Dynamics of resource closure operators

Dr. Alva L. Couch
Marc Chiarini
Tufts University

Outline of this talk

- **Violate** many of the “mores” of autonomic computing.
- **Demonstrate** that one can get away with this.
- **Duck!**

A critical juncture…

- Autonomic computing as conceptualized now will work if:
  - There are better models.
  - We can compose several control loops with predictable results.
  - Humans will trust the result.
- Source: Hot Autonomic Computing 2008: Grand Challenges of Autonomic Computing.

Not…!

- Models are already bloated, and some critical information is **unknowable**.
- The composition problem as posed now is **theoretically impossible** to solve.
- Trust is based upon **simple assurances** that many current systems cannot make.

Inspiration: computer immunology

- Burgess: we can manage systems via independently acting **immunological operators**.
- Autonomic computing can be **approximated** by these operators (Burgess and Couch, 2006).

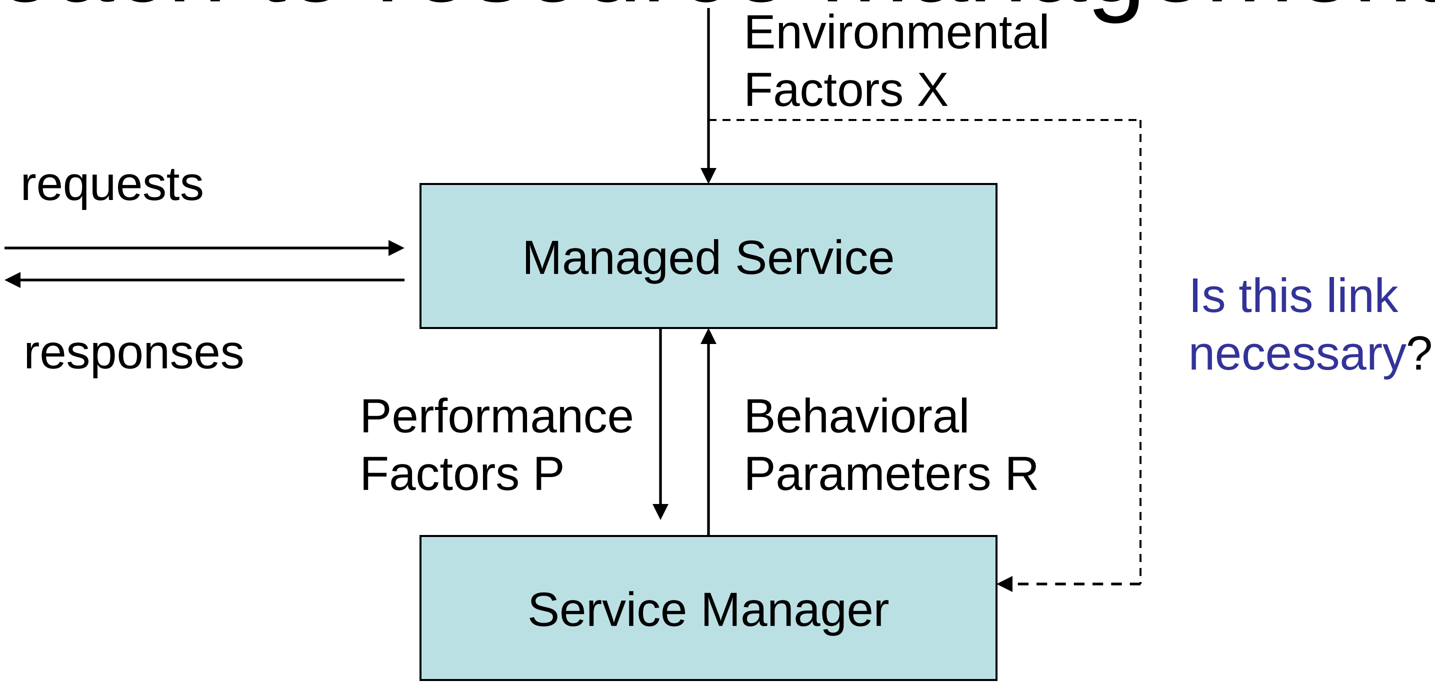
Open-world and closed-world assumptions

- IBM’s blueprint for autonomic computing is based upon a **closed-world assumption:** one can learn everything about a system.
- Burgess’ immunology is based upon an **open-world assumption:** some system attributes are unknowable.

