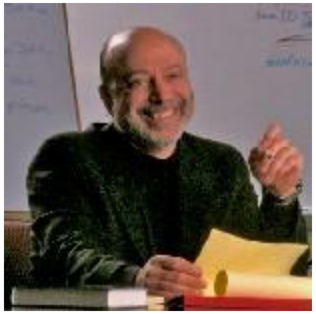
Autonomic Computing

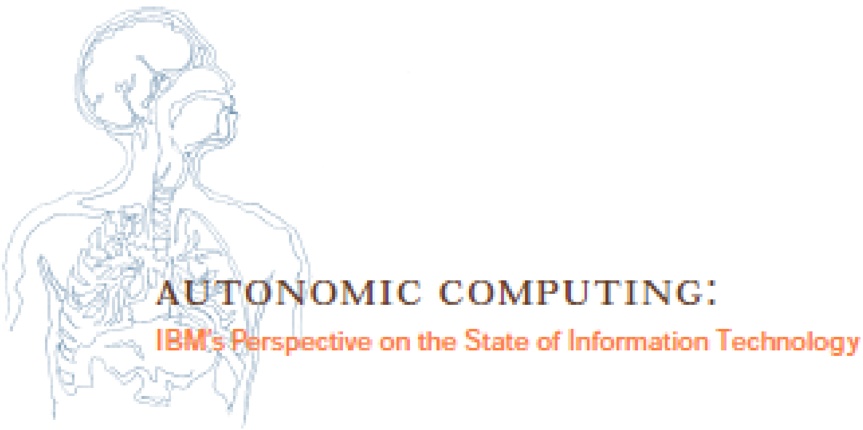
CERES Day

Sept. 14, 2010

Thorsteinn Rögnvaldsson

Paul Horn (then Senior Vice President and Director of Research, IBM), 2001





- Dealing with complexity is the single most important challenge facing the IT industry

- Build computer systems that are like the autonomous nervous system → autonomic computing systems

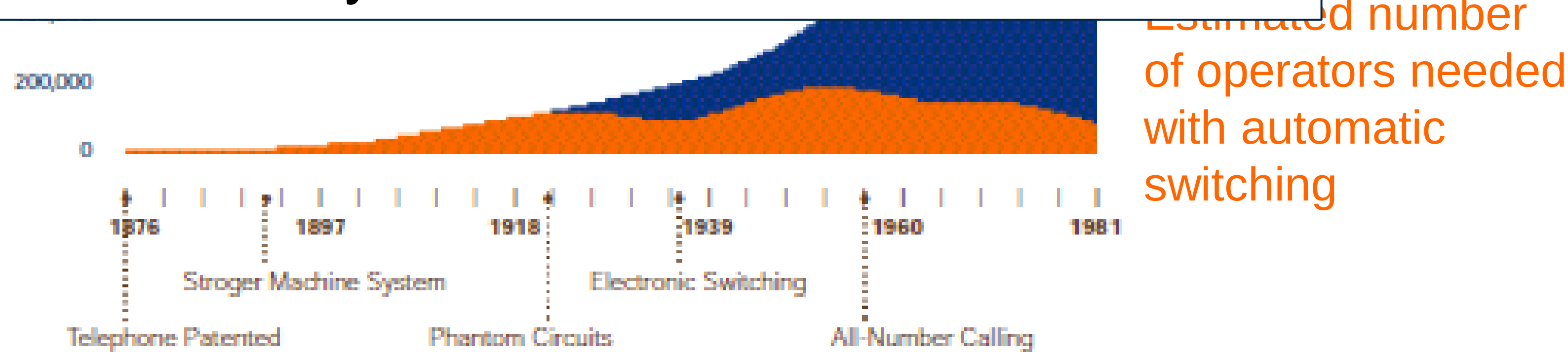
What it we hadn’t had automatic switchboards?

What it we hadn’t had automatic switchboards?

FIGURE 2

The development in computer systems is similar.
By 2020 we’ll need 100 million
IT system administrators in the US…

…if we can’t find better ways to
automate the system administration…



IT'S TIME

to design and build computing systems capable of *running themselves*, adjusting to varying circumstances, and preparing their resources to handle most efficiently the workloads we put upon them. These autonomic systems must anticipate needs and allow users to concentrate on what they want to accomplish rather than figuring how to rig the computing systems to get them there.

From Paul Horn's talk

Self-managing?

- **Self-configuration:** Automated configuration of components and systems.
- **Self-optimization:** Components and systems continually seek opportunities to improve their own performance and efficiency.
  - Self-scheduling, self-tuning, self-planning, self-learning
- **Self-healing:** System automatically detects, diagnoses, and repairs localized software and hardware problems.
  - Self-monitoring
- **Self-protection:** System automatically defends against malicious attacks or cascading failures. It uses early warning to anticipate and prevent failures.

