Online Supplement
to
Delay-Based Service Differentiation with Many Servers and Time-Varying Arrival Rates

Xu Sun and Ward Whitt
Department of Industrial Engineering and Operations Research, Columbia University
New York, NY, 10027

January 22, 2018

1 The Simulation Experiments

We first describe the experimental setting considered and the staffing algorithm. Then we describe our simulation algorithm.

1.1 The Experimental Setting

As in most earlier work on queues with time-varying arrival rates, we use the sinusoidal arrival-rate functions. Here we consider a two-class $M_t/M/s_t + M$ model with arrival-rate functions in (1.1).

$$\lambda_1(t) = a_1 + b_1 \sin(d_1 t) \quad \text{and} \quad \lambda_2(t) = a_2 + b_2 \sin(d_2 t) \quad \text{for} \quad 0 \le t \le T. \tag{1.1}$$

However, our simulation algorithm is not limited to sinusoidal arrival rate functions. For homogeneous (class-invariant) service, we determine the time-staffing staffing level based on the aggregate arrival rate $\lambda \equiv \lambda_1 + \lambda_2$. Specifically, we apply the well-know square-root safety staffing rule to the aggregate model. A crucial quantity is the time-varying offered-load $m^\infty(\cdot)$ which has a convenient formulas. Eick et al. (1993) showed that

$$m^\infty(t) = \int_{-\infty}^{t} G^c(t-u)\lambda(u)\mathrm{d}u = \mathbb{E}\left[\int_{t-S}^{t} \lambda(u)\mathrm{d}u\right] = \mathbb{E}\left[\lambda(t-S_e)\right]\mathbb{E}[S]$$

where S represents a generic service-time random variable with cumulative distribution function (CDF) $G(t)$, $G^c(t) \equiv 1 - G(t) \equiv \mathbb{P}(S > t)$ and S_e denotes a random variable with the associated stationary-excess CDF, defined by

$$G_e(t) \equiv \mathbb{P}(S_e \le t) \equiv \frac{1}{\mathbb{E}[S]} \int_0^t G^c(u)\mathrm{d}u, \quad \text{for} \quad t \ge 0.$$

If S has an exponential distribution ($G = M$), then m^∞ satisfies

$$\dot{m}^\infty(t) = \lambda(t) - \frac{m^\infty(t)}{E[S]}.$$

Given the formula of the offered-load process $m^\infty(\cdot)$, we can apply the following formula

$$s_t = \lceil m^\infty(t) + \tilde{c}(t)\sqrt{m^\infty(t)} \rceil \tag{1.2}$$

to determine the staffing level.

If service times are class-dependent, then apply

$$m_1^\infty(t) = \int_{-\infty}^t G_1^c(t-u)\lambda_1(u)\mathrm{d}u \quad \text{and} \quad m_2^\infty(t) = \int_{-\infty}^t G_2^c(t-u)\lambda_2(u)\mathrm{d}u.$$

and then calculate $m^\infty \equiv m_1^\infty + m_2^\infty$. We then use the formula in (1.2) to determine the staffing level.

1.2 The Staffing Algorithm

In the paper, we allow a customer in service to be relegated to a high-priority queue (HPQ) when the staffing level is forced to decrease and all servers are busy. In the simulation, we do not force a customer out of service if the staffing level is scheduled to decrease. Instead, we release the first server that becomes available after the time of scheduled staffing decrease.

In practical situations, one usually needs to develop a shift schedule that allows the servers to leave the system (as s_t decreases) in the order of their arrival. This can be done by allowing server switching, as described in §3.2 of Whitt and Zhao (2017). To be more precise, we do not require the customer in service to stay with the same server until the service is complete, when that server is scheduled to depart. Instead, we allow the service in progress to be handed off to another available server.

To determine the timing of staffing changes, we exploit the deterministic staffing function, as given in (1.2). In particular, given that function, we construct a sequence of staffing values $\{s_i\}$ and a strictly increasing sequence of staffing change times such that

$$s(t) = s_i \quad \text{for} \quad t_{i-1} \le t < t_i.$$

1.3 The Simulation Algorithm

Our simulation algorithm falls under the discrete-event simulation framework where each event occurs at a particular instant in time and marks a change of state in the system. Between consecutive events, no change in the system is assumed to occur, that is, the simulation program can directly jump in time from one event to the next.

In the simulation experiments, we first calculate the staffing levels and record the times of staffing changes and the corresponding staffing levels in two vectors. For each simulation run, we start the system empty at time zero, so that there is an initial warmup period before the system reaches its steady state. For the most part, the warmup period is easy to interpret in the plots, but it lasts longer as the service-time variability increases, we could elect to staff to stabilize during the warmup period too. But we do not do that, because it is not our primary concern. Customer arrivals, departures and abandonments are simulated in the usual way; see Chapter 7 in Ross (1990). In addition to the time of abandonment, the algorithm maintains the time of arrival for each customer in queue. At any point in time, the HoL delay can be computed by taking the difference between the current time and the arrival time of the HoL customer.

It remains to specify how different scheduling rules are implemented in the simulation experiments. In both the main paper and the present document, we report the simulation results with three different scheduling policies, namely, the fixed-queue-ratio (FQR) rule, the head-of-line-delay-ratio (HLDR) rule and the time-varying-queue-ratio (TVQR) rule.

The FQR rule uses two constants r_1 and r_2 as ratio (control) parameters. At each departure epoch, the algorithm looks to see whether there are customers in queue(s). If there is a non-empty queue, the algorithm compares $Q_1(t)/r_1$ with $Q_2(t)/r_2$. The algorithm chooses a class-1 customer to enter service if $Q_1(t)/r_1 > Q_2(t)/r_2$ and chooses a class-2 customer if $Q_1(t)/r_1 < Q_2(t)/r_2$. If there is tie, each class is chosen equally likely.

Similar to the FQR rule, *the HLDR rule* uses two constants v_1 and v_2 as ratio parameters. At each departure epoch, the algorithm looks to see whether there are customers in queue(s). If there is a non-empty queue, the algorithm compares the two weighted HoL delays $w_1(t)/v_1$ and $w_2(t)/v_2$. The class with a greater weighted HoL delay is chosen to enter service.

The TVQR rule operates in the same way as the FQR control, except that the algorithm uses two ratio functions (rather than two queue-ratio parameters) $r_1(\cdot)$ and $r_2(\cdot)$ satisfying

$$r_1(\cdot) = \frac{\lambda_1(\cdot)}{\lambda_1(\cdot) + 2\lambda_2(\cdot)} \quad \text{and} \quad r_2(\cdot) = \frac{2\lambda_2(\cdot)}{\lambda_1(\cdot) + 2\lambda_2(\cdot)}$$

and compares $Q_1(t)/r_1(t)$ with $Q_2(t)/r_2(t)$ at each decision epoch.