A minimalist approach

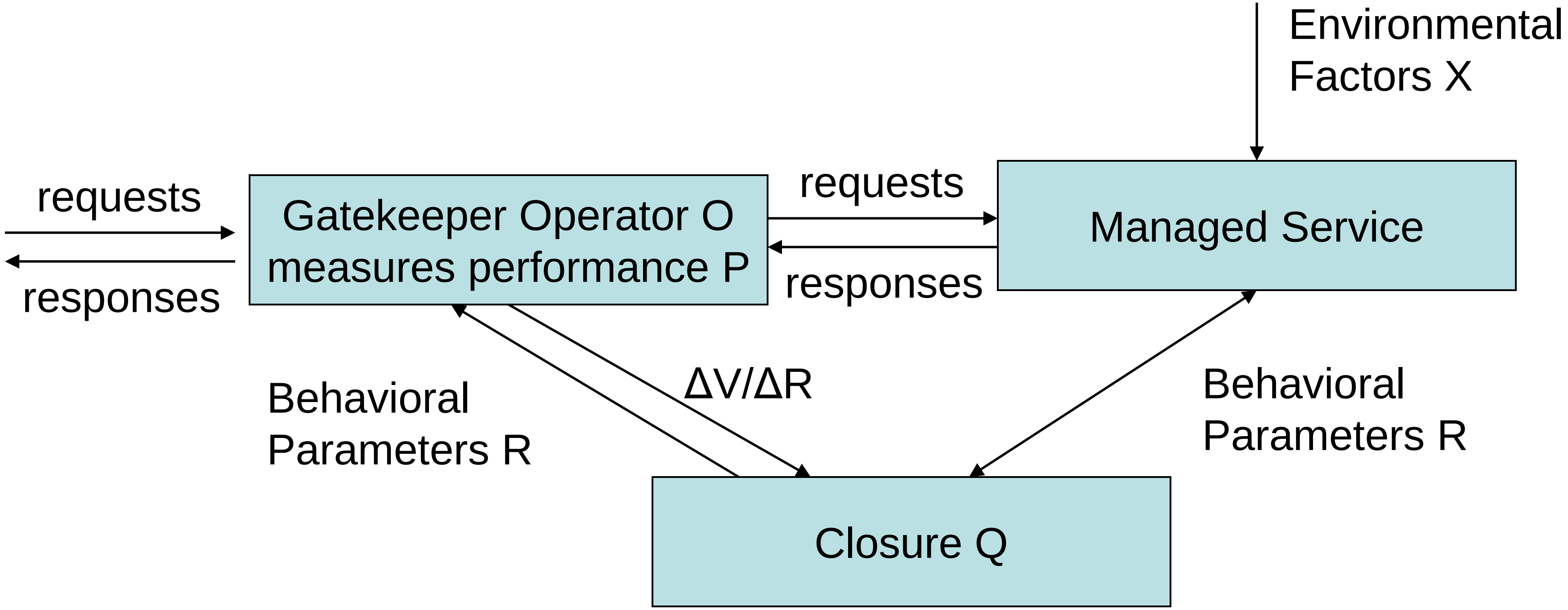
- Consider the **absolute minimum** of information required to control a resource.
- Formulate control as a **cost/value tradeoff**.
- Operate in an **open world.**
- Study mechanisms that maximize **reward = value-cost**.
- **Avoid modeling** whenever possible.

Traditional control-theoretic approach to resource management



- Develop a **model** of P(R,X) and a **model** of X.
- **Predict** changes in P due to changes in R.
- Weigh **value** V(P) of P against **cost** C(R) of R.

Our approach



- **Immunize R** based upon partial information about P(R,X).
- Distributed agent O knows V(P), predicts **changes in value** ΔV/ΔR.
- Closure Q knows C(R), weighs ΔV/ΔR against the change in cost ΔC/ΔR, and increments or decrements R.

Key differences from traditional control model

- Knowledge is **distributed**.
  - Q knows **cost but not value**
  - O knows **value but not cost**.
  - There can be multiple, distinct concepts of value.
- We **do not model P or X** at all.