Some early projects/programs

- 1997: DARPA – ”**Situational Awareness System**” (SAS). Sensor readings and communication in battlefields → guaranteed communication. Multihop ad-hoc routing, wide-band transmission .
- 2000: DARPA – ”**Dynamic Assembly for System Adaptability, Dependability and Assurance**” (DASADA). Software that can change itself by swapping or modifying components and/or changing interaction protocols. Use of dynamic gauges that determine run-time composition.
- 2001: IBM – “**Self Aware Distributed Systems**”. Automating real-time problem diagnosis in large-scale distributed systems by using state-of-the-art machine-learning, probabilistic reasoning and information-theoretic approaches. Uses probes.
- 2004: DARPA – “**Self-Regenerative Systems**”. Software is made resistant to attacks by generating a large number of versions with similar functionality.

Today? FP7 ICT WP

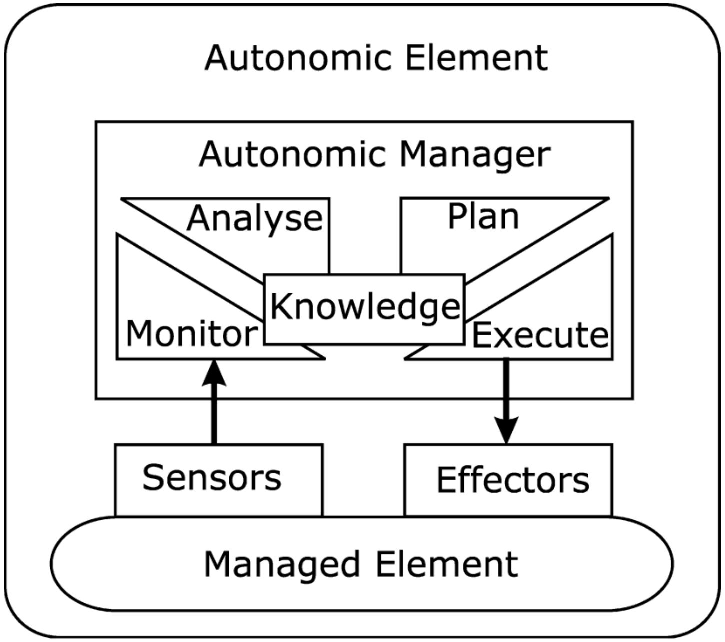
Objective ICT-2009.8.5: FET proactive 5: Self-Awareness in Autonomic Systems (Call Nov 2009)

“The challenge is to create computing and communication systems that are able to optimise overall performance and resource usage in response to changing conditions, adapting to both context (such as user behaviour) and internal changes (such as topology). To achieve this, autonomic systems should enable nodes to build up an awareness relating to higher and even global levels, e.g. of patterns of use, system performance, network conditions and available resources. This requires breaking through the tradition of fixing abstraction layers at design time, which hide issues at lower layers (e.g., by hiding mobility, heterogeneity, or drops in performance), but inevitably limit the scope for optimising resource usage and responding to changing conditions.”

Reference model

- Local software agents for the MAPE-K loop (Monitor-Analyze-Plan-Execute-Knowledge)

- Requires models for system operation…



What’s been done? (2000-2010)

- Work is concentrated on computer networks – typically settings that can be run in simulation (many examples require a high-quality simulator)
    - Computing and memory capacity is assumed high in the nodes
    - Communication bandwidth is assumed high
- Any real-world demonstrators? Few…
- Much of the published work deals with system architecture suggestions

The Backdoor example….

- Use gauges that can access system memory and I/O to monitor system operation
- When a failure is detected, another system takes over the client sessions (the last functioning state of the failed server is injected into a new server).
- Web auctioning system (server park). Recovery time < 25 ms.
- Lots of ”hands on”…

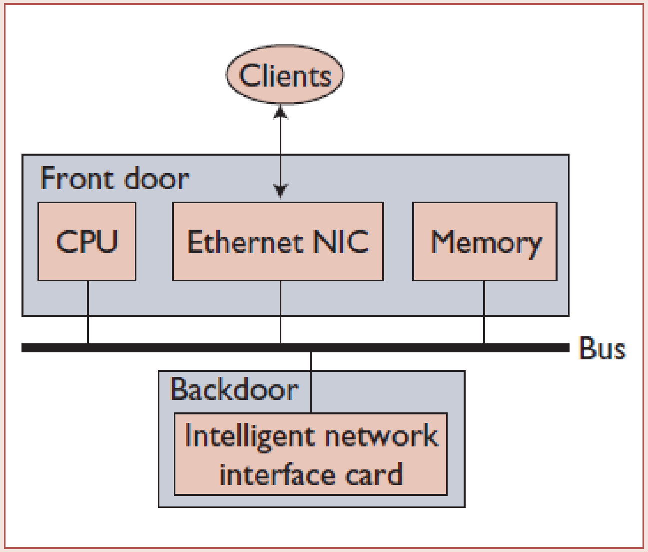
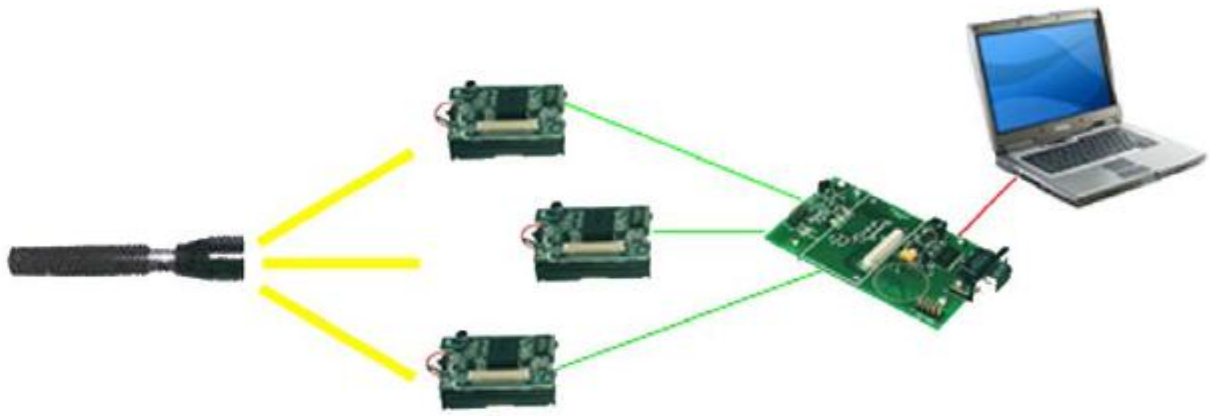


