The Role of Dead-Zones in Improving Run-to-Run Control Performance

Kostas Tsakalis, ASU, tsakalis@asu.edu

Aris Papadopoulos, ASU, arisagf@asu.edu

Kevin Stoddard, SEMY Eng. Inc., kstoddard@semy.com

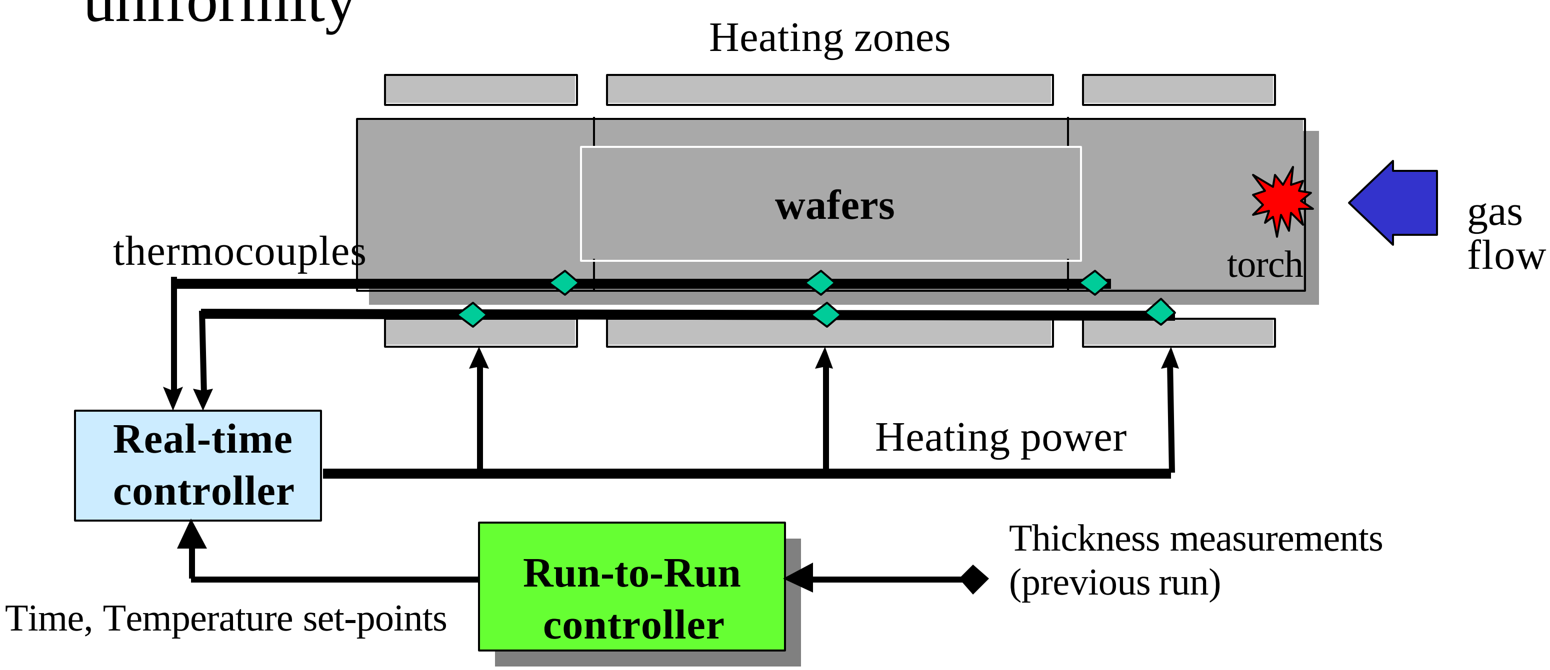
AEC/APC Symposium XI, Vail, Sept. 12-17, 1999

