Management Without (Detailed) Models

Alva L. Couch
Mark Burgess
Marc Chiarini

A critical juncture…

- Autonomic computing as conceptualized by many will work if:
  - There are **more precise models**.
  - We can **compose control loops**.
  - Humans will **trust the result**.
- Source: Grand Challenges of Autonomic Computing, HotAC 2008.

Not…!

- Models are already bloated, and some critical model information is **unknowable**.
- The composition problem as posed now is **theoretically impossible** to solve.
- Trust is based upon **simple assurances**.

Most autonomic control solutions

- Assume a **closed world** in which all influences are known.
- Work well in **expected** circumstances.
- React poorly to **unforeseen** situations.
- Example: “**catastrophic**” changes in physical hardware, co-location of services, client load.
- “Learned” data becomes **useless**, must “start over” in learning how system behaves.

In this talk, we…

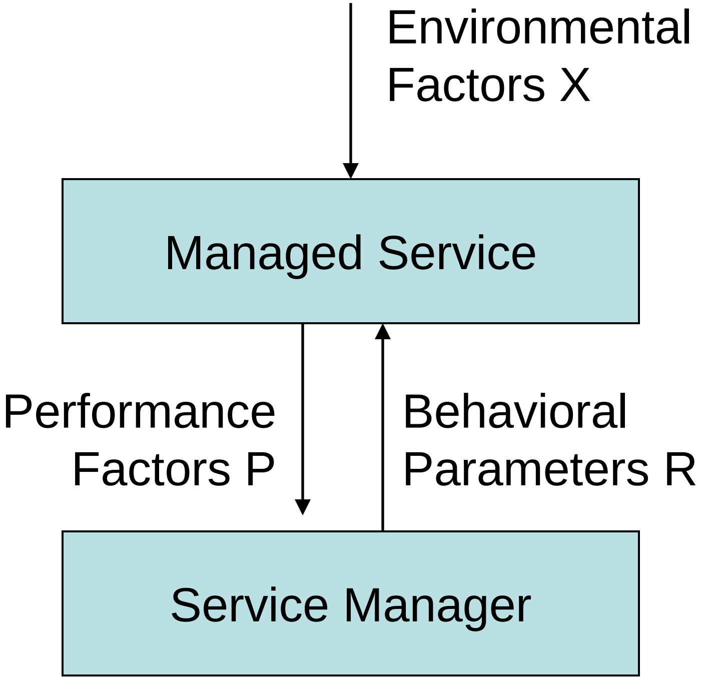
- **Design** for an **open world.**
- **Assume** that behavioral models are **inaccurate** and/or **incomplete.**
- **Mitigate** inaccuracy of models via **constraints** on their inputs and **cautious action**.
- **Exploit** unknown variation to explore possibilities and bound behaviors.

A minimalist strategy

- Consider the **absolute minimum** of information required to control a resource.
- Simplify the control problem to a **cost/value tradeoff**.
- Study “highly adaptive” mechanisms that **maximize reward = value - cost**

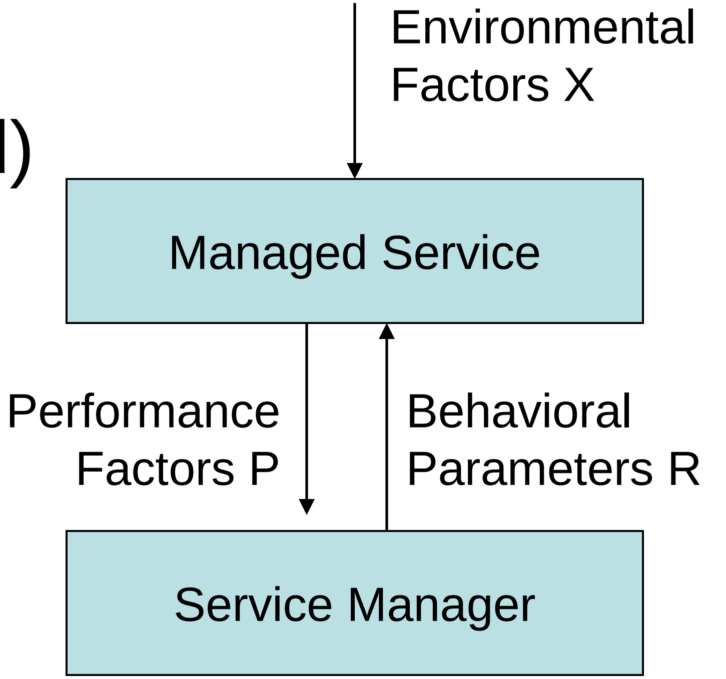
Overall system diagram

- **Resources R**: increasing R improves performance.
- **Environmental factors X** (e.g. service load, co-location, etc).
- **Performance P(R,X)**: throughput changes with resource availability and load.