Figure A. System equipped with a backdoor intelligent network interface card. The backdoor I-NIC can access its host system's resources without using the machine's processors.

Wireless sensor network example

- Allow nodes to adjust their communication frequency dynamically…huge drop in energy consumption (250 times less)



- But the setup was hand-tailored for one application (intruder detection with light sensors), no generalization to other settings (applications), no learning of communication functions, no exchange of agents,…
- Utility function…?

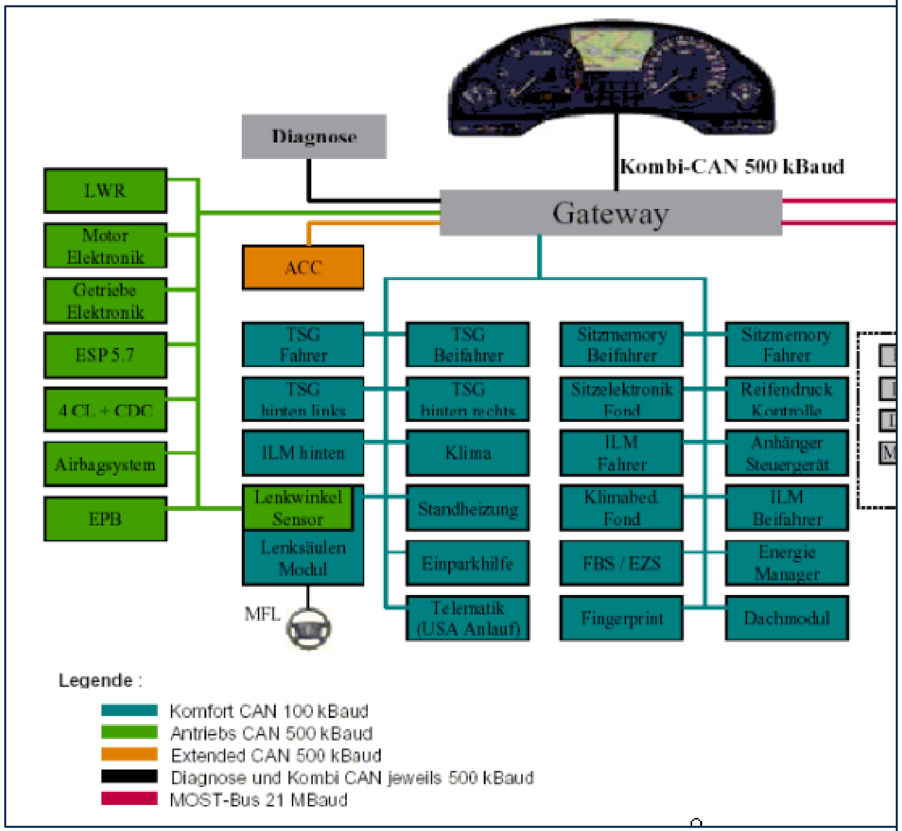
Autonomic computing

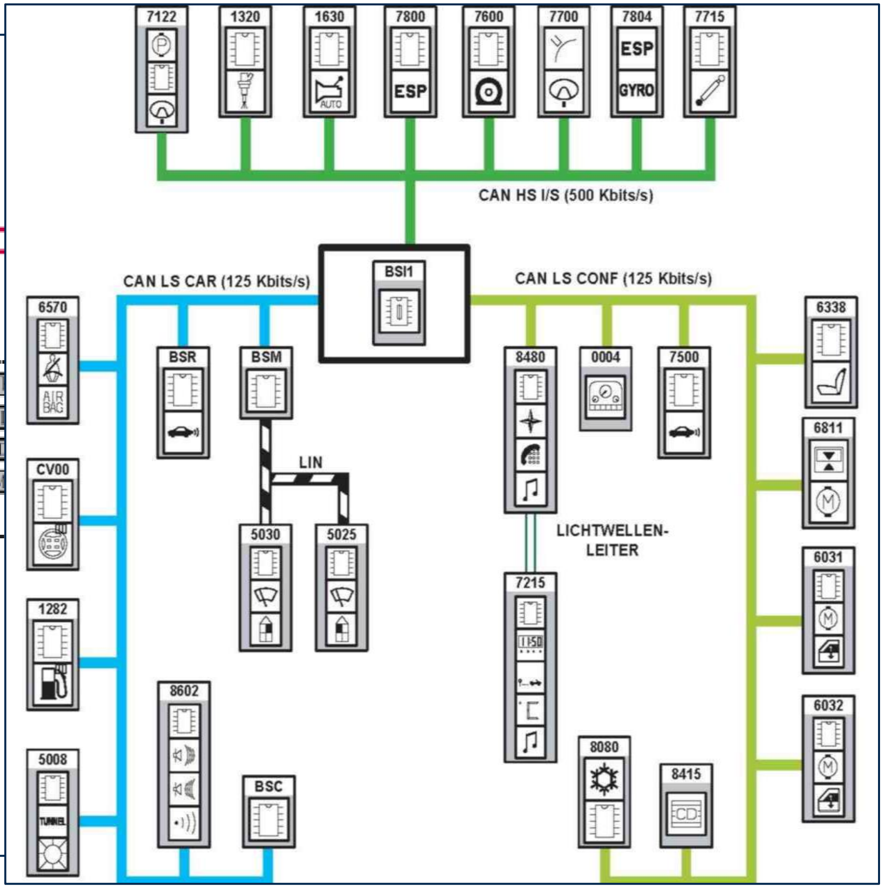
- Still lots left to do (not very much done actually if you look at the ”big picture”)
- But a very interesting (”hype”?) field – with EU funding…

Why am I telling you this?

IMPLICATIONS FOR VEHICLES, SMART GRID, AND…?

Vehicle complexity





Standard diagnostic approaches

- They require <u>prior</u> knowledge for the monitoring part
  - Explicit system model (physical or quantitative)
  - Data-driven model (statistical)
  - Mix of the two above
