Anytime-Valid Inference Under Outcome Delay: A Design-Based Approach

Michael Lindon[1] Nathan Kallus[2]

[1]Netflix, Los Gatos, CA [2]Cornell University, Ithaca, NY michael.s.lindon@gmail.com

Why This Problem Is Different

Delayed outcomes are common in online experiments:

- Marketing: ad exposure → purchase later
- Software: deployment → OOM error later

Treatment can change both **when** an outcome arrives and **what value** is observed.

At analysis time t, some outcomes are still censored. Treatment assignment affects both which outcomes have appeared by time t and the order in which units are observed.

Classical sequential methods index time by a fixed ordering of units. Here that ordering is random, so we define "sequential" with respect to external calendar time t.

An Honest Estimand for Inference

Determining early which arm is better under delay generally requires strong assumptions about future outcomes, such as stationarity, identical distributions over time, or proportional hazards. Those assumptions are often not credible.

Instead, we ask: **what estimand makes sense to report without such assumptions?** Our answer is cumulative reward up to calendar time t. The intervals summarize what can be said honestly *so far*, without extrapolating beyond the observed horizon.

What We Estimate

Unit i enters at time E_i and has fixed potential outcomes $(t_i(w), y_i(w))$ for $w \in \{0,1\}$.

Potential outcomes example: each unit has two potential times and two potential values, but only one pair is observed.

i	E_i	$t_i(0)$	$t_i(1)$	$y_i(0)$	$y_i(1)$	w_i	t_i	y_i
0	.0001	3.39	0.94	0.19	0.28	1	0.94	0.28
1	.0004	2.32	1.40	0.64	0.25	1	1.40	0.25
2	.0029	4.96	0.33	0.96	0.22	0	4.96	0.96

Treatment can change event time and outcome value, so the realized sample by time t depends on treatment assignment.

Potential reward process:

$$r_{it}(w) = y_i(w)\,\mathbf{1}[t_i(w) \le t]$$

Calendar-time estimands:

$$r_t(w) = \sum_{i \in \mathcal{E}_t} r_{it}(w),$$
$$\Delta_t = r_t(1) - r_t(0)$$

Example: if N users have entered by time t, then $r_t(1)$ is the revenue observed by time t if all N had received treatment.

Core estimand choice: cumulative reward asks what would have been observed up to time t if everyone had received arm w. It avoids modeling unobserved future outcomes.

Why design-based?

- no superpopulation model for outcome times or values
- robust to nonstationarity, heterogeneity, and staggered entry
- early uncensored units do not stand in for later censored units

Reward Estimators

Armwise reward estimator:

$$\hat{r}_{it}(w) = m_{it}(w) + \frac{\mathbf{1}[w_i = w]}{\pi_i(w)}\big(r_{it}(w) - m_{it}(w)\big)$$

$$\hat{r}_t(w) = \sum_{i \in \mathcal{E}_t} \hat{r}_{it}(w), \qquad \hat{\Delta}_t = \hat{r}_t(1) - \hat{r}_t(0)$$

Here $m_{it}(w)$ is an augmentation term.

What Provably Fails For The Treatment Effect

$\hat{\Delta}_t - \Delta_t$ is **not a martingale under any filtration**.

Obstacle 1: unestimable cross term

$$\mathrm{Cov}(\hat{r}_{it}(1), \hat{r}_{it}(0)) = -e_{it}(1)\,e_{it}(0).$$

This term depends on both potential outcomes, so it cannot be estimated from observed data. That is the same structural obstacle behind the classical design-based variance upper bound.

Obstacle 2: incompatible martingale structures

Randomization induces cross-arm dependence, and the two armwise martingales live on incompatible filtrations. So even with access to the cross term, the treatment-effect error would still not inherit a martingale structure.