A simple simulation

- We tested this architecture via simulation.
- Environment X = sinusoidal load function (between 1000 and 2000 requests/second).
- Resource R = number of servers assigned.
- Performance (response time) P = X/R.
- Value V(P) = 200-P
- Cost C(R) = R
- Objective: maximize V-C, subject to $1 \leq R \leq 1000$
- Theoretically, objective is achieved when $R=X^{1/2}$

Some really counter-intuitive results

- Q sometimes **guesses wrong,** and is only **statistically correct**.
- Nonetheless, Q can keep V-C **within 5% of the theoretical optimum** if tuned properly, while remaining highly adaptive to changes in X.

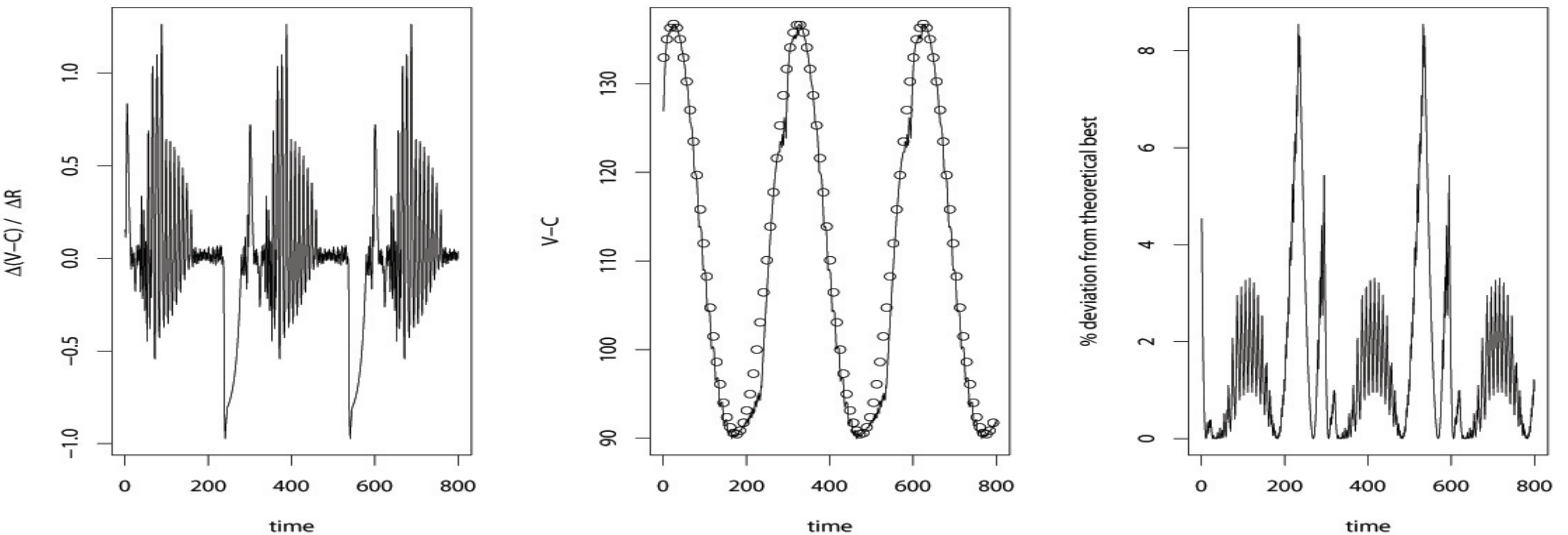
Parameters of the system

- Increment **ΔR**: the amount by which R is incremented or decremented.
- Window **w**: the number of measurements utilized in estimating ΔV/ΔR.
- Noise **σ**: the amount of noise in the measurements of performance P.