  - Often poor at modeling fault states

- Lots of human expertize (specifying models)
- Lots of data collection (different usage conditions, different drivers, different ambient conditions, …)

- Expensive to collect all data, build all models, validate all models,
- …and then there is a major system upgrade…

Autonomic computing concepts for vehicles?

- Self-monitoring, self-explaining, self-diagnosing, predicting maintenance?
- Using local agents in the embedded systems on-board the vehicles?

- Joint work with Volvo Technology

VOLVO

Autonomic computing concepts for vehicles?

- Self-monitoring, self-explaining, self-diagnosing, predicting maintenance?
- Using local agents in the embedded systems on-board the vehicles?
- A problem is feature selection on board vehicles when you don’t know the status of the vehicles. What should you capture? (supervised/unsupervised)
- Another problem is the possible lack of important signals (kept inside ECU:s and not available on communication network)

- Joint work with Volvo Technology

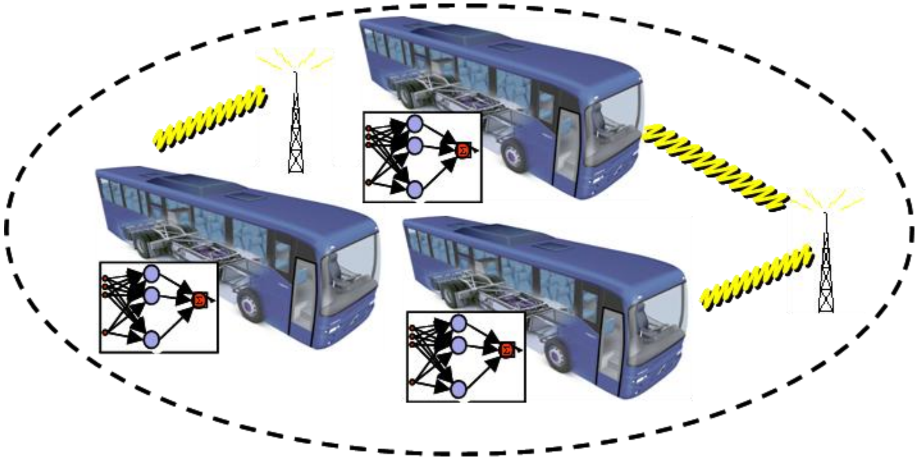
VOLVO

HÖGSKOLAN HALMSTAD

Does it work?