1.4 Data Collection and Statistical Precision

The estimate the mean queue lengths (delays), we record the queue length (delay) of each class at each departure epoch; i.e., *we take each departure epoch to be a sampling time.* We then divide the

time horizon $[0, 50]$ into many small subintervals of length $\Delta = 5/24$; for each subinterval, we compute the sample average of all queue lengths (delays) registered over that subinterval. Since the number of replications is $r = 4,000$, we have r values for each subinterval. The final mean queue length (delay) at each subinterval is then taken to be the sample average of the r values.

To see that our experimental design should provide good statistical precision, consider the stationary model with aggregate arrival rate $a_1 + a_2 = 150$. The departure rate therefore is around 150. Over each subinterval, there are approximately $150 \times 5/24 = 31.25$ sample points for each replication. Overall, there are $31.25 \times 4000 = 125,000$ data points within each subinterval. Note that these sample points are not mutually independent. But the $r = 4,000$ sample averages are necessarily statistically independent. Hence our overall estimate has a variance of order $O(10^{-4})$ and a std of order $O(10^{-2})$, which is very good.

2 The Results of the Experiments

Our base case is the two-class $M_t/M/s + M$ model with (class-invariant) exponential service and the sinusoidal arrival-rate functions in (1.1) with $(a_1, a_2) = (60, 90)$, $(b_1, b_2) = (-20, 30)$, $d_1 = d_2 = 1/2$. Impatience times are exponentially distributed with rates $\theta_1 = \theta_2 = 0.5$. Afterward, we consider $\theta_1 = \theta_2 = 0.2$ (low abandonment rate) and $\theta_1 = \theta_2 = 0.8$ (high abandonment rate), which helps us gain insight into the effects of customer abandonment on the system performance. Then we consider the more difficult case of class-dependent service.

2.1 Exponential Service with Homogeneous Service Rates

Our primary objective is to achieve a desired delay-ratio equal to 1/2. By the analysis of Gurvich and Whitt (2009) we infer that the long-run average queue-ratio should be approximately equal to $(1/2)(60/90) = 1/3$. We would then want to use the FQR rule with target queue-ratio $r = 1/3$. With this value, we understand that the ratio Q_1/Q_2 is expected to be around the target 1/3. In the simulation, we set $c = 0$.

2.1.1 The Base Case

We start with a base case where both service and impatience times are exponentially distributed with rates $\mu = 1$ and $\theta_1 = \theta_2 = 0.5$, respectively.

Figure 3a - 3b report the simulation estimates of the queue length and two types of delays for each class. The estimates were obtained by averaging over $4,000$ independent replications.

Panel 3a and Panel 3c show that the ratio of the two time-varying mean queue-lengths is indeed stabilized at around 0.31, but the plot in Panel 3b and Panel 3c show that the delay ratio is far being stabilized. The intuition behind Figure 3a - 3c is that how long each arriving customer waits in queue depends on the *future*. To elaborate, consider a class 1 customer who arrives at time t at which the arrival rate of class 1 is decreasing while the arrival rate of the other class is increasing. Then, from this time onwards, the queue of class 2 tends to build up more rapidly than the other queue. On the other hand, because the FQR control strives to maintain a fixed queue ratio, the system inevitably favors class 2 over class 1, i.e., it admits class 2 customers more frequently. As a consequence, any new arrival to queue 1 is worse off in terms of their waiting time in queue.

Now consider the same model but with the HLDR control where the target delay-ratio v is set at $1/2$. Figure 3d - 3e report the simulation estimates of the average queue-length and the head-of-line delay for each class. Again, these estimates were acquired by averaging over $4,000$ sample paths over the time horizon $[0, T]$.

Panel 3d and Panel 3f show that the two average queue-lengths often change in the opposite direction, but the plot on the right in Figure 3e and Figure 3f show that the delay-ratio is stabilized remarkable well. Moreover, the system achieves differentiated service without exploiting information about time-varying arrival rate functions.

As discussed in the paper, an alternative to achieving the desired delay-ratio $1/2$ is to use the TVQR control. Figure 3g - 3h display the mean queue-lengths and the mean head-of-line delays with the TVQR control. These two plots exhibit similar patterns of Figure 3d - 3e under the HLDR control. But a closer look at Figure 3i shows the TVQR rule is less effective in stabilizing the delay ratio.

2.1.2 Impact of Customer Abandonment

Comparing Figure 2 and Figure 3 with Figure 1, we observe that customer abandonments have a significant impact on the performance of different scheduling rules. Indeed, the both the queue ratio and the delay move further away from the target/desired ratio as the abandonment rate grows. Moreover, there is fundamental difference between the fluctuations in Figure 1 - 3. We see that these fluctuations are far less with less customer abandonments, as illustrated by Panel (c), (f) and (i) of the three figures. Thus, we conclude that the ratio-control rules are less effective with higher abandonment rate.

2.1.3 Numerical Justification of the HSHT Limit

Here we let the service rates and the abandonment rates be fixed at $\mu_1 = \mu_2 = 1$ and $\theta_1 = \theta_2 = 0.5$, respectively, but let the system size grows. Figure 4 shows the queue and delay ratios as a function of system size with QoS coefficient $c = 0$. Figure 4 suggests that these ratio control rules become more effective as the scale increases, consistent with our MSHT limits developed in the paper.

2.1.4 Sample Path Little's Law

In §4 of the main paper we established a *sample-path (SP) MSHT Little's law (LL)* that is a consequence of the MSHT limits in Theorem 4.1 and Theorem 4.2, which is a generalization of the the SP-MSHT-LL for the stationary model; see e.g., Theorem 4.3 in Gurvich and Whitt (2009). The SP-MSHT-LL states that, for large scale service systems that are running in the QED MSHT regime,

$$Q_i(t) \approx \lambda_i(t)V_i(t) \quad \text{for all} \quad t, \tag{2.1}$$

where $Q_i(t)$ is the queue length, $\lambda_i(t)$ is the arrival-rate function and $V_i(t)$ is the potential delay at time t for class i. In Figure 5, we show the individual samples of the queue length and the actual delay of each class for our base case. In particular, we plot the sample path of $Q_i(\cdot)$ together with the sample path of $\lambda_i(\cdot)w_i(\cdot)$ for $i = 1, 2$, under the three ratio-control rules, namely the FQR, the HLDR and the TVQR policy. Panel (a) and Panel (b) show that, with the FQR rule, the sample paths change over time but the two curves agree closely, with error of small order. Panel (c) and Panel (d) suggest that, with the HLDR rule, the SP-MSHT-LL holds approximately as well. Similarly, Panel (e) and Panel (f) confirms the SP-MSHT-LL with the TVQR rule.