Tuning the system

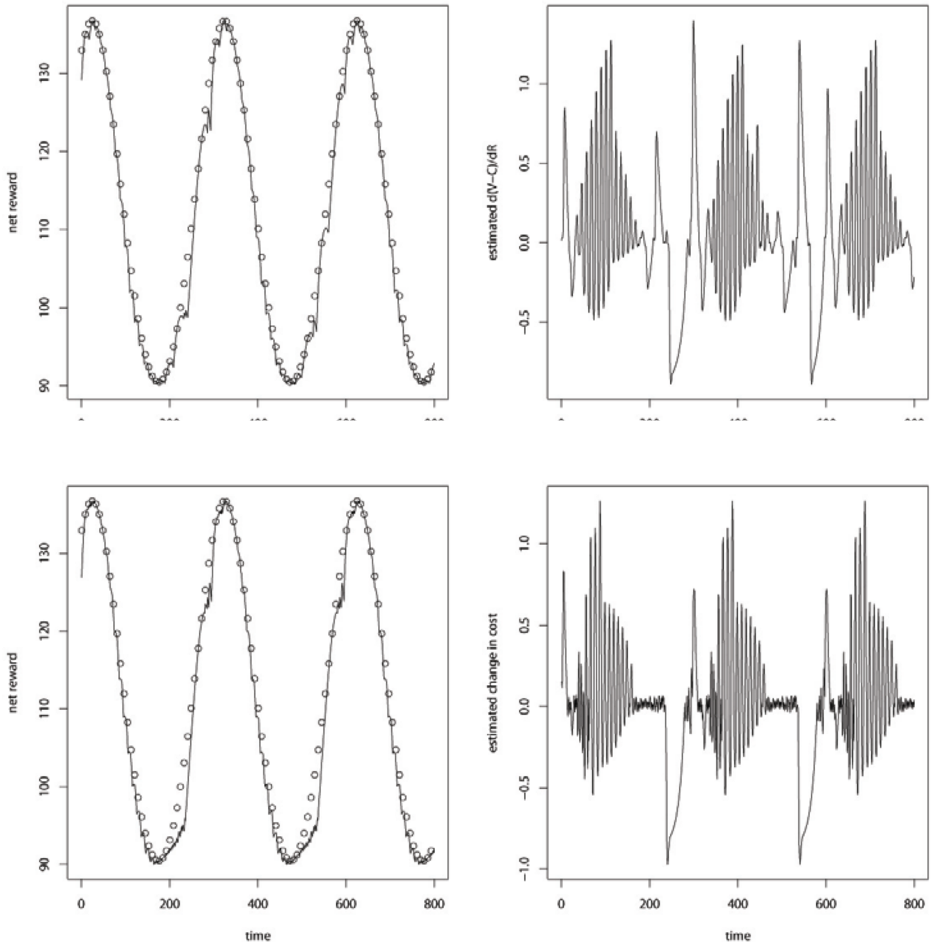
- The accuracy of the estimator that O uses is **not critical.**
- The window w that O uses is **not critical,** **(**but larger windows **magnify** estimation errors!)
- The increment ΔR that Q uses is a **critical parameter** that affects how closely the ideal is tracked.
- **This is not machine learning!!!**

A typical run of the simulator



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

Model is not critical

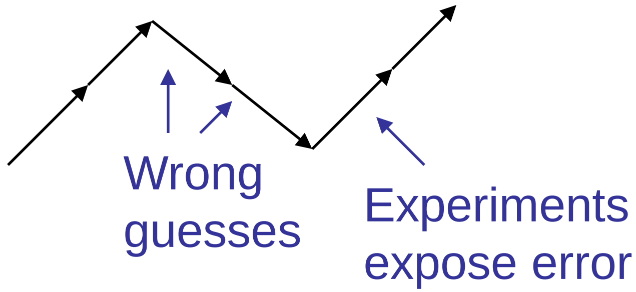


- Top run fits V=aR+b so that ΔV/ΔR≈a, bottom run fits to more accurate model V=a/R+b.
- Accuracy of O's estimator is **not critical**, because estimation errors from unseen changes in X dominate errors in the estimator!

Why Q guesses wrong

- We don’t model or account for X, which is changing.
- Changes in X cause **mistakes in estimating ΔV/ΔR**, e.g., load goes up and it appears that value is going down with increasing R.
- These mistakes are **quickly corrected**, though, because when Q acts incorrectly, it gets almost instant feedback on its mistakes from O.