Introduction

- Run-to-Run Control Problem in Diffusion Furnaces
- ARRC Algorithms
  - Control input updates, Parameter estimation
- Speed vs. Variance Trade-offs
  - Bounded Noise: Nonlinear modifications
- Sensitivity to tuning parameters: smooth dead-zone
- Process Drift: Higher order controller (integral action)
- Conclusions

Run-to-Run Control in Diffusion

- Wet oxidation process for silicon oxidation
- Loss of symmetry (thermal gradients, long-term drift)
- R2R control inputs: processing time, temperature set-points
- Objectives: minimize deviations from target, across-the-load uniformity



Run-to-Run Control Algorithms

- SEMY's ARRC (Advanced Run-to-Run Control)
- **Modeling:** Least squares fit of experimental data
- **Control Updates:** Newton-like corrections

$$y_{k+1} = f(u_{k+1}) = f(u_k)+[df/du](u_{k+1}-u_k)$$

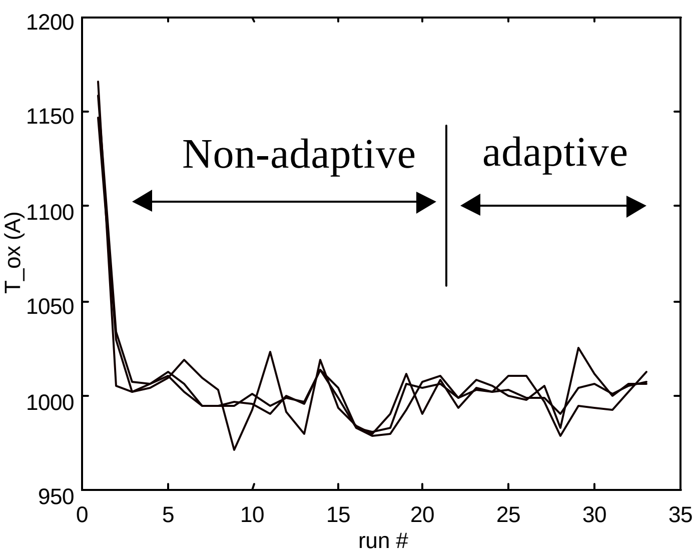
u: process input, y: process output, f: process model

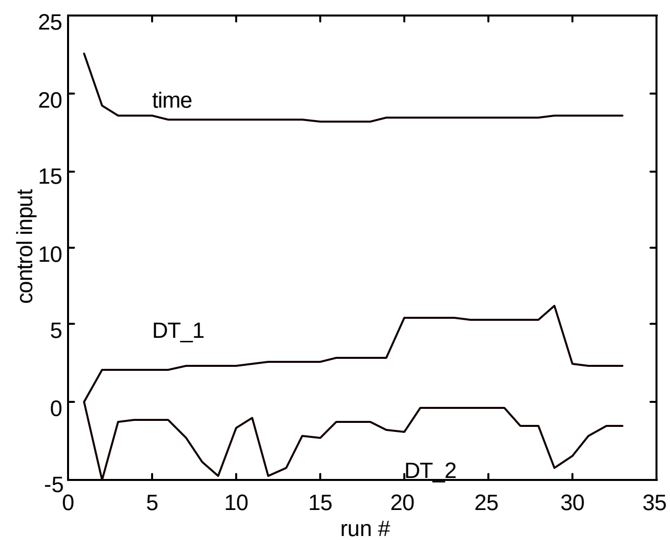
- **Parameter Updates:** Fading-memory least-squares with parameter constraints

R2R Controller Implementation

- Experimental test of a simple R2R controller:*
  - Quick centering of the process.
  - "Reasonable" steady-state variance, compared to the uncontrolled process. (can it be improved?)

* K. Tsakalis, M. Yelverton, B. Cusson, K. Stoddard, B. Schulze, "Run-to-Run Control: Application to Oxidation Processes," *Proc. 18th IASTED International Conference MIC'99*, Innsbruck, Feb. 1999.