2.2 Exponential Service with Class-Dependent Service Rates

The foregoing experiments have examined the performance of various scheduling policies with homogeneous service times. In most engineering applications, it is desirable to have class-dependent service times. For this part of the experiments, we assume that all service times are exponentially distributed but class-dependent. This is equivalent to assuming class-dependent service rates. In particular, we assume that class-1 and class-2 customers have service rates $\mu_1 = 2/3$ and $\mu_2 = 3/2$ respectively. This may reflect what happens in an emergency department where high acuity patients tend to have a longer length-of-stay (LoS) whereas low acuity patients tend to have a shorter LoS. Figure 6 reports the simulation estimates of the mean queue lengths and the mean delays in a two-class $M_t/M/s_t + M$ model with class-dependent service. The performance under three scheduling policies is remarkably

similar to the case with homogenous service-time distribution. In particular, the queue ratio is well stabilized with the FQR rule while the delay ratio is perfectly stabilized with the HLDR rule.

Paralleling §2.1.4 we show in Figure 7 the individual samples of the queue length and the actual delay of each class for the case of class-dependent service. Consistent with our expectation, these plots strongly support the SP-MSHT-LL derived in §4 of the main paper.

3 A Realistic Example

In this section, we provide additional simulation results for the example considered in §5 of the main paper but focusing on the case with zero abandonment.

Here we assume that $\theta_1 = \theta_2 = 0$. From our analysis, we know that the staffing component reduces to the staffing problem for a single-class $M_t/M/s_t$ model. We can thus apply the modified-offered-load (MOL) staffing algorithm for the $M_t/M/s_t$ model together with HLDR or TVQR scheduling rule to stability the performance measures at the target level.

Figure 8 depicts the potential delays over the time interval $[0, 130]$ for the HLDR rule (left) and the TVQR rule (right). We plot the potential delays for both classes. All estimates were obtained by averaging over 2000 independent replications. Figure 8 shows that it takes significant amount of time for the system to reach its steady state but the potential delay of each class has a tendency to approach the associated target.

Figure 9 plots the tail probabilities over the time interval $[0, 130]$ for the HLDR rule (plots at the top) and the TVQR rule (plots at the bottom). Here we assume that the target tail probability $\alpha = 0.5$. We plot the tail probabilities for both classes. All estimates were obtained by averaging over 2000 independent replications. We observe that with zero abandonment performance stabilization becomes more difficult. Nonetheless, the tail probabilities have a tendency to approach the associated target.

References

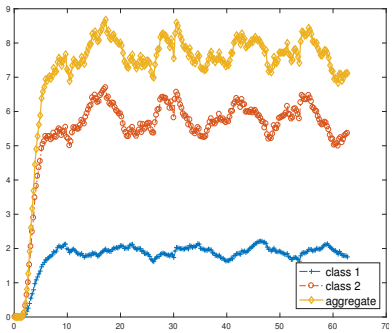
Eick SG, Massey WA, Whitt W (1993) The physics of the $M_t/G/\infty$ queue. *Operations Research* 41(4):731–742.

Gurvich I, Whitt W (2009) Queue-and-idleness-ratio controls in many-server service systems. *Mathematics of Operations Research* 34(2):363–396.

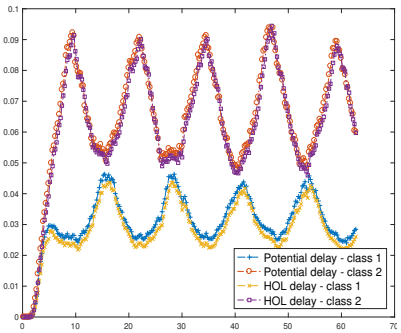
He B, Liu Y, Whitt W (2016) Staffing a service system with non-poisson non-stationary arrivals. *Probability in the Engineering and Informational Sciences* 30(4):593–621.

Ross SM (1990) *A course in simulation* (Prentice Hall PTR).

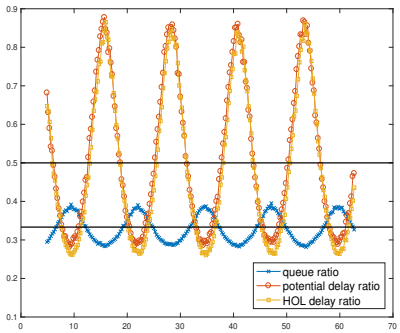
Whitt W, Zhao J (2017) Staffing to stabilizing blocking in loss models with non-Markovian arrivals. *Naval Research Logistics* .



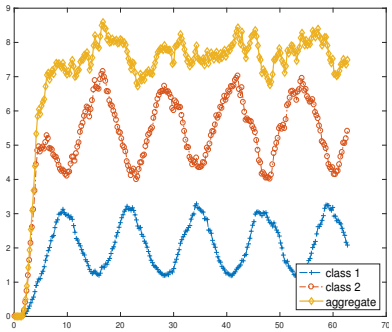
(a) mean queue lengths (FQR)



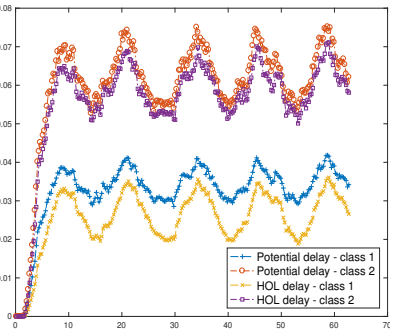
(b) mean delays (FQR)



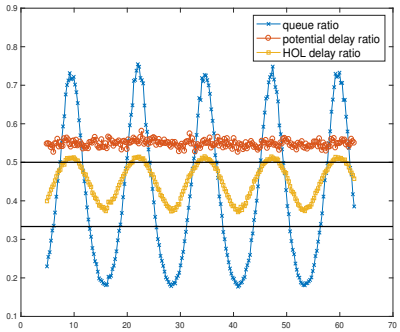
(c) queue & delay ratios (FQR)



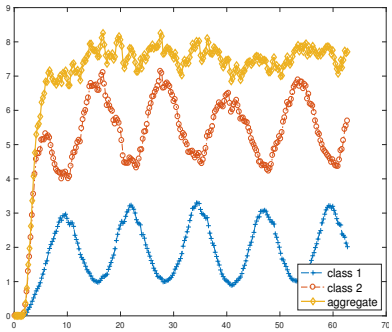
(d) mean queue lengths (HLDR)



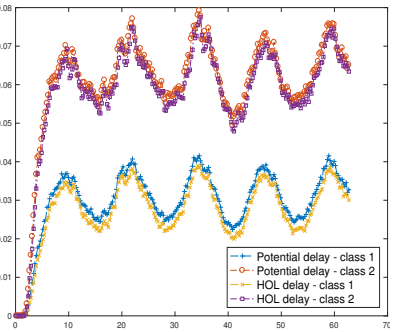
(e) mean delays (HLDR)



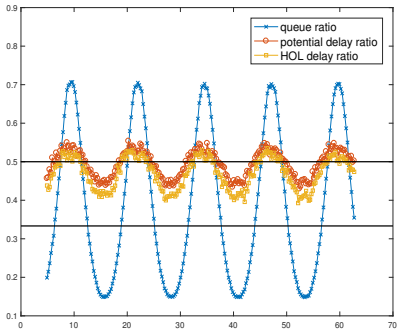
(f) queue & delay ratios (HLDR)



(g) mean queue lengths (TVQR)

