# Statistical Estimation of Aperiodic Response Times when Scheduled on top of Static Timelines†

Pam Binns
Honeywell Laboratories
Minneapolis, MN 55418
Pam.Binns@honeywell.com*

## Abstract

*Safety-critical embedded control systems must support both periodic and stochastic functions on common hardware. Scheduling techniques that produce a timeline with predefined blocks for periodic functions are used to guarantee deadlines for critical closed loop periodic control functions (i.e. sensor read, control, actuator write). Aperiodic processing occurs in periodic timeline gaps. We present aperiodic response time estimation data using a new automated binning technique based on a fluid flow analysis coupled with observations on conditional probabilities for response time values. Prior simulation results showed good response estimates under a broad range of conditions for a first-in first-out (fifo) aperiodic server. This paper presents new response time estimates, collected from a real-time testbed, using the binning algorithm when an earliest deadline to start (eds) server is used.*[1] *The eds results are compared to fifo server results. We discuss benefits, shortcomings, and possible future directions for this statistical approach.*

## 1. Problem Statement and Some Related Work

Safety-critical embedded control systems must support both periodic and aperiodic functions on common hardware (e.g. sending messages on a common data bus). Periodic tasks have a fixed period and a worst case execution time (WCET). Periodic applications are typically of the form sensor read, control, and actuator write. State-of-practice scheduling applications for critical control applications are often static (*e.g.* ARINC 653 or messages on a time-triggered bus) and produce a timeline with predefined blocks for periodic functions to guarantee deadlines for critical closed loop periodic control functions.

In statically scheduled systems, aperiodic processing occurs in periodic timeline gaps or equivalently, the background of the periodic timeline. The stochastic stream has randomly generated interarrival times (with no minimum inter-arrival time requirement) and service (or message transmission) times, and for eds servers, randomly generated relative deadlines. The aperiodic interarrival and service rates are denoted by $\lambda$ and $\mu$, respectively. Applications that give rise to situations like this include processing button pushes, remote procedure calls, sending messages over time-triggered buses, and scheduling stochastic events in a statically time partitioned system (across multiple time boundaries over which the task waits or executes). To the best of our knowledge, the general problem of aperiodic response time prediction remains unsolved. The spectrum of approaches for predicting response times ranges from first principle analyses to simulation.

Periodic task scheduling and analysis has been well studied[1]. The literature on purely event driven systems is huge. Tractable analytic approximations have been successfully found when queue lengths are long ([4] for fifo priority queues and [5] for edf queuing networks) and all traffic is aperiodic. For predominantly periodic traffic, utilizations can be close to 1, but aperiodic queue lengths remain short, in which case approximations based on heavy traffic theory (with long queue lengths) can provide very optimistic estimates. Numeric solutions of a stochastic process specification can be viable, but for high fidelity models the solution times tend to be long and solution values are approximate and specific to the parameter set.

At the other end of the spectrum is simulation or actual system observation, from which data is collected and an empirical (response time) distribution function (EDF) is constructed. Given a sample of $k$ independent and identically distributed (*iid*) response time values, $\{x_1, x_2, ..., x_k\}$, the true (unknown) response distribution is $p_z = Pr[R \leq z]$ is estimated by $\hat{p}_{z,k} = k^{-1}[\#x_j \leq z]$. Using a Kolmogorov-Smirnov theorem [3] a level $1 - \delta$ confidence band is constructed by

$$P[\sup_z |\hat{p}_{z,k} - p_z| > \epsilon] \leq 2e^{-2k\epsilon^2} = \delta. \qquad (1)$$

A single empirical distribution function can be compared to a single simulation in that the results apply only to the particular setting (*viz.* system and parameter configurations such as $\lambda$ and $\mu$) from which the data was gathered.

We investigate response time data using a binning technique based on a fluid flow analysis coupled with observations on conditions for response time values. The same bin values are used for a fixed periodic timeline, with aperiodic parameters being permitted to vary. The compact number

*This work was supported in part by DARPA and AMCOM under contract number DAAH01-00-C-R226.