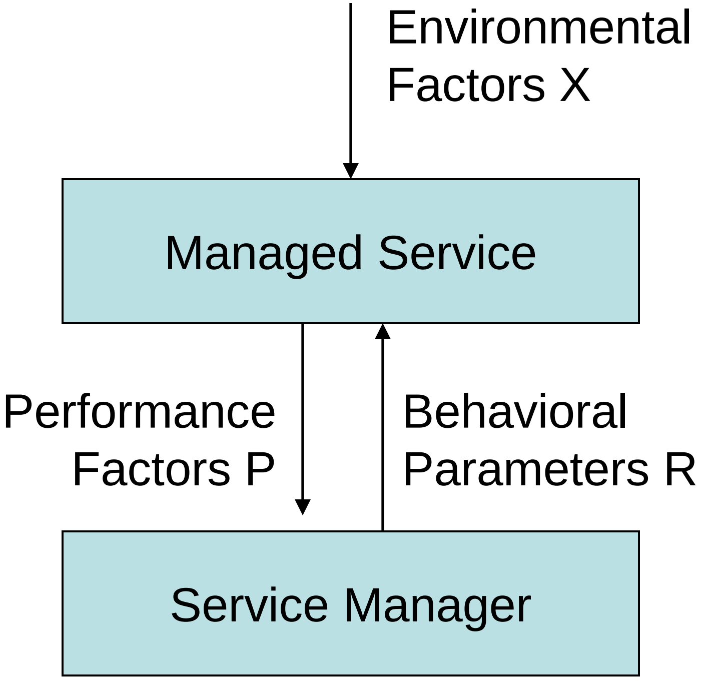
Example: web service in a cloud

- **X** includes input load (e.g., requests/second)
- **P** is throughput.
- **R** is number of assigned servers.



Value and cost

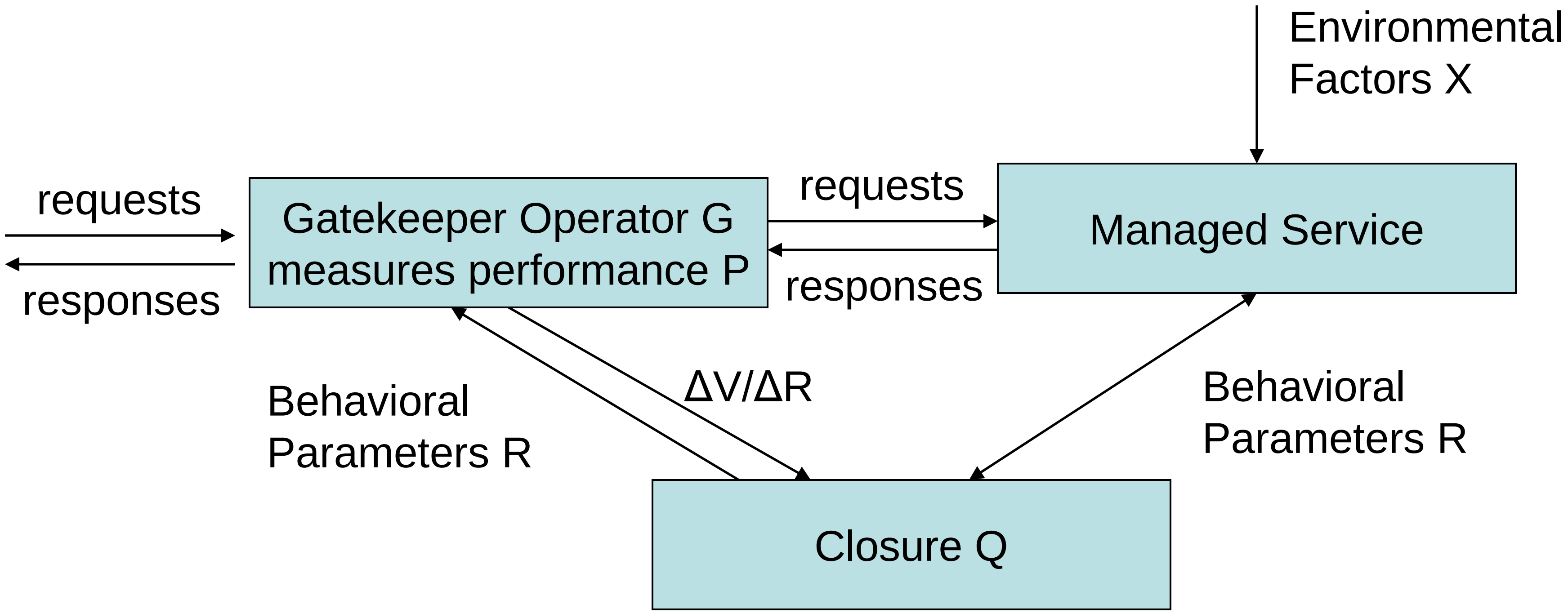
- **Value V(P)**: value of performance P.
- **Cost C(R)**: cost of providing particular resources R.
- Objective function **V(P(R,X))-C(R)**: net reward for service.



Prior paper: last week…!