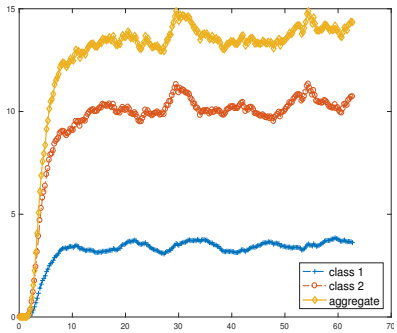


(h) mean delays (TVQR)

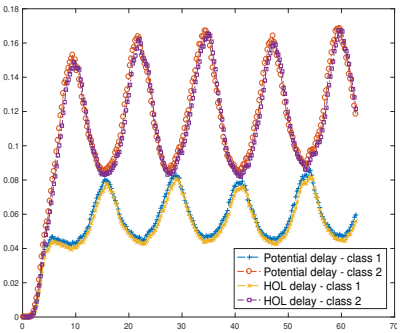


(i) queue & delay ratios (TVQR)

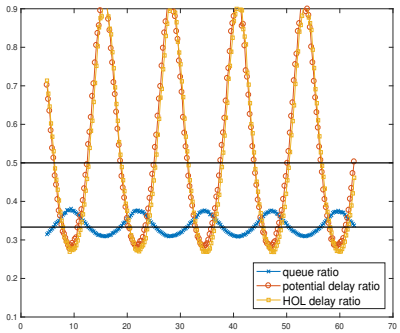
Figure 1: Statistical summaries for a two-class $M_t/M/s_t+M$ queue under three scheduling policies with arrival rate functions $\lambda_1(t) = 60 - 20\sin(t/2)$, $\lambda_2 = 90 + 30\sin(t/2)$, service rate $\mu = 1$, abandonment rate $\theta_1 = \theta_2 = 0.5$ and $\tilde{c} = 0$.



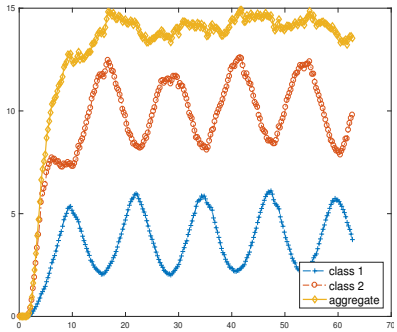
(a) mean queue lengths (FQR)



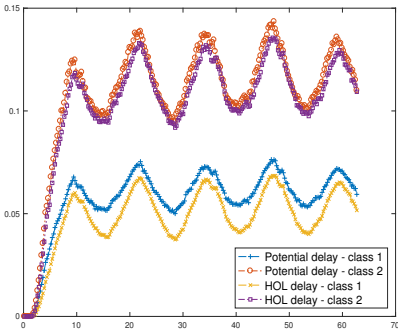
(b) mean delays (FQR)



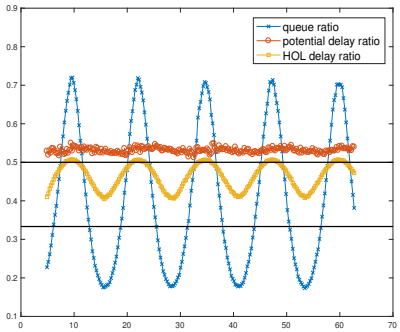
(c) queue & delay ratios (FQR)



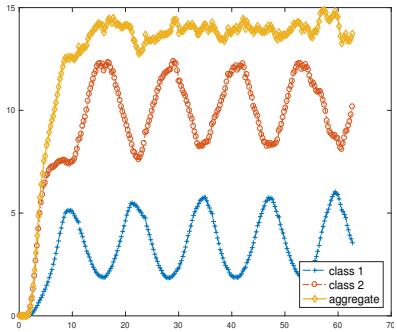
(d) mean queue lengths (HLDR)



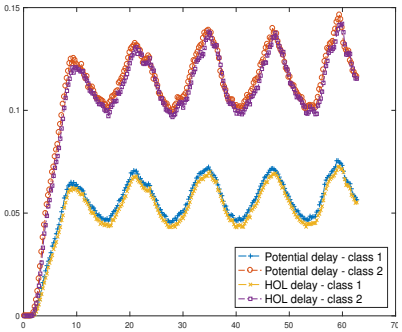
(e) mean delays (HLDR)



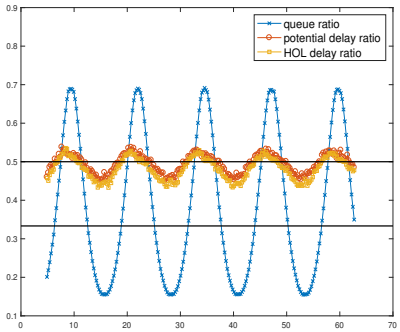
(f) queue & delay ratios (HLDR)



(g) mean queue lengths (TVQR)

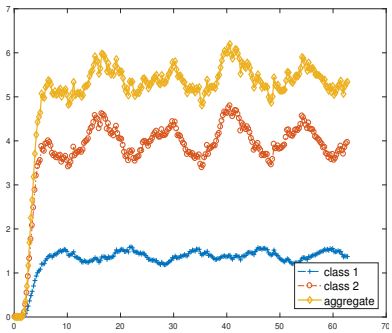


(h) mean delays (TVQR)

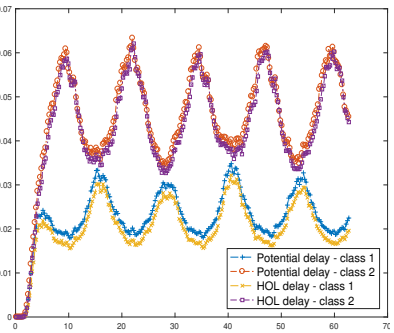


(i) queue & delay ratios (TVQR)

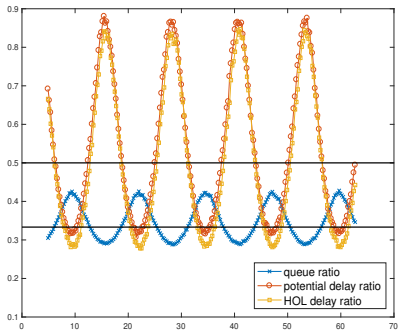
Figure 2: Statistical summaries for a two-class $M_t/M/s_t+M$ queue under three scheduling policies with arrival rate functions $\lambda_1(t) = 60 - 20\sin(t/2)$, $\lambda_2 = 90 + 30\sin(t/2)$, service rate $\mu = 1$, abandonment rate $\theta_1 = \theta_2 = 0.2$ and $\tilde{c} = 0$.