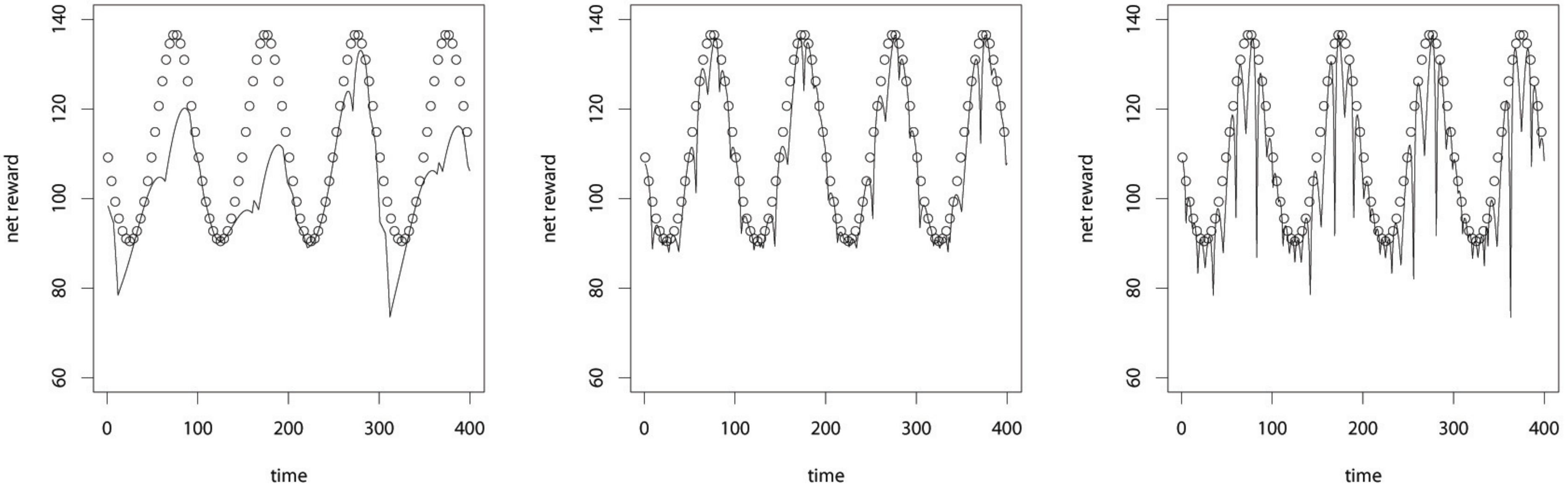
Error due to increasing load is corrected quickly



A brief tour of results

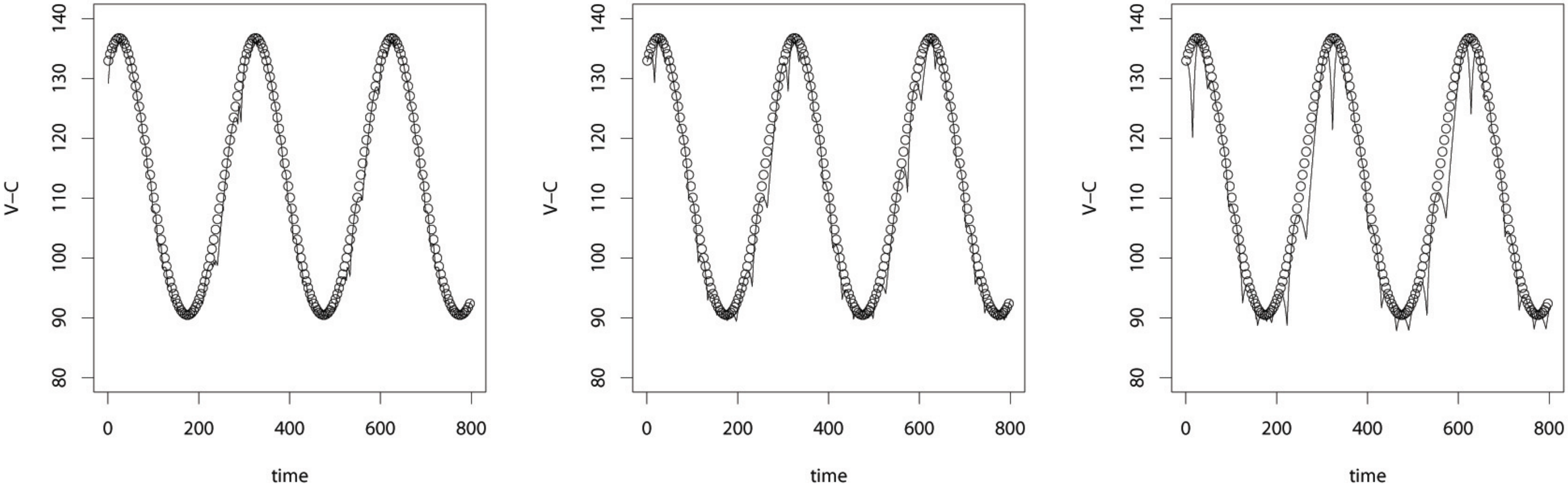
- Effect of ΔR = Q’s increment for R.
- Effect of w = window size for estimator.
- Effect of Gaussian noise in X signal.