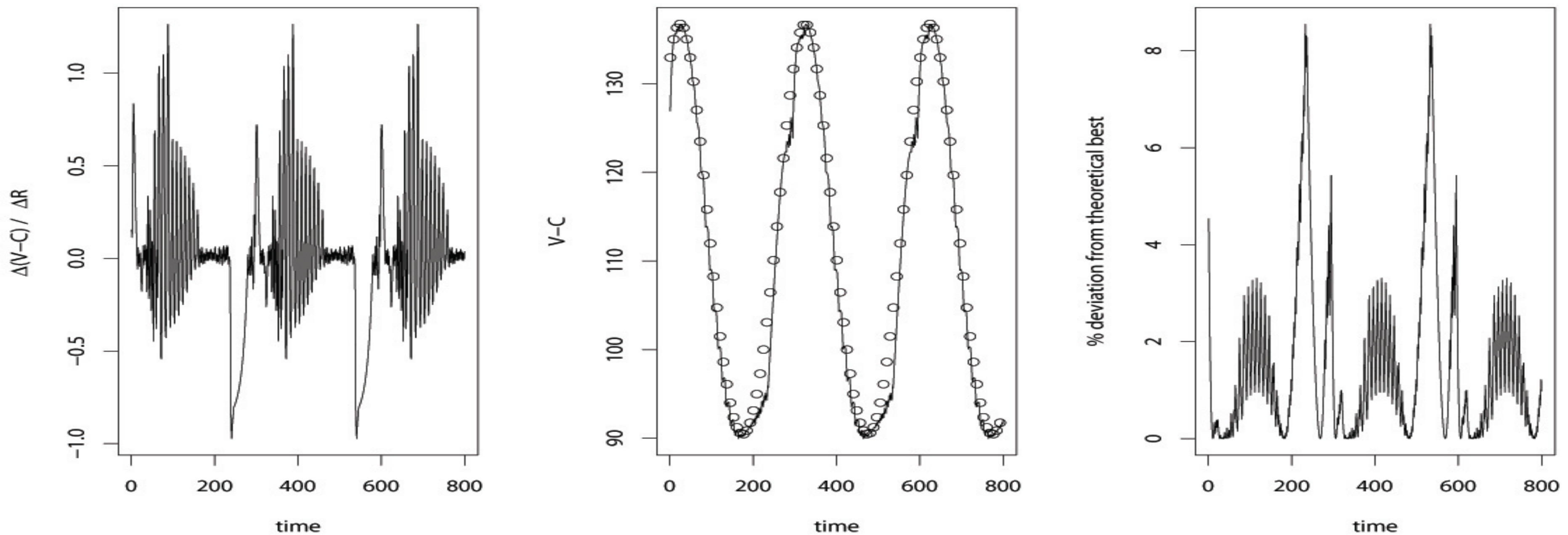
- If P(R,X) is **simply increasing** in R and X, and
- V(P) and C(R) are **simply increasing** in R. and
- V(P)-C(R) is a **convex function**, and
- X changes are **bounded** by sufficiently small $\Delta X/\Delta t$, then
- One can **ignore** X, **estimate** P(R), and **maximize** V(P(R))-C(R) by **incremental hill climbing**.
- Couch and Chiarini, "Dynamics of resource closure operators", *Proc. AIMS 2009,* Twente, The Netherlands.

Brief overview of AIMS paper



- G knows V(P), predicts **changes in value** $\Delta V/\Delta R$.
- Q knows C(R), computes $\Delta(V\text{-}C)/\Delta R$, chooses appropriate **sign** for increment ΔR.