[1] Deadlines in eds and edf (earliest deadline first) are with respect to start and end of execution, respectively. In eds, started task executions or message transmissions are not preempted by other tasks from the same input source. True preemptions on buses are rarely supported.

of bins yields a parameterized response time representation suitable for real-time estimation and goodness-of-fit tests for decision making (See section 3.1).

## 2. Binning Point Algorithm Outline

Bin generation, illustrated in Figure 1, consists of two phases. Below, we try provide some intuition for how it works. Algorithm details are available in [2].

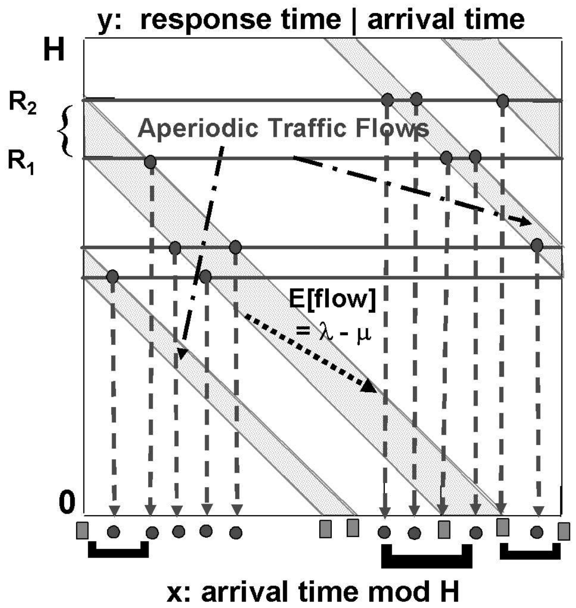


**Figure 1. Bins Generation Algorithm**

The hyperperiod $H$ is the smallest time for the periodic message transmission cycle to repeat. Periodic blocking (busy) intervals alternate with gaps. The $i^{th}$ blocking interval begins at time $b_i$ relative to frame start where $b_1 = 0$ and $b_{m+1} = H$. The $i^{th}$ gap (idle interval) starts at time $g_i$ and has duration $b_{i+1} - g_i$.

Define the initial set of binning points $BI_i = \{0 = b_1, g_1, ..., b_m, g_m, b_{m+1} = H\}$ (*i.e.* the squares along the $x$-axis). This set identifies abrupt changes in the stochastic flow of aperiodic message transmissions. During a busy interval, arriving aperiodic messages queue and are blocked. In an aperiodic gap, any queued backlog discharges at an average rate $\mu - \lambda$ until the queue empties or is again blocked. Figure 1 illustrates block/gap response time *bands* for a static periodic timeline. Aperiodic *work* (transmission or execution) occurs only in the shaded bands. The white bands correspond to blocking intervals.

For an aperiodic fifo server, the response time is equal to the pending aperiodic and periodic work at the time of arrival (including its work), plus all future periodic blocking times. Given an arrival at time $t$ (mod H), the response time must fall in one of the shaded areas when drawing a vertical line from $t$ up. Further binning points are needed to classify response times in bands for which multiple arrival times might have occurred. For example, in Figure 1 if an observed response time $R \in [R_1, R_2]$ then the set of all times during which the arrival might have occurred is marked along the x-axis using rotated brackets (*i.e.* [ ). For a particular response time interval, if there is more than one time interval (mod H) during which the arrival might have occurred, we call the set of intervals confounding regions. For $R \in [R_1, R_2]$ there are three confounding regions.

A final set $BI_f$ of binning points over a hyperperiod is computed by computing confounding regions relative to response time intervals defined when transitioning from busy to idle. Finding these new points (marked by circles on the $x$-axis) amounts to solving for points where lines intersect, which is both automatable and computationally very tractable. We have found that under a range of conditions, it is sufficient to consider only values of $R$ defined when alternating between blocking intervals and gaps because the aperiodic process flow behaves linearly within the gaps.