Increment ΔR=1,3,5



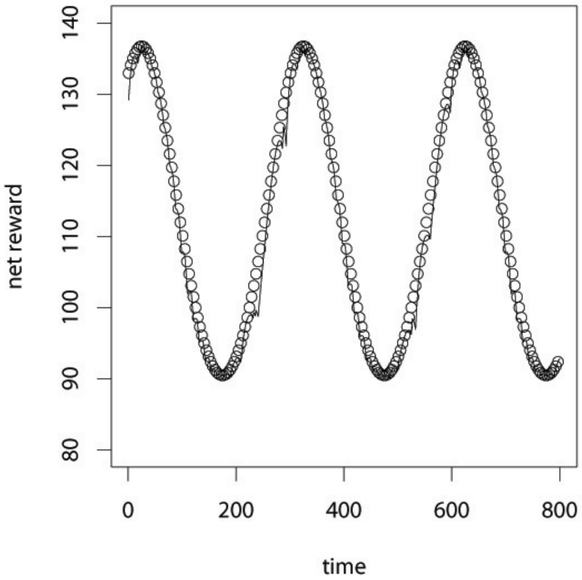
- Plot of time versus V-C.
- ΔR too small leads to undershoot.
- ΔR too large leads to overshoot and instability.

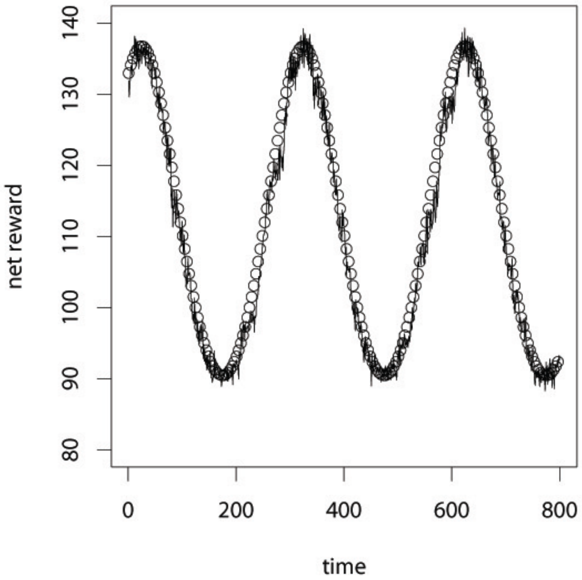
Window w=10,20,30

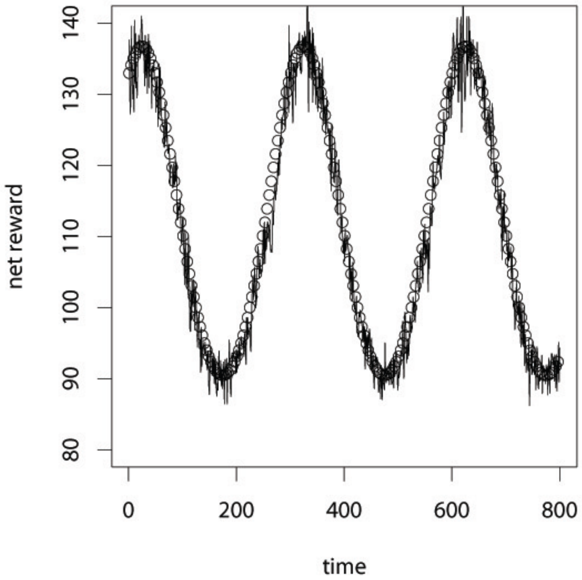


- Plot of time versus V-C.
- Increases in w **magnify errors in judgment** and decrease tracking.