A simulation of the method

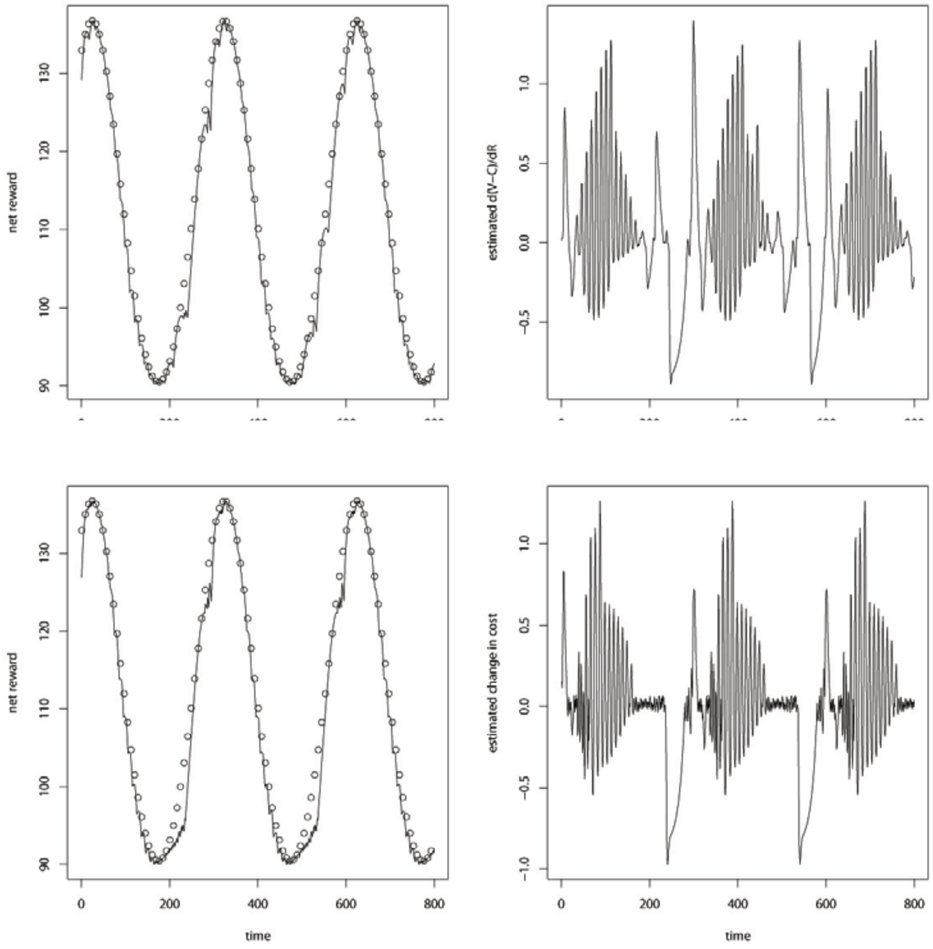


- Δ(V-C)/ΔR is seemingly random (left).
- V-C closely follows theoretical ideal (middle).
- Percent differences from ideal are small (right).

This is not machine learning

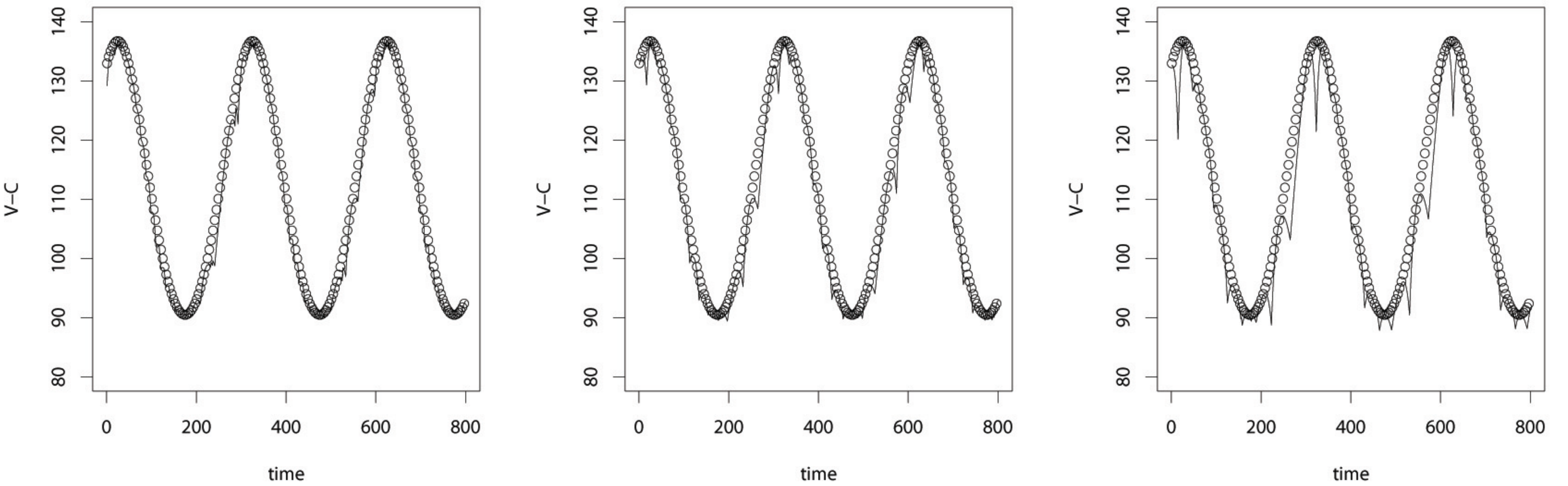
- Accuracy of the model for P(R) is **not critical.**
- Algorithm behavior improves when **less** history is used.

Model is not critical



- Top run approximates V as aR+b so that ΔV/ΔR≈a,
- Bottom run fits V to more accurate model a/R+b.
- Accuracy of G’s estimator is **not critical**, because estimation errors from unseen changes in X dominate errors in the estimator!