Run-to-Run Control Algorithms

- A basic R2R controller

$$u_{k+1} = u_k + g_c \left(\frac{\partial f}{\partial u} \right)^{-1} d(e_k)$$

e_k: tracking error, *d(.)*: dead-zone function, g_c: gain

  – *d(.)* identity (linear controller): Standard trade-off between speed of convergence/drift attenuation and steady-state variance.

- Approx. error system: $e_{k+1} = e_k - g_c\, d(e_k) - Dn_{k+1} + Dr_{k+1}$
  $Dx_{k+1} = x_{k+1} - x_k$; *r:* reference/drift (low frequency),
  n: additive noise (high frequency or stochastic)

ARRC model adaptation (detail)

- Parameter Updates: Fading-memory least-squares

$$\boldsymbol{q}_{k+1} = \Pi\{\boldsymbol{q}_k + \boldsymbol{g}_p P_k^{-1} w_{p,k} e_{p,k} /(1+\boldsymbol{g}_p w_{p,k}' P_k^{-1} w_{p,k})\}$$

$$P_{k+1} = \boldsymbol{a} P_k + (1-\boldsymbol{a})Q + \boldsymbol{g}_p \boldsymbol{a} w_{p,k} w_{p,k}'$$

  – $w_p = \partial f / \partial \boldsymbol{q}$, a = fading memory, Π: parameter projection on a constraint set
  – Parameter projections and dead-zones are important to provide some immunity to noise-induced parameter drift
  – Ability to perform partial adaptation
  – Typical indirect adaptive control properties

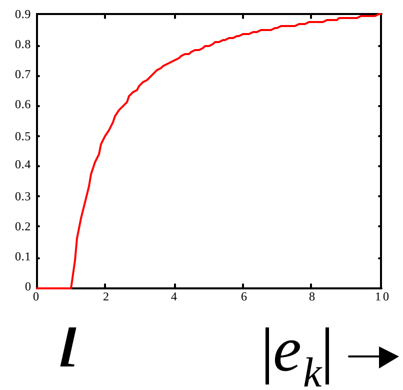
Run-to-Run Controller Properties 1

- Approximate error contributions
  - Low frequency: $\| e_k \| \leq {}^{1}\!/_{g_c} \| \Delta r_{k+1} \|$
  - Stochastic (uniform iid): $\mathrm{var}(e_k) = \sqrt{{}^{2}\!/_{2-g_c}}\ \mathrm{var}(n_k)$
- For a process with stochastic noise and no drift, the optimal gain approaches zero!
- For a drifting process, the optimum gain and variance depend on the drift and noise (practical estimates ?)

Run-to-Run Controller Properties 2

- Nonlinear gain idea: use “high” gain when error is large.
- Dead-zone effective gain:

  ($\boldsymbol{l}$: dead-zone threshold)

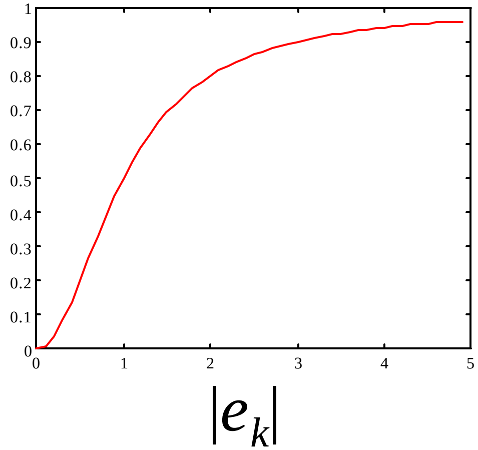


- **Dead-zone property**: In the absence of drift, if $|n_k| < \boldsymbol{l}$, then the error converges to a residual set $|e_k| < \boldsymbol{l}$
- **Implication:** Fast convergence to the residual set and “low” steady-state variance. (Drift induces O($\boldsymbol{l}$) bias)

Run-to-Run Controller: Smooth Dead-Zone

- Nonlinear gain, smooth approximation of the dead-zone:

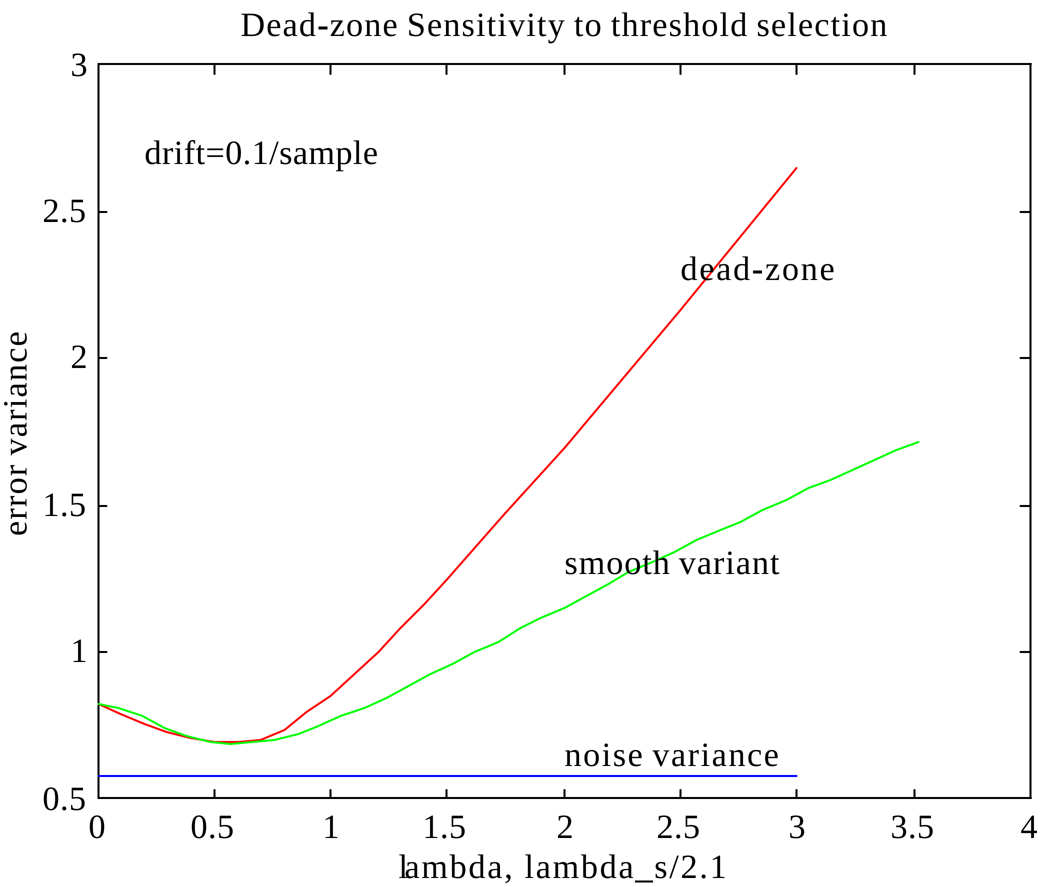
$$g_c = g(e_k/l_s)^2/[1+(e_k/l_s)^2]$$



  - Asymptotic stability (but not exponential)
  - When $n_k \rightarrow 0$ then $e_k \rightarrow 0$ (roughly as a cube-root)
  - Expect less sensitivity to threshold selection

Comparison of Dead-Zone Variants

- Dead-zone exhibits higher sensitivity to threshold selection than its smooth variant.
- Optimum and threshold relation is drift-dependent.