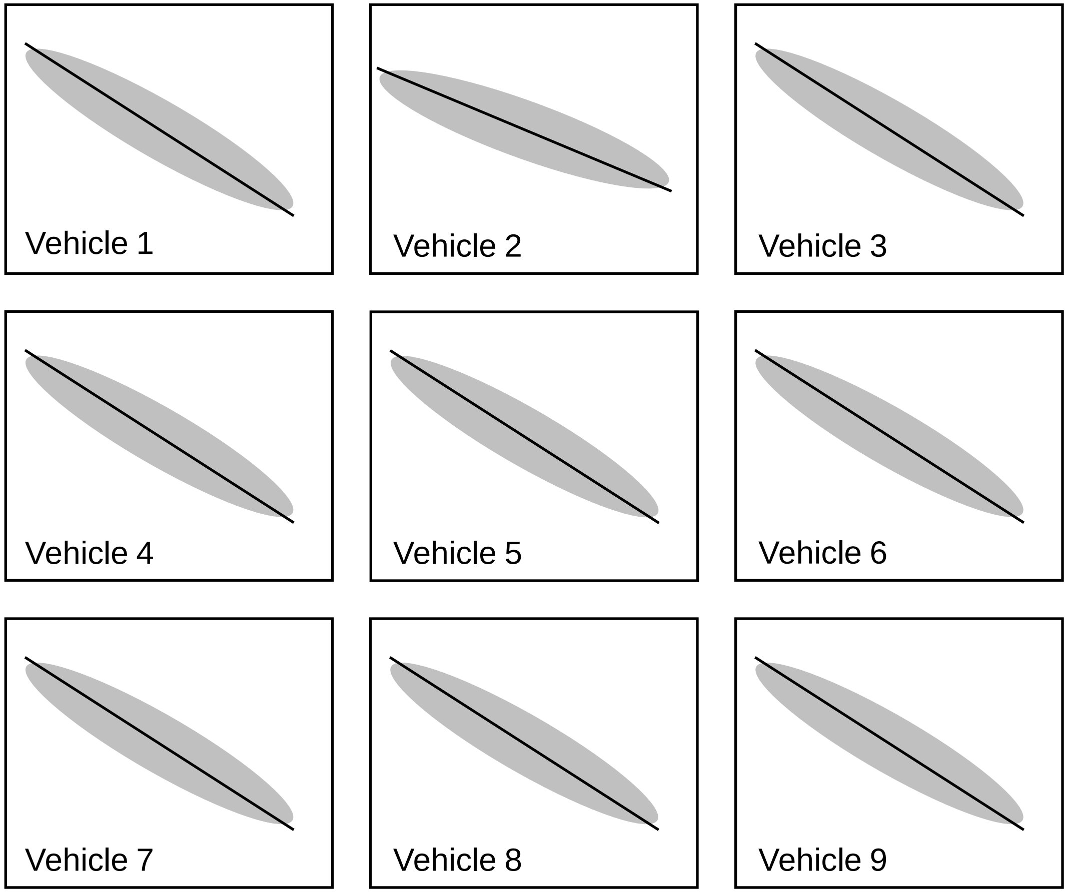
Still work in progress

- The idea is to use groups of vehicles for on-road data collection and definition of ”normal” and ”abnormal”.
- Can detect faults on vehicles, without having been told what the fault looks like or given a prior model of normal operation.

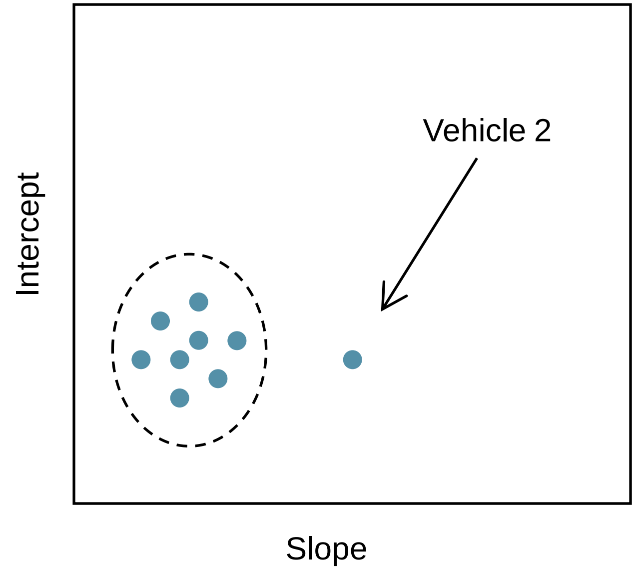


Idea (sketch)

Models are selected based on how ”non-random” they appear.



The x-axis is one sensor and the y-axis another sensor.