History: 10,20,30 steps

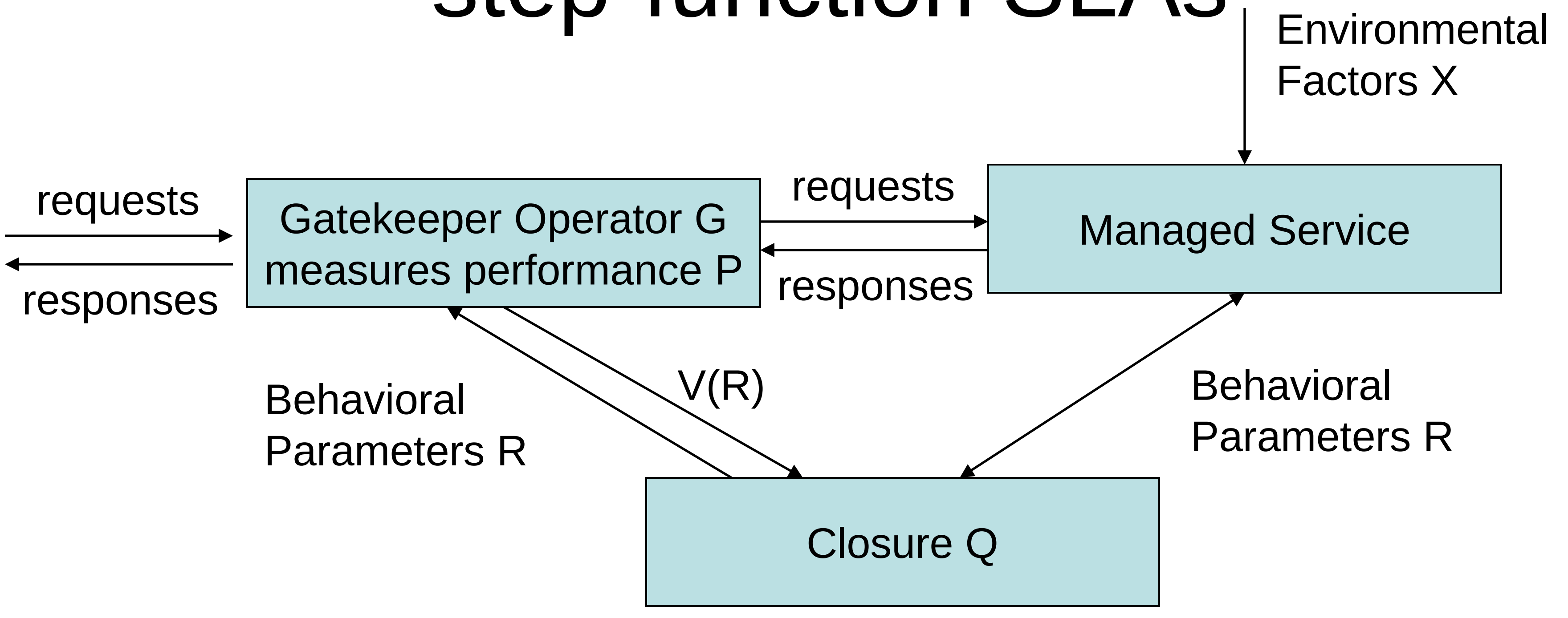


- Solid curve is **simulated** behavior,
- Circles represent **optimal** behavior.
- Using more history **magnifies prior errors.**

Limitations

- Preceding only works if functions V, C, P are never constant on an interval.
- What if the functions V, C are step functions (as in a Service-Level Agreement (SLA))?