0%, 2.5%, 5% Gaussian Noise

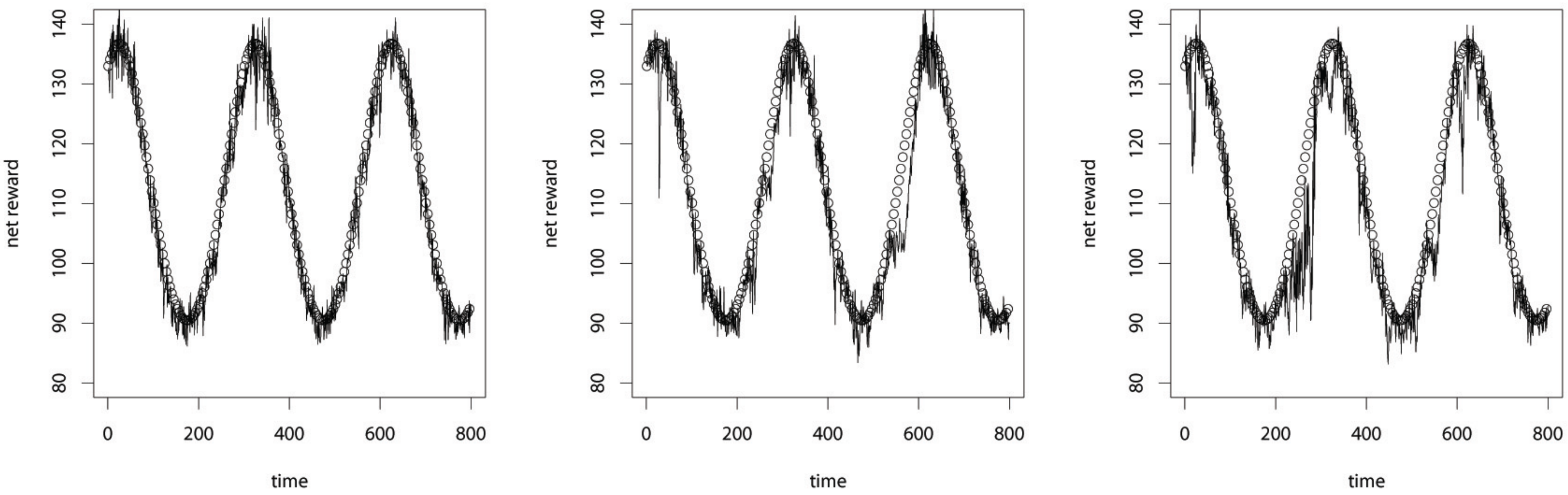






- Plot of time versus V-C.
- Noise does not significantly affect the algorithm.

w=10,20,30; 5% Gaussian Noise



- Plot of time versus V-C.
- Increasing window size increases error due to noise, and does not have a smoothing effect.

Limitations

For this to work,

- One must have a reasonable concept of cost and value for R.
- V, C, and P must be simply increasing in their arguments (e.g., $V(R+\Delta R)>V(R)$)
- $V(P(R))-C(R)$ must be convex (i.e., a local maximum is a global maximum)

Open questions

- How to design V and C to match SLAs.
- How to assure convexity of $V(P(R))-C(R)$.
- How to tune the size of ΔR.
- How to handle functions that can stay constant with increased resources or performance

Some hope…!

- To the best of our knowledge, a majority of value-cost functions are convex.
- If the first difference derivatives

  $(V_i(P_i+\Delta P)-V_i(P_i))/\Delta P$

  are simply increasing or decreasing in P, then

  $[\sum V_i(P_i(R))]-C(R)$

  Is convex.
- Step functions are easy to handle (to be discussed in ATC-2009 paper next week).

The big deal

- We did this without machine learning.
- We did it without a complete model.
- We traded complete modeling of P for constraint modeling of X (and P), a much simpler problem!
- Life gets simpler!

Dynamics of resource closure operators

Dr. Alva L. Couch

Marc Chiarini

Tufts University