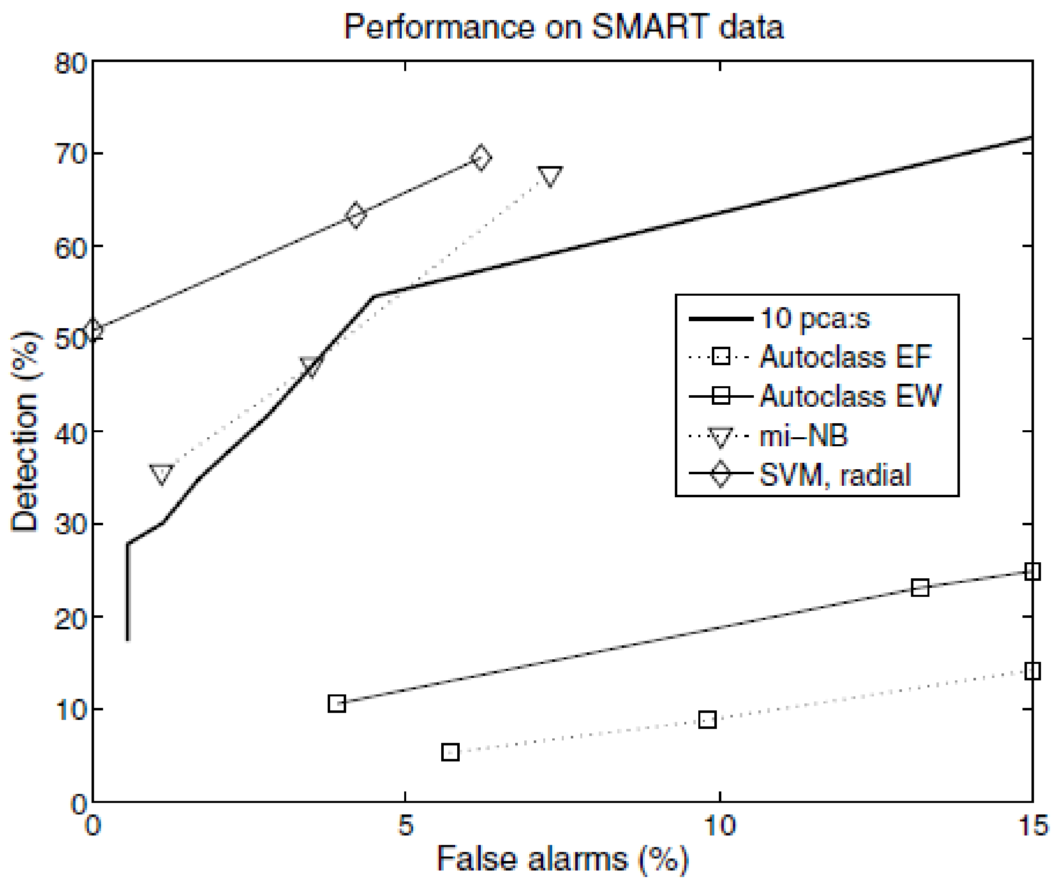
The assumption is that relationships between sensor values are a ”snapshot” of vehicle operation and change with wear and faults

Example: Failing hard-disk drives

- Task: Predict the near-term failure of an individual hard disk drive, and issue a backup warning to its user before data loss from failure occurs.
- Data: time series of “SMART” (Self Monitoring & Reporting Technology) attributes from a single drive model (available from cmrr.ucsd.edu/smart).
- (1) Model signal relationships with PCA and measure subspace distance.
- (2) Model signal relationships with linear models and measure parameter distance.
- “Normal” group is modeled with a bootstrap sample of the population.

Example: Failing hard-disk drives

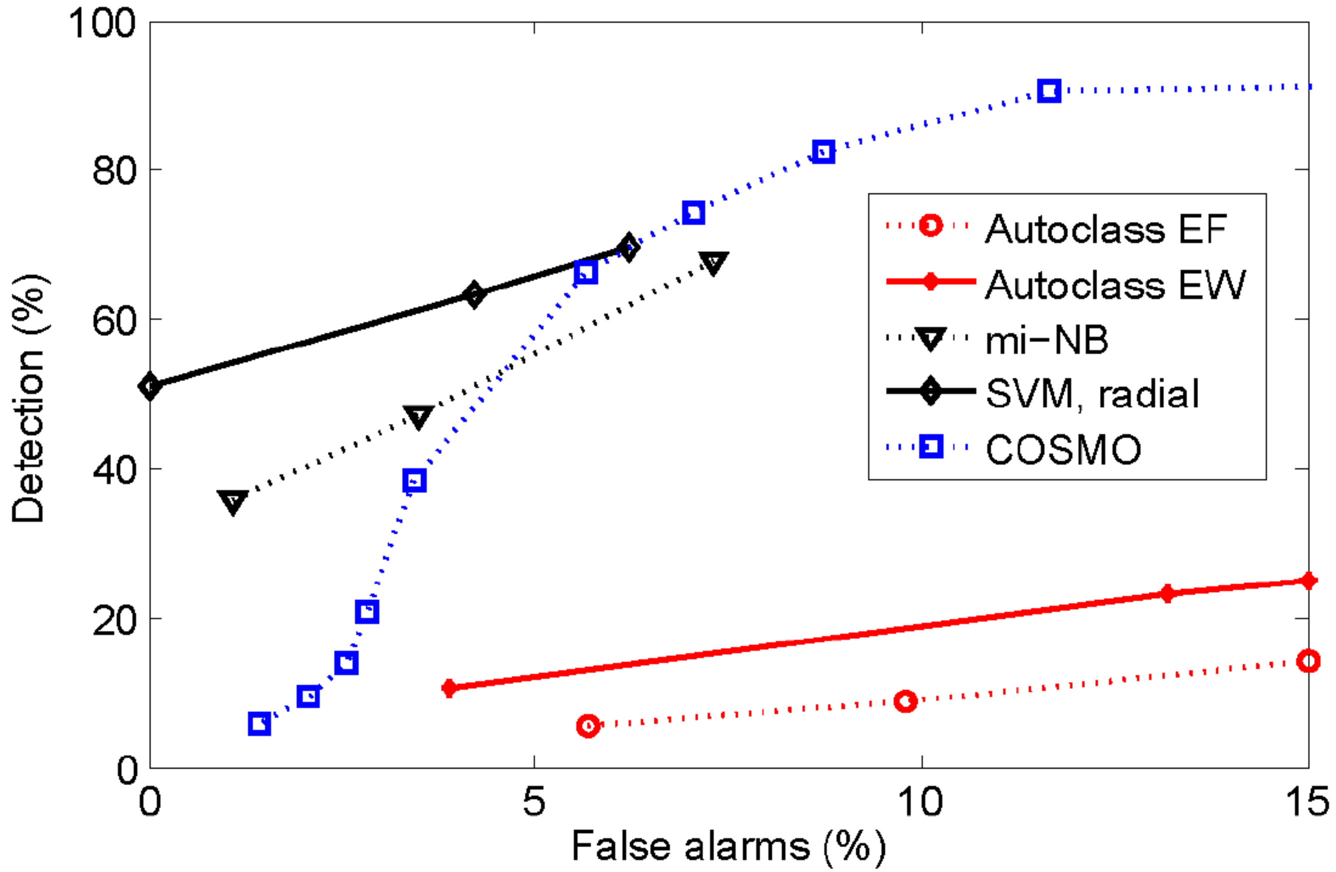
- Task: Predict … and issue a backup warn…
- Data: time ser… …logy) attributes from … smart).
- (1) Model sign… …stance.
- (2) Model sign… …ameter distance.
- "Normal" grou… …on.