Armwise Confidence Sequences

Per arm:

$$\Pr\Big(\forall t \ge 0: \ |\hat{r}_t(w) - r_t(w)| \le b(\hat{V}_t(w);\alpha)\Big) \ge 1-\alpha$$

$$\hat{V}_t(w) = \sum_{i \in \mathcal{E}_t} (1-\pi_i(w))\,\hat{e}_{it}(w)^2,$$
$$\hat{e}_{it}(w) = \hat{r}_{it}(w) - m_{it}(w)$$

Boundary:

$$b(V;\alpha) = \sqrt{\frac{V\eta^2+1}{\eta^2}\log\left(\frac{V\eta^2+1}{\alpha^2}\right)}$$

Armwise error:

$$M_t(w) = \hat{r}_t(w) - r_t(w)$$

Single-arm filtration:

$$\mathcal{F}_t(w) = \sigma\Big(\{(E_i, X_i) : E_i \le t\} \cup \{w_i : t_i(w) \le t\}\Big)$$

Key result: the AIPW error process $M_t(w)$ is a martingale with respect to $\mathcal{F}_t(w)$ **if and only if** $m_{it}(w) = 0$ for $t < t_i(w)$ and $m_{it}(w)$ is constant on $[t_i(w), \infty)$.

These armwise sequences are the ingredients for inference on Δ_t.

Confidence Sequences on Δ_t

Union bound: build confidence sequences for $r_t(0)$ and $r_t(1)$ separately, then combine them:

$$\hat{\Delta}_t \pm \Big(b\big(\hat{V}_t(0);\tfrac{\alpha}{2}\big) + b\big(\hat{V}_t(1);\tfrac{\alpha}{2}\big)\Big)$$

With constant assignment probability π, compare its width to the variance-upper-bound benchmark:

$$R_\pi(V_0, V_1) = \frac{b(V_0;\alpha/2) + b(V_1;\alpha/2)}{b\big(\frac{V_1}{1-\pi} + \frac{V_0}{\pi};\alpha\big)}$$

If $V_0 = V_1 = V$, then $R_\pi(V,V) \to 2\sqrt{\pi(1-\pi)}$; under balanced assignment this equals 1. If $V_0 = o(V_1)$, then $R_\pi(V_0, V_1) \to \sqrt{1-\pi}$; under balanced assignment this is $1/\sqrt{2}$.

Delayed outcomes create exactly this asymmetry: $V_t(1)$ and $V_t(0)$ can evolve very differently early in the experiment.

Evidence 1: Synthetic Delayed Outcomes

DGP: $E_i \overset{iid}{\sim} \mathrm{Uniform}(0,10)$ and $w_i \sim \mathrm{Bernoulli}(0.5)$.

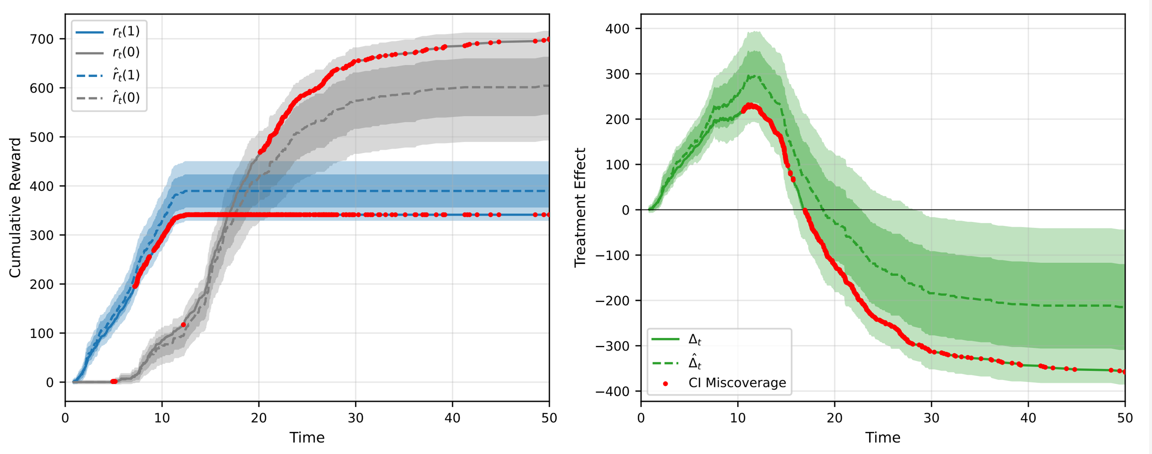
$$\lambda(t \mid E, w) = \lambda_{\mathrm{base}}(t-E, w)\,\lambda_{\mathrm{cycle}}(t)\,\lambda_{\mathrm{shock}}(t, w)$$