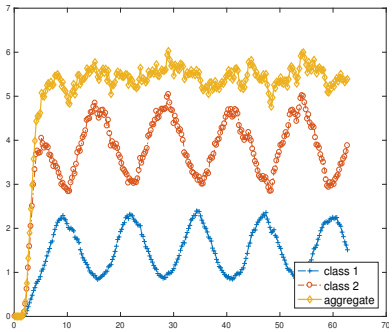
(a) mean queue lengths (FQR)



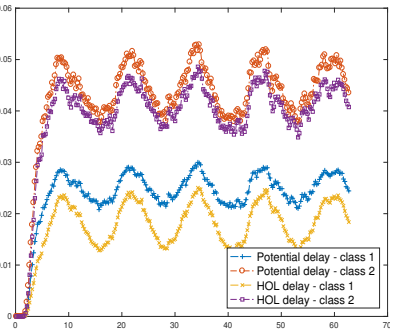
(b) mean delays (FQR)



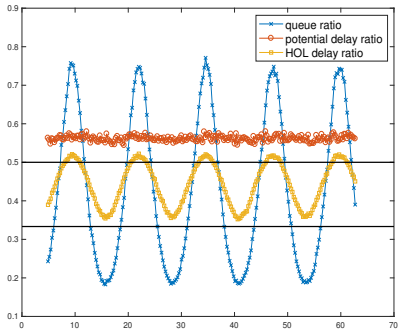
(c) queue & delay ratios (FQR)



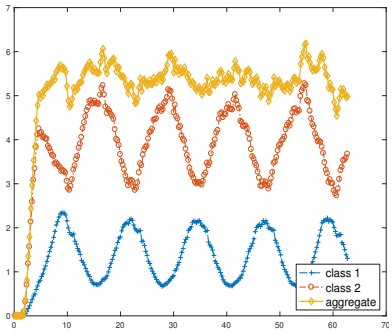
(d) mean queue lengths (HLDR)



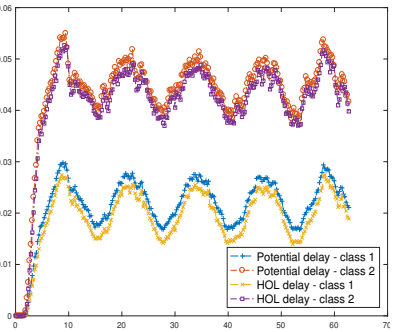
(e) mean delays (HLDR)



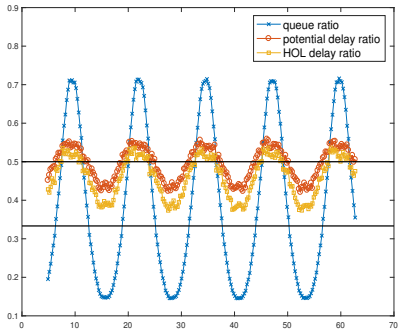
(f) queue & delay ratios (HLDR)



(g) mean queue lengths (TVQR)

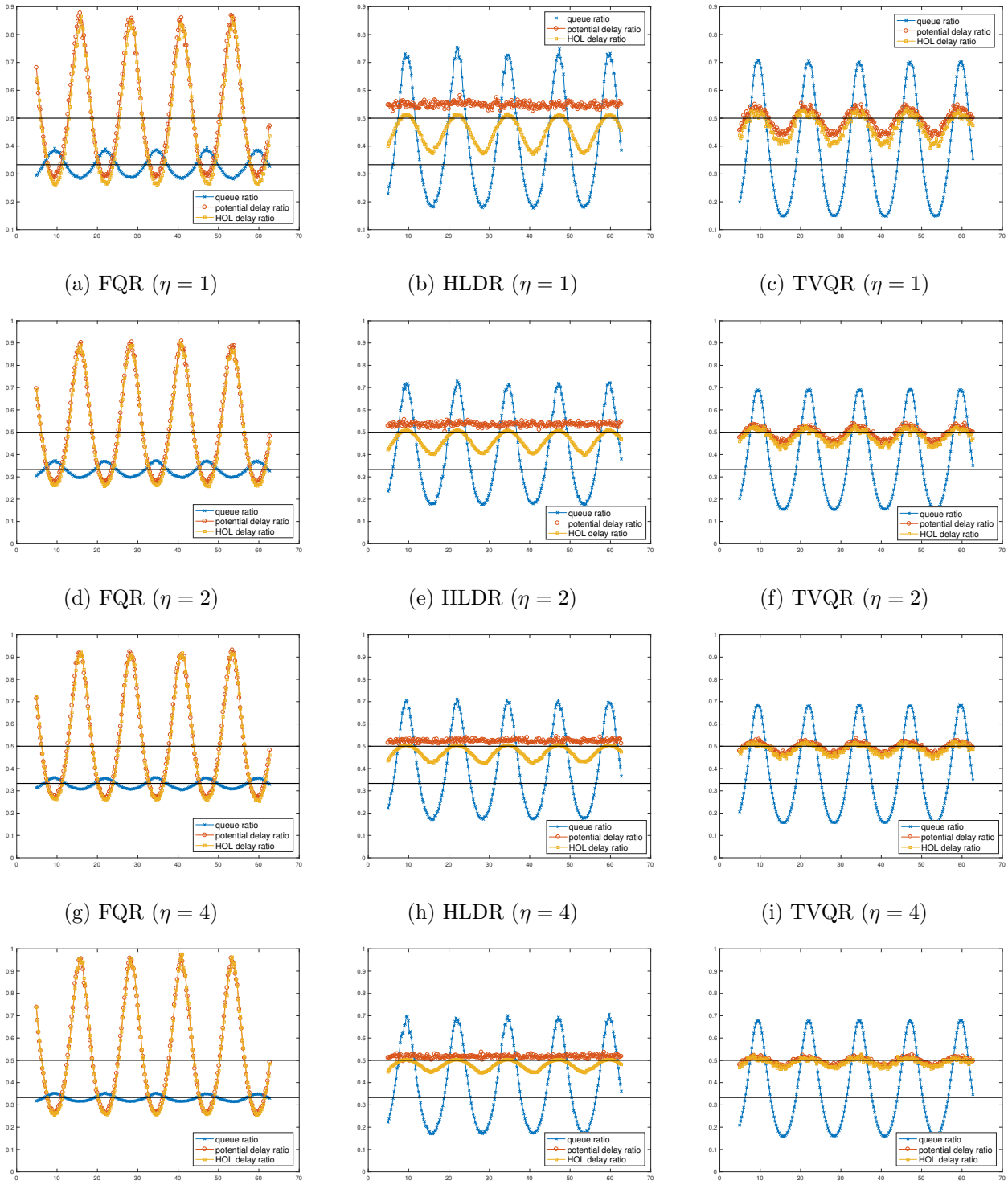


(h) mean delays (TVQR)



(i) queue & delay ratios (TVQR)

Figure 3: Statistical summaries for a two-class $M_t/M/s_t+M$ queue under three scheduling policies with arrival rate functions $\lambda_1(t) = 60 - 20\sin(t/2)$, $\lambda_2 = 90 + 30\sin(t/2)$, service rate $\mu = 1$, abandonment rate $\theta_1 = \theta_2 = 0.8$ and $\tilde{c} = 0$.