Intuitively, $BI_f$ contains points at which changes in stochastic flow are likely to be observed. If response times over $n$ hyperperiods are of interest, $BI_f$ is expanded to $\bigcup_{1 \leq j \leq n} j \cdot BI_f$. A compact, fixed size response time vector is defined by $R(BI_f) = [p_1, p_2, ..., p_l]$ where $p_i = P[\mathrm{R} \leq bp_i]$ for $bp_i \in BI_f$. Only in simple cases for fifo servers do we know how to exactly calculate $p_i$. Instead, we use empirical estimates of $p_i$ with confidence bands as specified in Equation 1. Since we are observing (rather than computing) response times, our binning point analysis only makes assumptions about aperiodic work discharge rate (or flow), which does not depend on the ordering of queue entries. Hence our algorithm appears to hold for service disciplines with other orderings, such as eds. Linear interpolation is used for response time estimation between binning points.

## 3. Response Time Validation and Discussion

We now compare some results of our binning techniques with the traditional empirical distribution function estimate and its associated confidence bands. Space limits the amount of data that can be shown. The data was produced by a synthetic workload on a real-time testbed. To approximate some avionics and automotive applications, the percentage of periodic traffic, $\rho_1$, constitutes the majority of traffic. Table 1 lists values for two different $BI$, the busy/idle binning points in a single hyperperiod $H$, for $H = 200$ ms.

In all figures, the mean arrival and service rates, $\lambda$ and $\mu$ are measured in tasks per second. The hyperperiod $H = 200$ ms. The maximum absolute relative deadline to start of service is $D$ ms. Task interarrival times are uniform on $[0, 2\lambda^{-1}]$, task (message) durations are uniform on $[0, 2\mu^{-1}]$ and task deadlines are uniform on $[-D, D]$. The absolute deadline to start of service is the arrival time plus

| $m$ | $\rho_1$ | $BI_i = [0 = b_1, g_1, ..., b_m, g_m, H]$ <br> $BI_f =$ all BPs in H; size$(BI_f) = l$ | size $BI_f$ |
|---|---|---|---|
| 3 | 0.67 | $BI_i = H[0, 0.42, 0.5, 0.7, 0.75, 0.8, 1.0]$ <br> $BI_f = [0, 10, 16, 40, 50, 56, 60, 66, 76,$ $84, 100, 116, 124, 134, 140, 144, 150, 160,$ $180, 184, 190, 200]$ | 23 |
| 1 | 0.90 | $BI_i = H[0, 0.90, 1.0]$ <br> $BI_f = [0, 20, 180, 200]$ | 4 |

**Table 1. Initial and Final Binning Point Sets**

the random relative deadline. We elected eds service instead of edf service to avoid task message preemption, which is not supported on most buses.

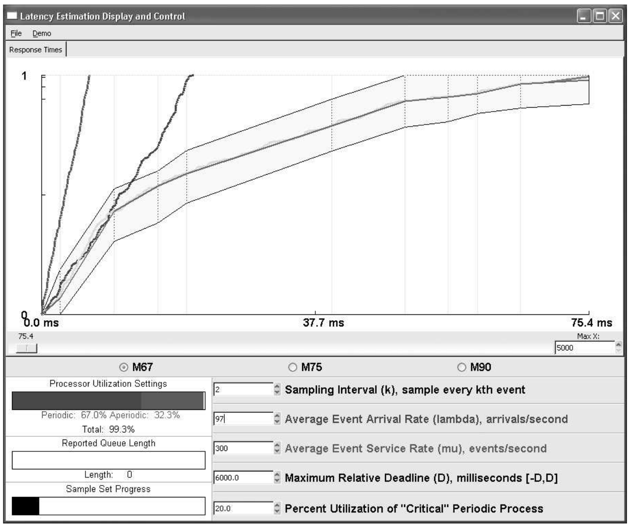


**Figure 2. M67 EDS Response Time Data**

When $D = 0$, the aperiodic server is fifo. When queue lengths are often zero or one, fifo and eds (or edf) servers differ little, sometimes to the point of being statistically indistinguishable. When the largest absolute relative deadline is one to two orders of magnitude larger than the average interarrival time, differences in the fifo and eds response time distributions appeared most visible.

The three darkest lines in the figures (from left to right) are task service time, interarrival time, and response time distributions. 80% confidence bands are computed to bracket and shade a response time distribution. The faint vertical lines are binning points. The sample size $k = 250$.

In Figures 2 and 3, $\rho_1 = 0.67$, $\mu = 300$, and $\lambda = 97$. The response time numbers along the $x$-axis run from 0 to 75.4 ms. The $y$-axis is a probability, so runs from 0 to 1.