with treatment-specific log-normal baseline hazards, weekly seasonality, and localized calendar-time shocks.

$$y_i(w) = \beta_w\big(1 + 0.15\log(1+s_i)\big)\big(1 + 0.1\sin(2\pi t_i(w)/7)\big)\epsilon_i$$

where $s_i = t_i(w) - E_i$, $\beta_0 = 1$, $\beta_1 = 0.6$, and $\epsilon_i \sim \mathrm{Uniform}(0.9, 1.1)$.

This creates a pull-forward effect: treatment outcomes arrive earlier but have smaller value on average.



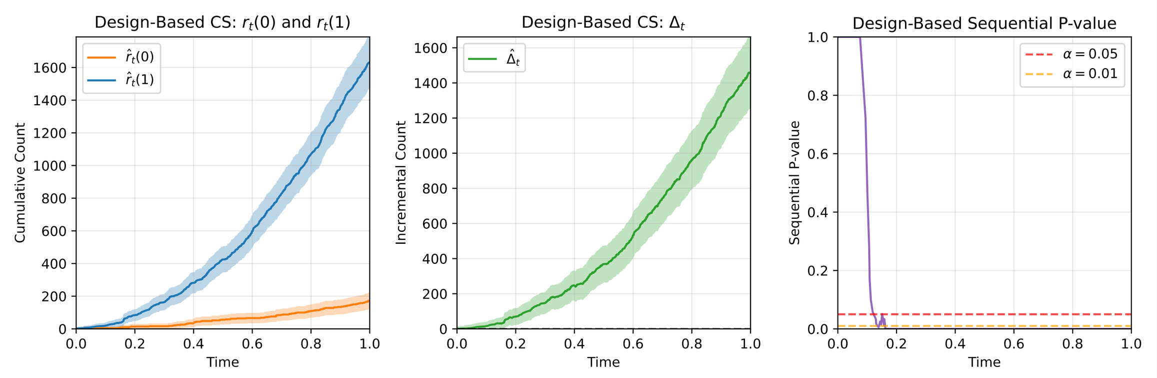
One realization. Classical pointwise intervals repeatedly miscover (red dots), while the anytime-valid confidence sequences maintain coverage. Treatment looks better early, but control wins later in cumulative reward.

Evidence 2: Real Software Telemetry

Software canary monitoring for out-of-memory (OOM) errors. A memory leak gradually depletes available memory, so the delay from software startup to crash can range from minutes to hours depending on usage patterns and device state.

This makes OOM errors a natural delayed-outcome problem: the release may be harmful, but the evidence arrives gradually. In a canary rollout, the operational question is whether to abort and roll back the new release before more devices are affected.

Setting $y_i(w) = 1$ makes the estimand a counterfactual counting process: how many unique devices would have encountered an OOM error by the current time t if all had received arm w?



The same design-based construction yields armwise confidence sequences, a treatment-effect confidence sequence, and a sequential p-value for bug detection without probability point-process models (like Poisson).

Takeaways

1. **Honest estimand:** cumulative reward up to calendar time t, a counterfactual business outcome rather than a property of a distribution, chosen precisely to avoid extrapolation beyond the observed horizon.
2. **Why design-based:** treats potential outcomes as fixed, needs no superpopulation model, and is robust to nonstationarity and staggered entry.

Paper

Scan for the arXiv version.