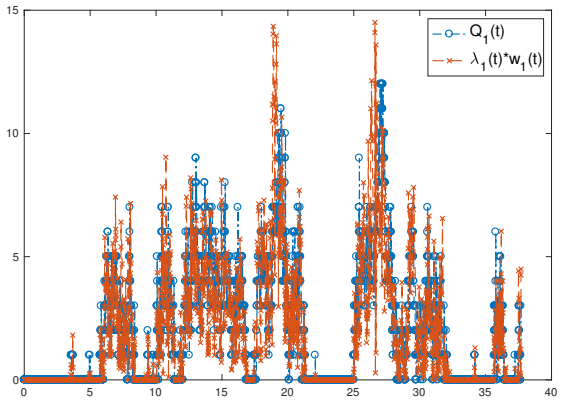
(a) FQR ($\eta = 1$) (b) HLDR ($\eta = 1$) (c) TVQR ($\eta = 1$)

(d) FQR ($\eta = 2$) (e) HLDR ($\eta = 2$) (f) TVQR ($\eta = 2$)

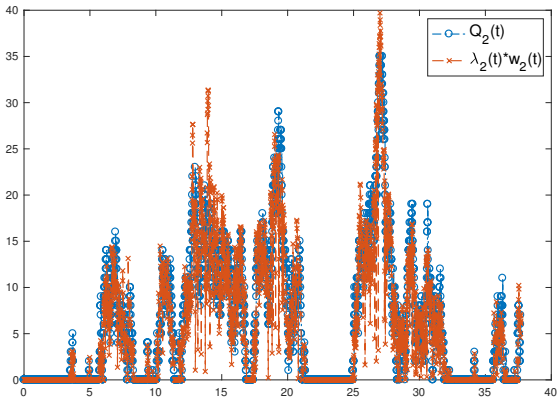
(g) FQR ($\eta = 4$) (h) HLDR ($\eta = 4$) (i) TVQR ($\eta = 4$)

(j) FQR ($\eta = 8$) (k) HLDR ($\eta = 8$) (l) TVQR ($\eta = 8$)

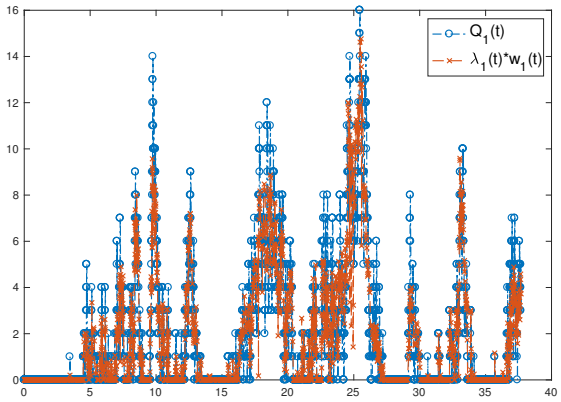
Figure 4: Queue and delay ratios as a function in system size for a two-class $M_t/M/s_t + M$ queue with arrival rate functions $\lambda_1(t) = \eta \cdot (60 - 20\sin(t/2))$, $\lambda_2 = \eta \cdot (90 + 30\sin(t/2))$, service rate $\mu = 1$, abandonment rate $\theta_1 = \theta_2 = 0.5$ and $\tilde{c} = 0$.



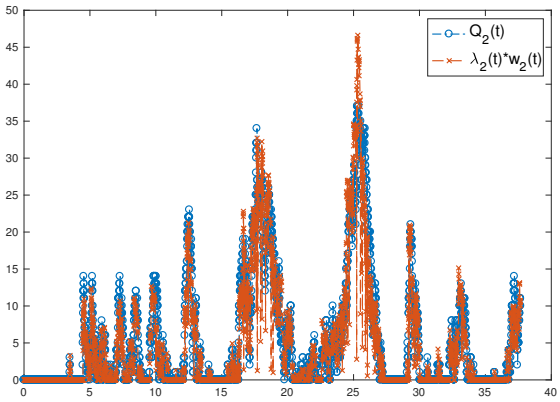
(a) Class 1 (FQR)



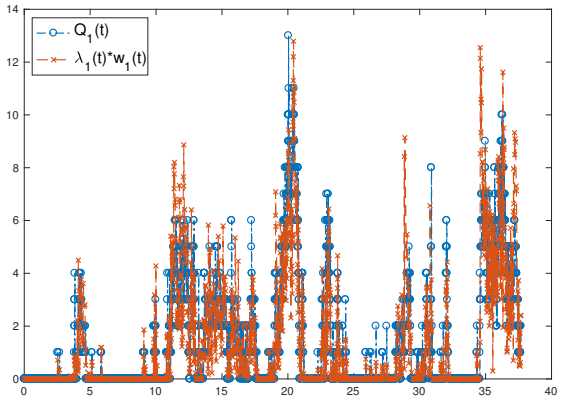
(b) Class 2 (FQR)



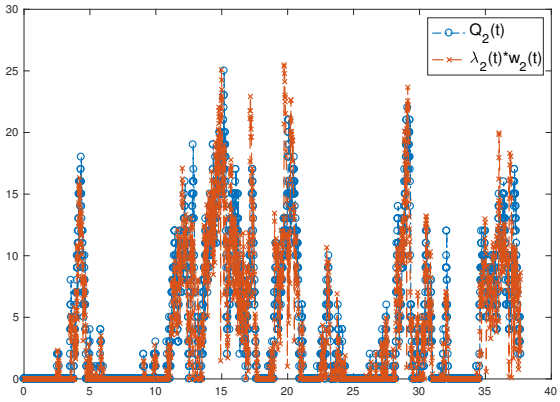
(c) Class 1 (HLDR)



(d) Class 2 (HLDR)

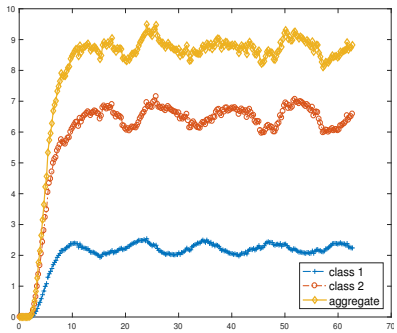


(e) Class 1 (TVQR)

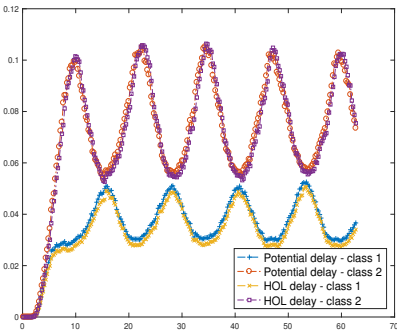


(f) Class 2 (TVQR)

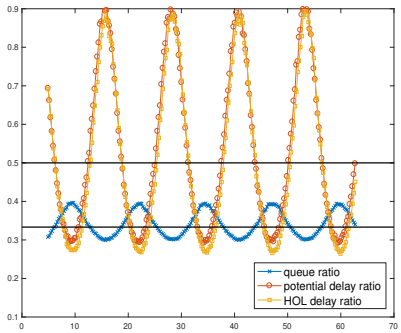
Figure 5: Sample paths of $Q(t) \equiv (Q_1(t), Q_2(t))$ and $w(t) \equiv (w_1(t), w_2(t))$ for the two-class $M_t/M/s_t + M$ model with arrival rate functions $\lambda_1(t) = \eta \cdot (60 - 20\sin(t/2))$, $\lambda_2 = \eta \cdot (90 + 30\sin(t/2))$, service rate $\mu = 1$, abandonment rate $\theta_1 = \theta_2 = 0.5$ and $\tilde{c} = 0$.