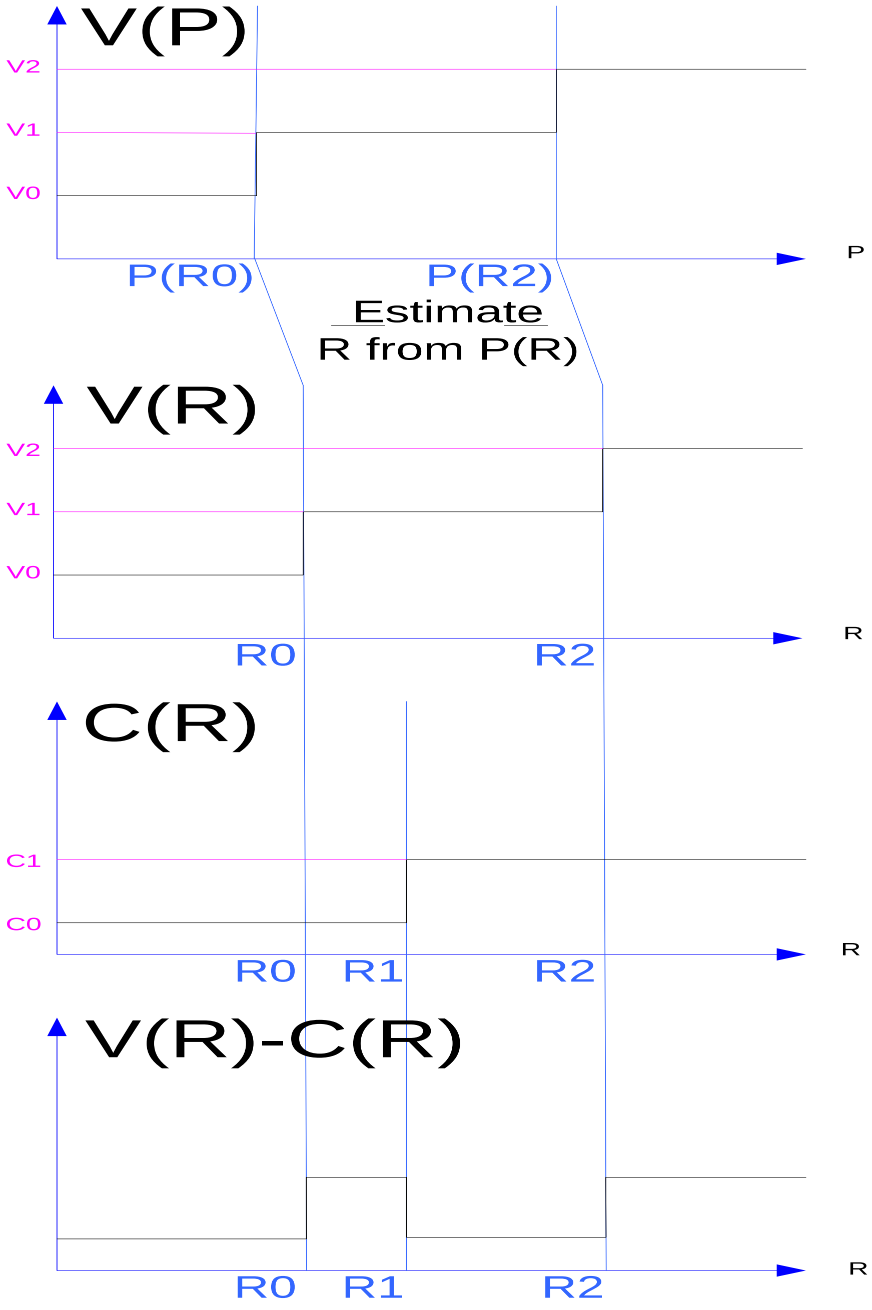
Back to this paper: step-function SLAs



- Distributed agent G knows V(P), R; predicts **value** V(R).
- Q knows C(R), maximizes V(R)-C(R) by incrementally changing R.
- V(R) and C(R) are step functions, i.e., tables of keys and values.

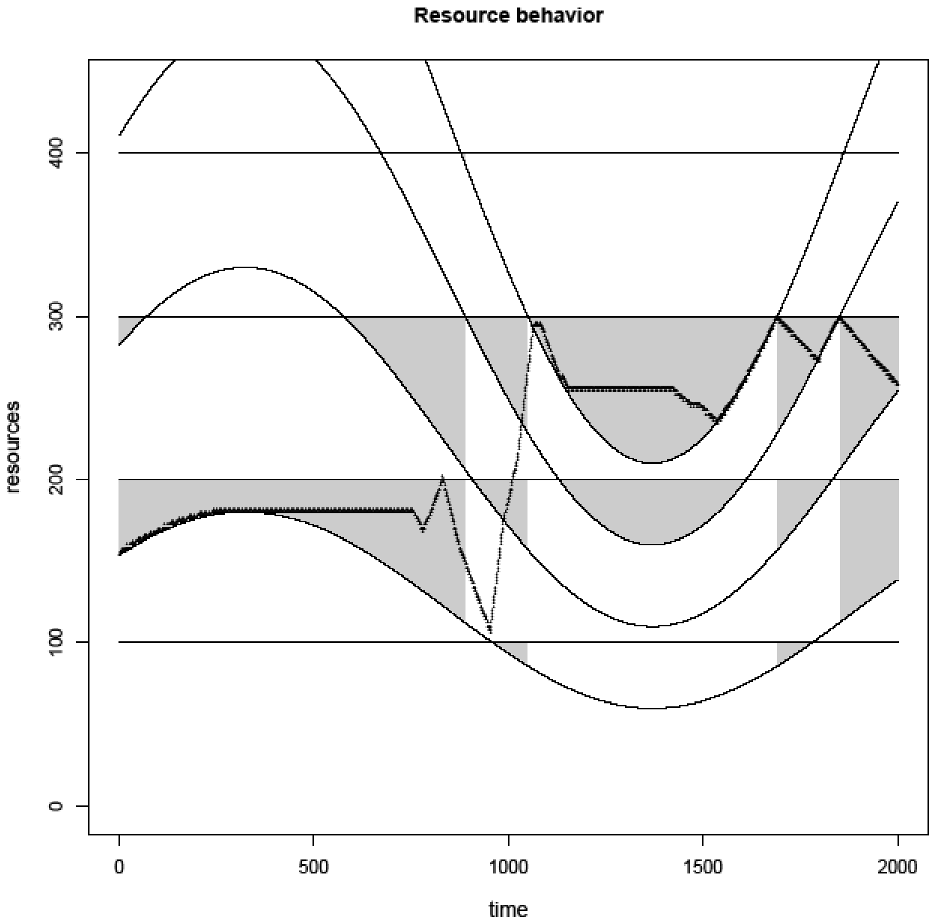
Estimating V-C

- Estimate R from P.
- Estimate V(R) from V(P).
- Subtract C(R).
- Levels V0, V1, V2, C0, C1 and cutoff R1 **do not change**.
- R0, R2 **change over time** as X and P(R) change.