2nd Order R2R Controllers

- An additional integrator in the controller provides an internal model of the drift:

$$v_{k+1} = v_k + g_c \left(\frac{\partial f}{\partial u} \right)^{-1} d(e_k)$$

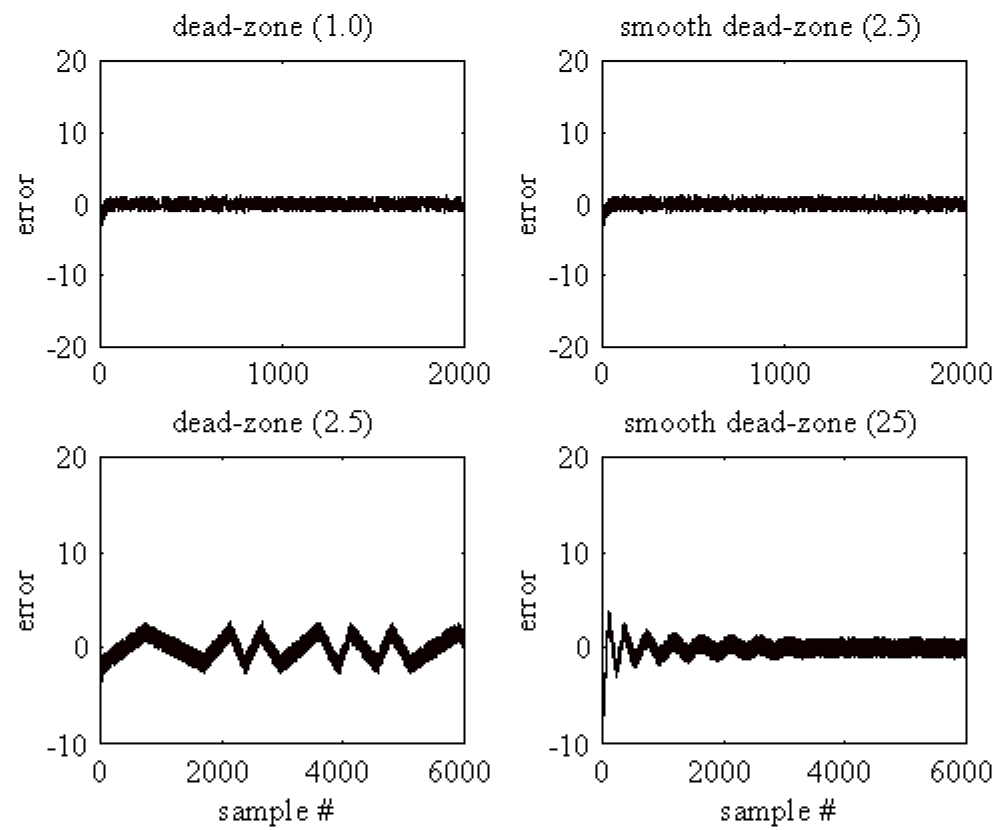
$$w_{k+1} = w_k + v_k$$

$$u_k = k_1 v_k + k_2 w_k$$

- k_1,k_2: a PID-like design (e.g., via LQR theory)
- Asymptotically zero-mean error regardless of the drift
- Smooth dead-zone: Asymptotic stability but not exponential in the final approach (much harder analysis)

2nd Order R2R Controllers

- Controller tuning is more involved due to sensitivity peaking.
- Almost open-loop variance can be achieved with “correct” gains and d-z thresholds.
- Smooth variant is less sensitive to threshold selection.



Run-to-Run Controller Tuning

- **1st order:** Construction of a look-up table: optimum gain & d-z threshold vs. drift-to-noise ratio
  - Low dimensional search via normalization
  - Typical solution: unity gain (high)

- **2nd order:** Sensitivity analysis via simulation (optimize frequency response characteristics; ad-hoc but effective)
  - Typical solution: Reasonable PI-gains

Conclusions

- Steady-state variance vs. speed of convergence (and drift correction) trade-off.
- For bounded noise, significant improvement is obtained with nonlinear controllers (dead-zone-type).
  - Near open-loop variance regardless of drift.
- Applicable to estimation/adaptation.
- Controller tuning via simulation-based look-up tables.
- Required properties of the noise can be estimated from modeling experiment and available production data.