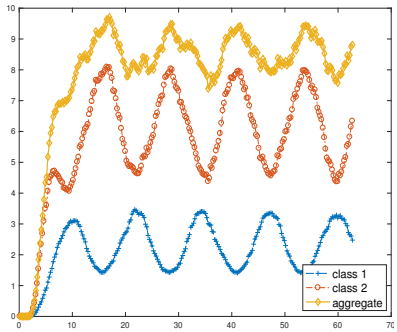
(a) mean queue lengths (FQR)



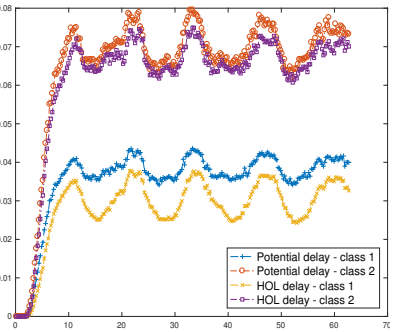
(b) mean delays (FQR)



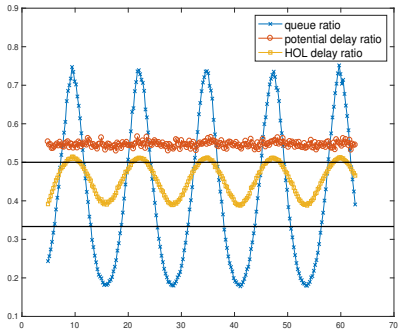
(c) queue & delay ratios (FQR)



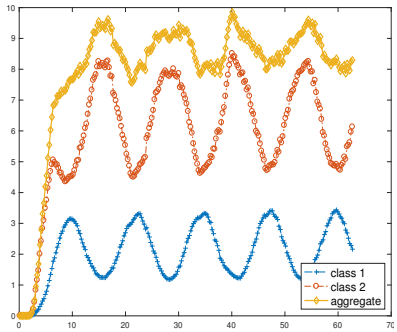
(d) mean queue lengths (HLDR)



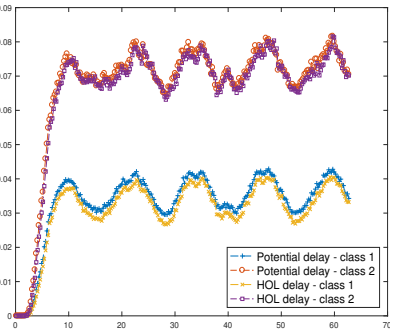
(e) mean delays (HLDR)



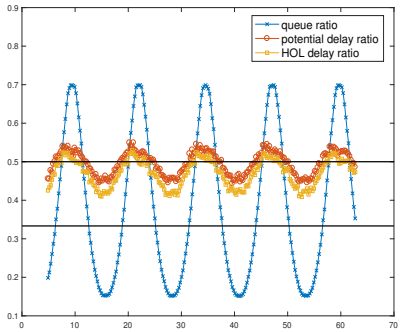
(f) queue & delay ratios (HLDR)



(g) mean queue lengths (TVQR)

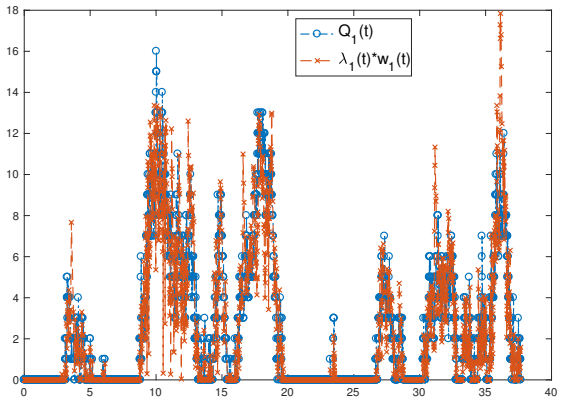


(h) mean delays (TVQR)

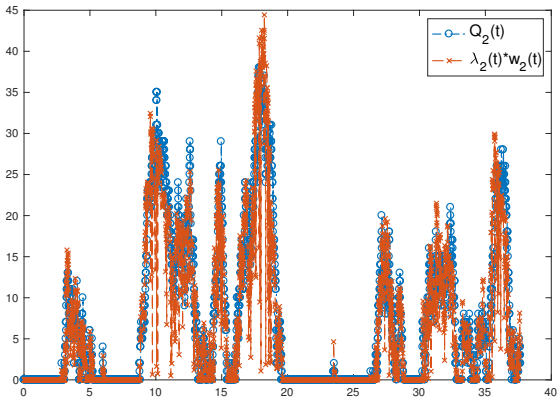


(i) queue & delay ratios (TVQR)

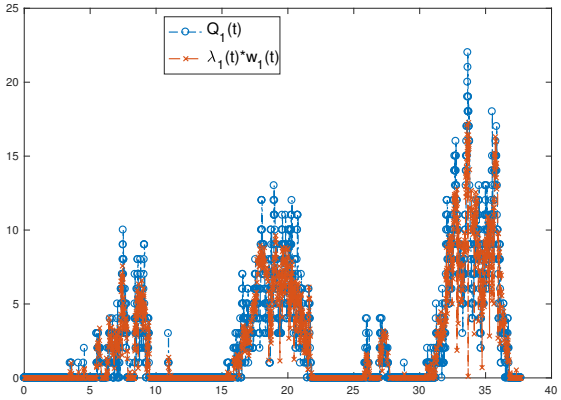
Figure 6: Statistical summaries for a two-class $M_t/M/s_t+M$ queue under three scheduling policies with arrival rate functions $\lambda_1(t) = 60 - 20\sin(t/2)$, $\lambda_2 = 90 + 30\sin(t/2)$, service rates $(\mu_1, \mu_2) = (2/3, 3/2)$, abandonment rate $\theta_1 = \theta_2 = 0.5$ and $\tilde{c} = 0$.



