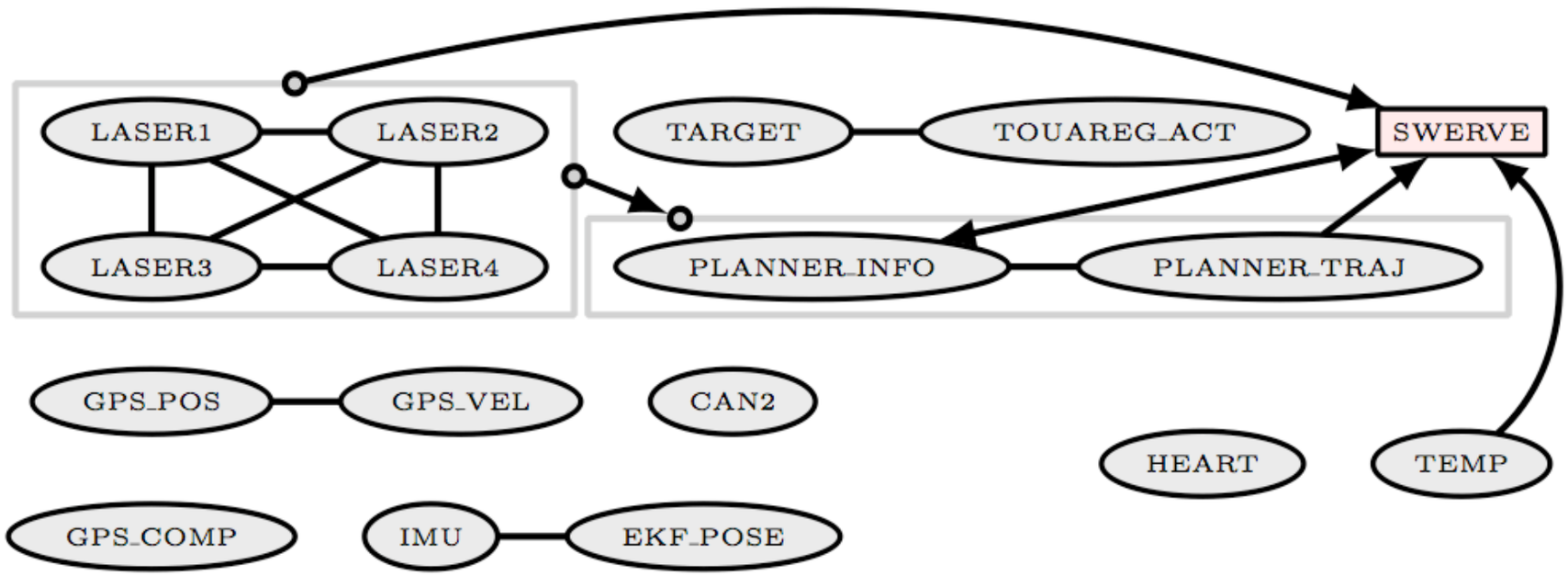
Isolating the Bug

- Model components using timing

$$\Lambda(t) = \sum_{k \in R(t)} R(t,k) \log_2 \frac{R(t,k)}{H(k)}$$

- Specify bug as time interval
- Build a SIG for Stanley





Recent Results

- Junior: modeled Stanley's successor
- **Syzygy**: community epidemic detection
- **Qi**: current project
  - Query language
  - Supercomputers [DSN 07] and clusters
  - Nodeinfo algorithm [ICDM 08]
- We are always looking for data!