Figure 2 is an eds server with $D = 6000$ and Figure 3 is a fifo server. The eds confidence band in Figure 2 is unchanged in Figure 3 to illustrate the contrast between fifo and eds at the same parameter settings. The shallower left hand rise and smaller right tail in response times are characteristic when comparing a fifo server to an eds server.

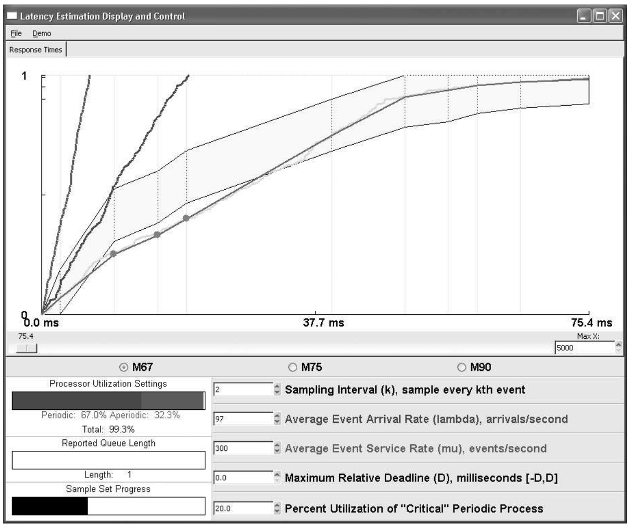


**Figure 3. M67 FIFO RT Data in EDS CB**

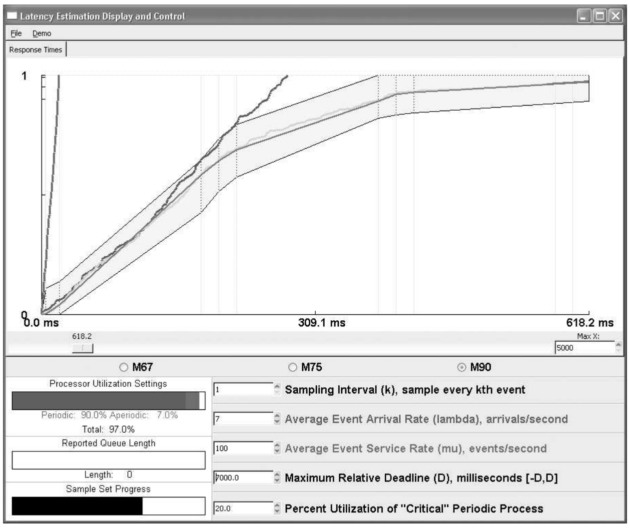


**Figure 4. M90 EDS RT Data in FIFO CB**

In Figure 4, $\rho_1 = 0.9$, $\lambda = 7$, $\mu = 100$, and $D = 7000$. The $x$-axis response time numbers run from 0 to 618.2 ms. The confidence bands were computed for a fifo server at the same parameter settings. Since aperiodic queues are rarely longer than 1, no statistically significant difference in response times between the servers was observed. Total measured system utilization is 97%. Note how the binning points capture many of the inflection points in the aperiodic response time curve when periodic traffic is heavy.

**3.1 Some Advantages of Binning Points**

The number of bins is compact and deterministic. The number of support points (x-values with non-zero probability) is defined by the number of bins $n$, not the number of sample points $k$. One can also attempt to reduce the final number of binning points. Points in $BI$ that are close to one another might be collapsed (*e.g.* delete one or use an average).

Because the number of bins $n$ depends only on the timeline generated by periodic message transmission, the support points are known *a priori* so response time values observed on-line can be quickly recorded. One benefit of online real-time response time estimation using observed data in a networking environment is an efficient and compact representation when passing information to higher level decision making protocols.