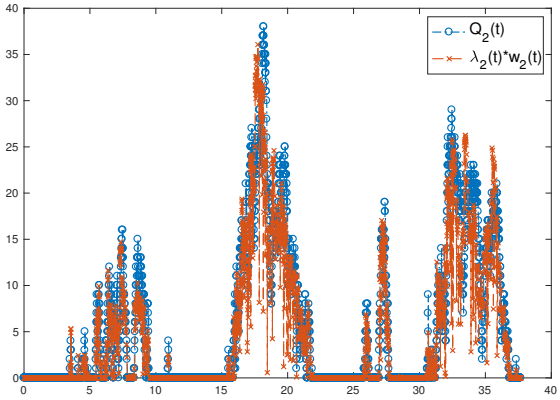
(a) Class 1 (FQR)



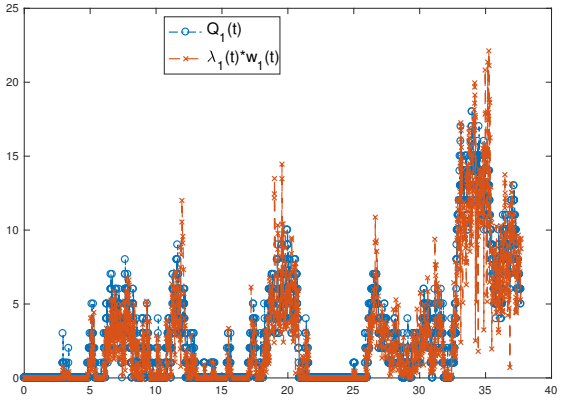
(b) Class 2 (FQR)



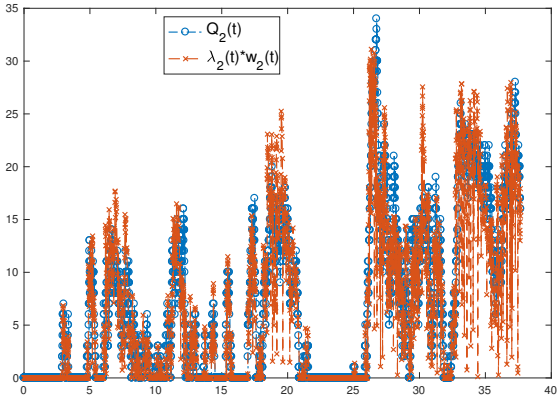
(c) Class 1 (HLDR)



(d) Class 2 (HLDR)

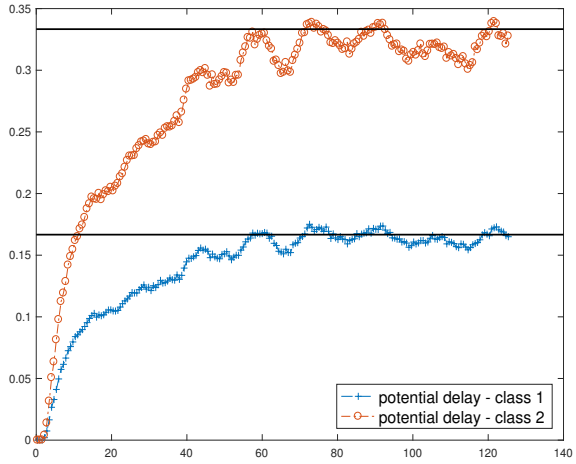


(e) Class 1 (TVQR)

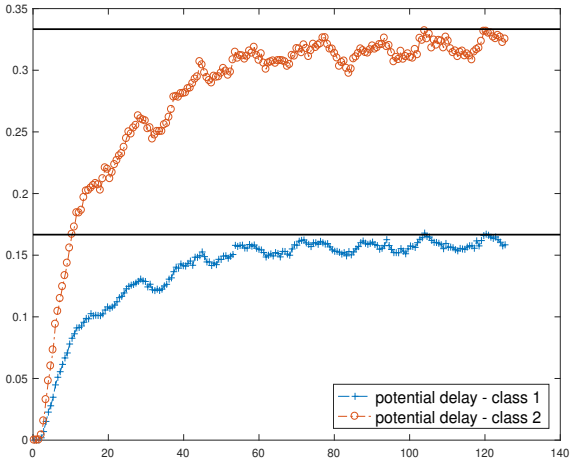


(f) Class 2 (TVQR)

Figure 7: Sample paths of $Q(t) \equiv (Q_1(t), Q_2(t))$ and $w(t) \equiv (w_1(t), w_2(t))$ for the two-class $M_t/M/s_t + M$ model with arrival rate functions $\lambda_1(t) = \eta \cdot (60 - 20\sin(t/2))$, $\lambda_2 = \eta \cdot (90 + 30\sin(t/2))$, service rate $(\mu_1, \mu_2) = (2/3, 3/2)$, abandonment rate $\theta_1 = \theta_2 = 0.5$ and $\tilde{c} = 0$.

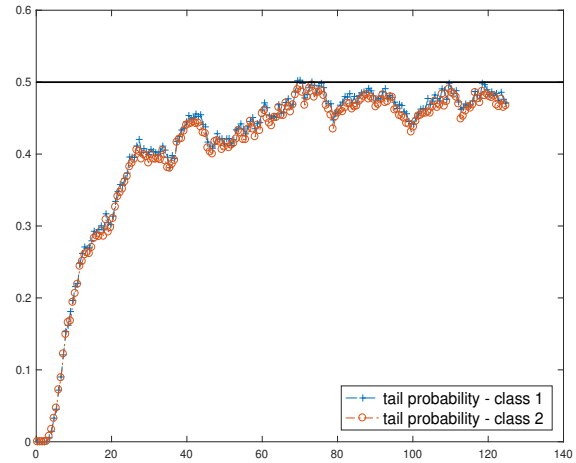


(a) HLDR

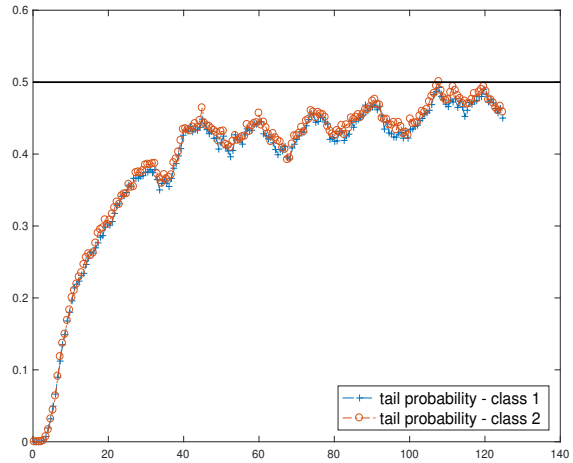


(b) TVQR

Figure 8: Potential delays for a two-class $M_t/M/s_t$ queue with arrival-rate functions $\lambda_1(t) = 60 - 20\sin(2t/5)$, $\lambda_2 = 90 + 30\sin(2t/5)$, common service rate $\mu = 1$ and abandonment rate $\theta = 0$.



(a) HLDR ($\alpha = 0.5$)



(b) TVQR ($\alpha = 0.5$)

Figure 9: Tail probabilities for a two-class $M_t/M/s_t$ queue with arrival-rate functions $\lambda_1(t) = 60 - 20\sin(2t/5)$, $\lambda_2 = 90 + 30\sin(2t/5)$, common service rate $\mu = 1$ and abandonment rate $\theta = 0$.