The approach performs on par with supervised (Naïve Bayes) approach and significantly better than likelihood density based methods (EM alg.).

Example: Failing hard-disk drives

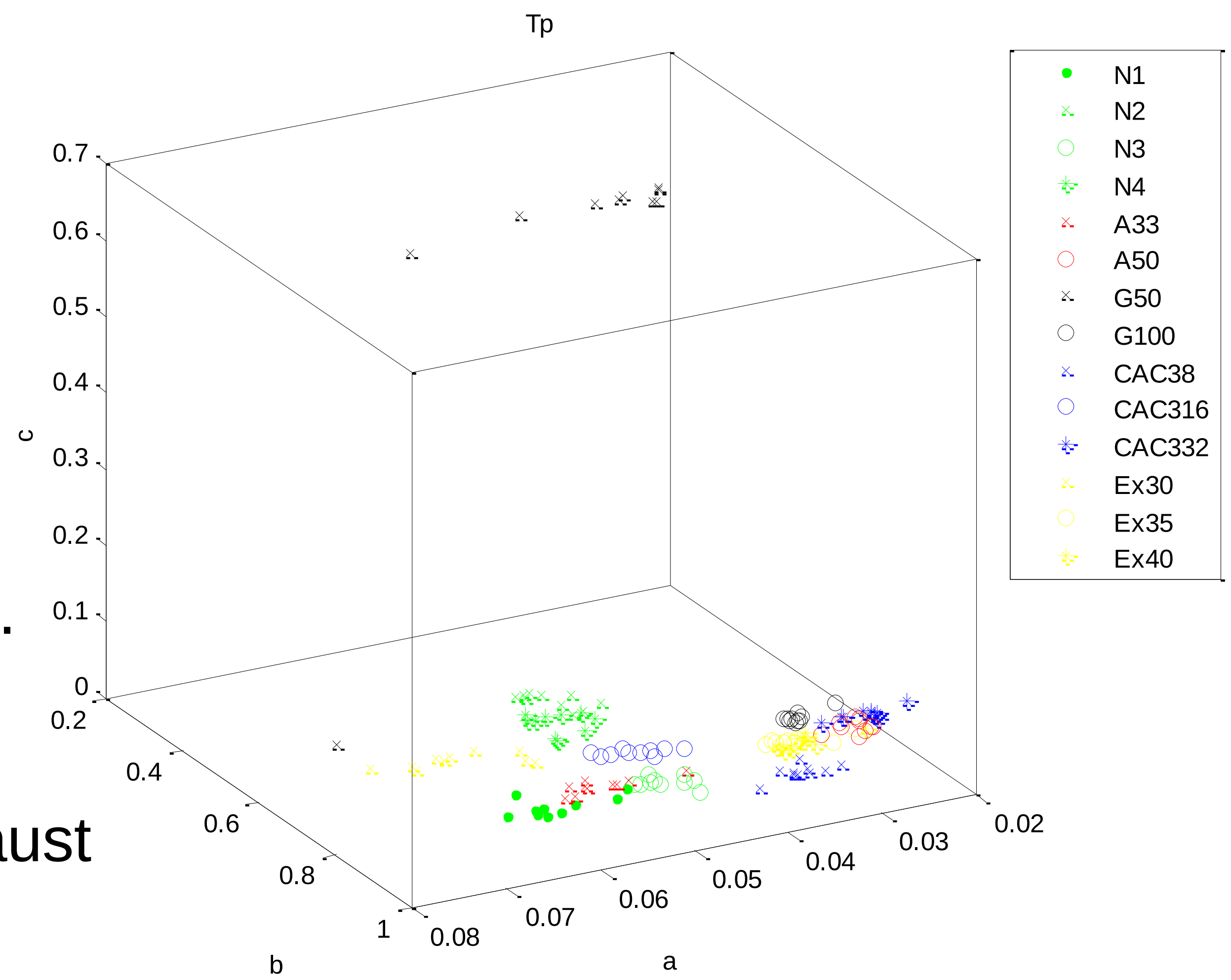
- Task: … ıd issue
  a bac…
- Data: … gy)
  attrib… ıart).
- (1) M… ınce.
- (2) M… ıeter
  distar…
- “Norr… .