Alternatively, decisions might be made locally using statistical tests to detect changing local conditions. Suppose $F_n$ and $G_n$ are response time estimates where $F_n$ was computed earlier in time than $G_n$, where $n$ is the number of binning points. Since all points reside on a common support, when comparing two different binned response time estimates, statistical tests to detect changes in latencies are $O(n)$ compared to $O(k^2)$, where often $k \gg n$. The value of $k$ depends on desired significance level and power of the tests. The two-sided Kolmogorov-Smirnov statistic $D_n = \sup_x |F_n(x) - G_n(x)|$ might be applicable to detect changes at level $\alpha \leq 0.2$ using the test

$\text{Reject } H_0 : F = G \text{ at level } \alpha \text{ for } D_n > \sqrt{-(\ln(\alpha/2))/k}.$

**3.2 Some Disadvantages of Binning Points**

Our approach is not without potential shortcomings. We have observed that when queue lengths are very long, also with long response time values (say, 10 or more hyperperiods long) closed form heavy traffic models ([4, 5] are likely to provide a reasonable and more compact approximation.

We have observed cases where the linearly interpolated binning point response time confidence bands do not contain the observed empirical distributions. Space restrictions preclude illustrations here. Examples where linear interpolation is not a good estimator along with candidate remedies can be found in [2].

When tasks complete early and idle periods are not inserted for the remaining duration (*e.g.* a preemptive fixed-priority scheduler), the upper band structure shown in Figure 1 is not exact. For fixed release times, the lower blocking bands appear unchanged. The selection of a "good" set of upper band binning points will depend on the variability of the periodic tasks' execution times. Our bands provide an upper bound estimate given available response time data when periodic tasks use their WCETs. There may be cases where average periodic task execution times might determine useful upper bands.

# 4. Some Future Directions

Our response time estimation procedure appears to work best when aperiodic message transmission durations are small relative to $H$ and when the majority of aperiodic response times span at most a few hyperperiods (*e.g.* aperiodic ack/naks or interrupts). Our technique is applicable for estimating response times for soft real-time aperiodics, and possibly for finding reliable estimators that might be used for hard-deadline aperiodics (*e.g.* a value for which 95% of all messages arrive with high probability).

Formulating system models so data can reliably be used in statistical tests with quantifiably high assurance can be challenging. Suitable data might not be available (*i.e.* too little data, not *iid* data, *etc.*) or statistical tests might not even exist. When estimators are used only for non-critical applications, rigorously quantifiable statistical claims may not be required (from a certification perspective).

Heuristics are common real-time change detection mechanisms for dynamic adaptation. Accurate differentiation between transient fluctuations and long term system state changes is a challenge when applying (or developing) statistical tests for quantifiable detection of dynamic changes in system configuration.

We evaluated two different aperiodic servers; fifo and eds. Our analysis used only the fluid flow discharge rate and made no explicit assumptions about message ordering provided deadlines are independent of both message arrival and service times. Other server disciplines might also work. We previously evaluated exponential servers and interarrival time distributions. We suspect that our binning technique might also work with other distributions.

We have ignored the practical concerns of message overheads such as context switching and bus arbitration. Arbitrating among distributed aperiodic queues on a multi-drop bus at the end of a periodic busy interval is a difficult problem in its own right. Lastly, our response time estimation procedure was applied to aperiodic traffic over only a single periodic timeline (as defined by a single bus or processor).


# 5. Acknowledgements

Many thanks to Dan Churchill for his implementation efforts and to Steve Vestal for helpful suggestions.


# References


[1] N.C. Audsley, A. Burns, R.I. Davis, K.W. Tendell and A.J. Wellings, "Fixed Priority Pre-emptive Scheduling: An Historical Perspective," *Real-Time Systems*, 8, 1995

[2] Pam Binns, "Real-Time Estimation of Event-Driven Traffi c Latency Distributions when Layered on Static Schedules," *IEEE Proceedings of the 2003 International Conference on Dependable Systems and Networks* (DSN 2003)

[3] Vladimir N. Vapnik, "Statistical Learning Theory," John Wiley & Sons, 1998

[4] Ward Whitt, "Weak Convergence Theorems for Priority Queues: Preemptive Resume Discipline,"*Journal of Applied Probability*, Vol. 8, 1971

[5] H. Zhu, J. Lehoczky, J. Hansen, R. Rajkumar, "Design Trade-Offs for Networks with Soft End-to-end Timing Constraints", *Proceedings of the Tenth Real-Time and Embedded Technology and Applications Symposium*, RTAS 2004