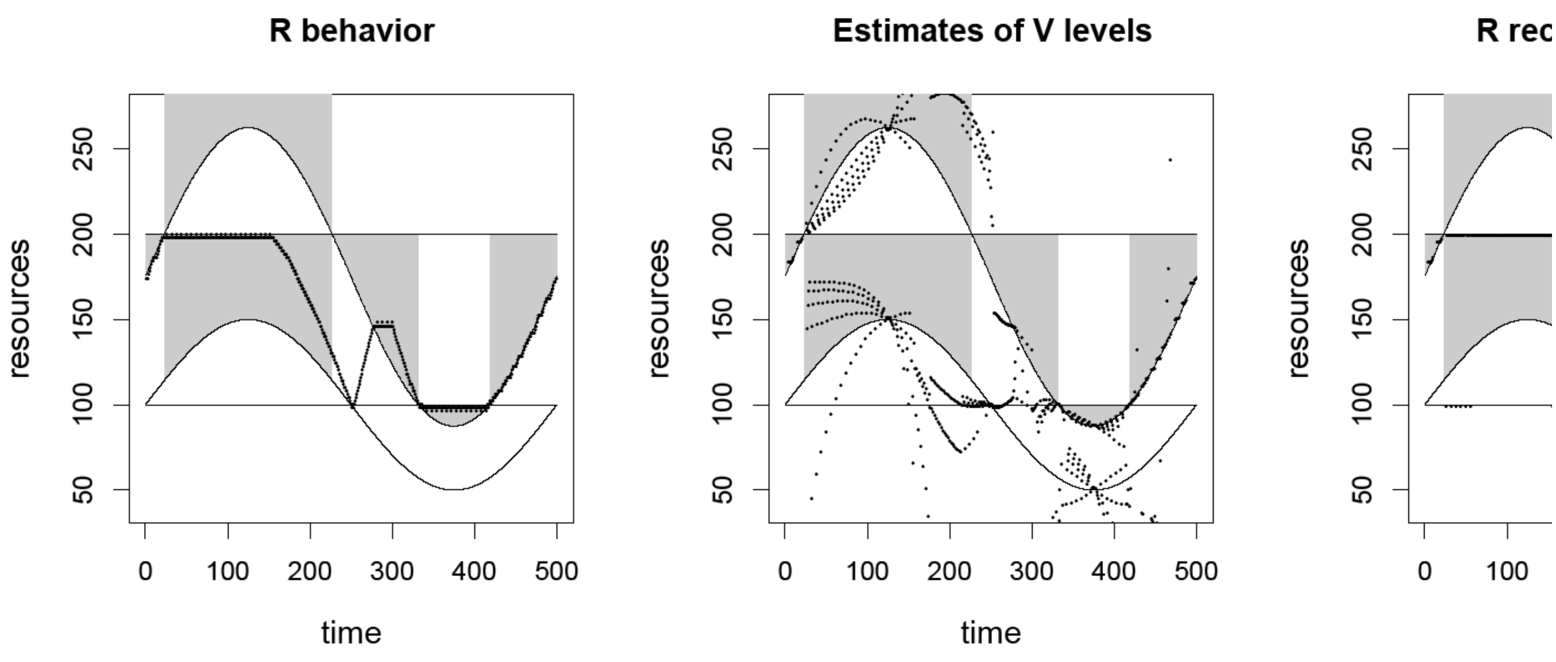
Level curve diagrams

- Horizontal lines represent (constant) **cost cutoffs**.
- Wavy lines represent (varying) theoretical **value cutoffs**.
- Best V-C only changes at times where **a value cutoff crosses a cost cutoff.**
- Regions between lines and between crossovers represent **constant V-C.**
- Shaded regions are areas of **maximum V-C.**