Similar to PCA approach (COSMO = Consensus Self-organizing Models)

Heavy duty truck driving on the road. Different faults injected.

Green = No fault
Red = Air filter clogg.
Black = Grill clogg.
Blue = CAC leak
Yellow = Cong. exhaust

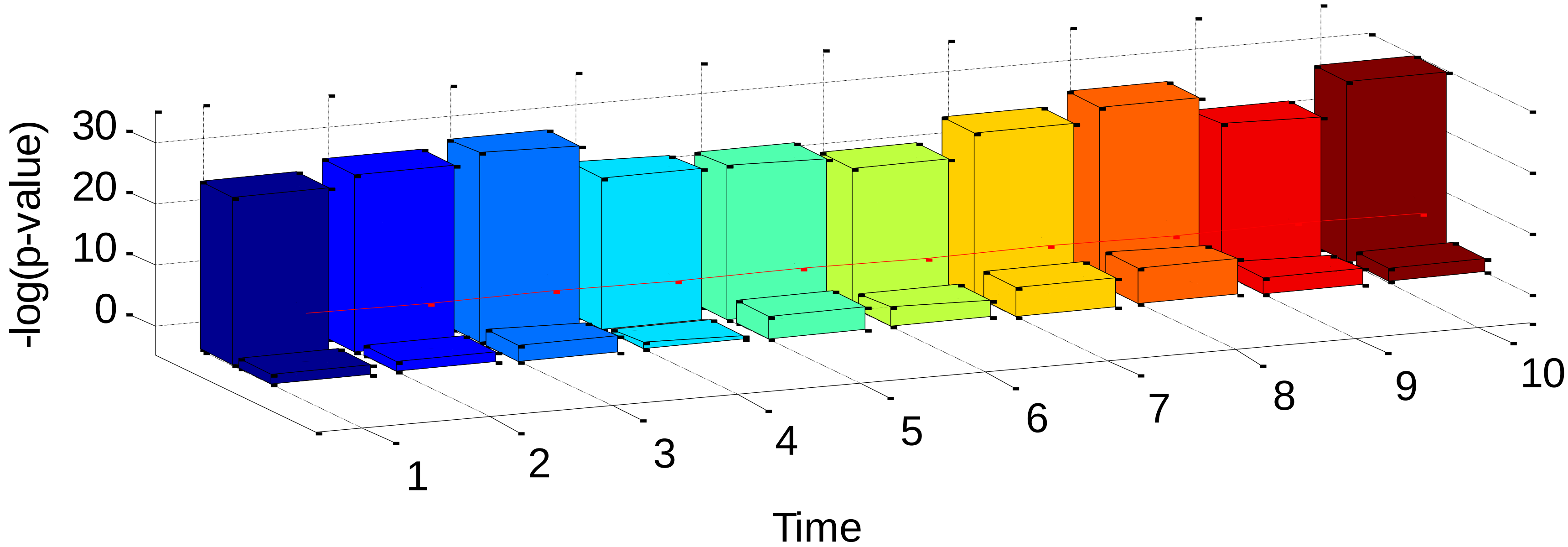


One model; with more models can more faults be identified

Over time (4-5 hours)?

P-value threshold (1e-4) is the red line

The back row is Grill100, front row is Normal



Where are we today

- We (almost) have a ”backdoor” for listening to data traffic on-board vehicles
  - Field test (data collection)
- Exploring how relationships are related to equipment wear and different faults
- Exploring good model structures

- We can detect driver behavior (aggresive, normal, smooth)

Summary

- The autonomic computing paradigm is (in my view) relevant for vehicles
  - Growth in system complexity and need for automated reasoning (self-monitoring, self-reconfiguration,…) – replace the human expert in the lab with an expert system that can poll the systems uder real operation.
  - Need for low-cost (and non-intrusive) systems for monitoring.

- The specific conditions (low communication bandwidth, low on-board computing power, impossible/hard to simulate) set interesting boundaries

- Models (and model parameters) as snapshots of system operation and for comparison seem possible to use