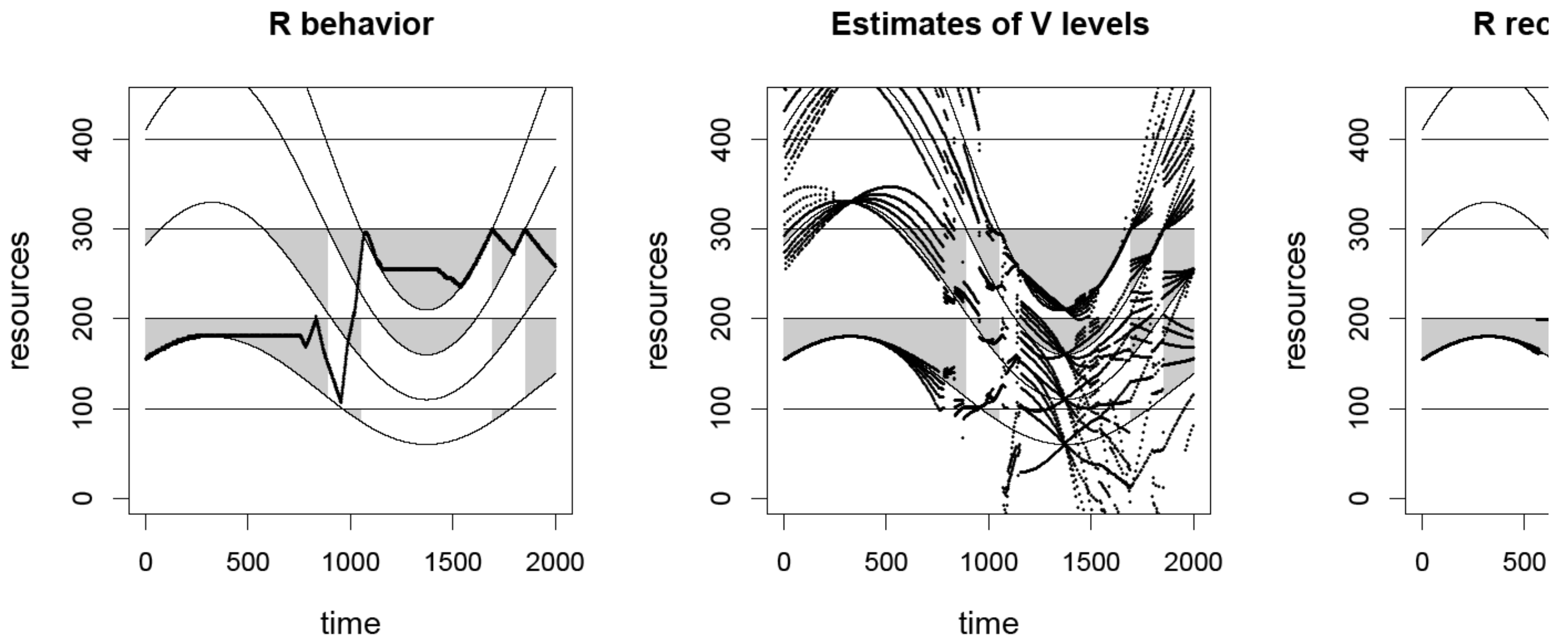
Estimating nearest-neighbor value cutoffs

- Estimate the **two steps** of V(R) around the current R.
- Fitted model for P(R) is **not critical**.
- **V-C must be convex in R.**



Estimating all value cutoffs

- Accuracy of P(R) estimate **decreases with distance** from current R value.
- Choice of model for P(R) is **critical.**
- V-C **need not be convex in R.**



In other words,

- One can make tradeoffs between convexity and accuracy!

How well does this do?

- In a realistic situation, we don’t know optimum values for R.
- Must estimate ideal behavior.
- Method: exploit X variation.

Observed efficiency (a simplified description)

- Consider n time steps i=1,n.
  - Let N_i be the **observed** V_i-C_i at step i. Let $N = \sum N_i$
  - Let T_i be the **theoretical best** V_i-C_i at step i. Let $T = \sum T_i$
  - Let M_i be the **maximum estimated** V_i-C_i at step i.
  - Let $M = n \cdot \max(M_i)$.
- Call N/T the **efficiency** of the process for n steps.
- Call N/M the **observed efficiency** of the process.
- Over a large enough sample n, where X varies, $M \geq T$ and $N/M \leq N/T$.
- Thus **observed efficiency** N/M is a **lower bound** on efficiency.

How accurate is the estimate?

- Three-value simulation.
- Sinusoidal load.
- More details and results in paper.

loadPeriod	optimum	observed	difference
100	0.800000	0.618421	0.181579
200	0.565310	0.453608	0.111702
300	0.751067	0.647853	0.103214
400	0.896478	0.760870	0.135609
500	0.826939	0.728775	0.098164
600	0.857651	0.760732	0.096919
700	0.946243	0.845524	0.100719
800	0.893867	0.807322	0.086545

Some caveats

- In some simulations, M could not be estimated.
  - Too many situations in which V could not be estimated.
  - Insufficient grounds for interpolating.
- In very rare cases, M is slightly > T.
  - Sample too small to predict maximum.
  - Not enough variation in input load.

In this talk, we…

- **Designed** for an open world.
- **Assumed** that behavioral models are **inaccurate** and/or **incomplete**.
- **Mitigated** inaccuracy of models via **constraints** on input and **cautious action**.
- **Exploited** unknown variation to explore possibilities.

But…

- This is an extreme case.
- Step functions are better handled by non-incremental means.
- There are many algorithms between the extremes of model-based and model-free control.
- We can model X and P(R,X) and still obtain these benefits…
- … provided that we are willing to stop using models that become observably incorrect over time!
- More about this in the next installment (MACE 2009)!

Questions?
Management Without (Detailed) Models

Alva L. Couch

Mark Burgess

Marc Chiarini