2006 CCRTS

THE STATE OF THE ART AND THE STATE OF THE PRACTICE

TOPIC: C2 Modeling and Simulation

Modeling Supervisory Control and Team Performance in the Air Defense Warfare Domain with Queueing Theory Part II

Joseph DiVita, Robert Morris, and Glenn Osga

Joseph DiVita
San Diego Systems Center, Space and Naval Warfare
Code 244209
53560 Hull Street
San Diego, CA 92152
Phone 619 553 6461/fax 619 553 3650
Joseph.divita@navy.mil

Abstract

In our previous research, we hypothesize that the performance of a supervisory control operator that must process tasks recommended by a system task manager is analogous to the performance of a vacationing server, $M/E_r/1$ queue. Thus, we assume that the arrival of tasks is Markovian and that service consists of r- stages of processing each of which is exponentially distributed. In addition, we assume that when there are no tasks in the queue to process, the operator "takes a vacation," i.e., goes off and performs other duties. The model assumed vacation time was exponentially distributed. We derived the queueing statistics for this system. These statistics include (1) the average number of customers (tasks) in the queue, (2) the average time a task spends in the queue, and (3) the average waiting time in the queue. We extended this model to a two-class priority $M/E_r/1$ vacationing server system. The results of these predictions were compared to actual operator performance. Our current research generalizes the arrival processes. That is, instead of assuming that the arrival of tasks follows a Poisson process, we assume a Markov-Modulated Poisson Process (MMPP). The MMPP allows for a change in the rate in which tasks arrive to the system. Thus, "rush hour" effects and the ebb and flow of task arrivals may be taken into account by the new model. In the Command and Control environment, it is particularly important to estimate the "rush hour " effect on time critical events. A new set of queueing statistics was generated for a two-class MMPP/M/1 vacationing server system. This allowed us to compare the model to operator performance on the test scenario over extended periods of time.

Approach

The goal of our research program is to develop quantitative models of operator and system performance that will form the basis of a scientific design approach that can be utilized by Combat System Design Engineers. Our approach demonstrates how operator workload may be quantified so that manning requirements may be specified for future Naval Combat Systems.

The increased automation of combat weapon systems is radically changing the role of the human operator from that of controller to supervisor. As a supervisor, the operator is responsible for monitoring and performing multiple tasks. To support the multitasking activity associated with supervisory control, a Task Manager (TM) display is being incorporated into future combat weapon systems such as the Multimodal Watchstation (MMWS) and the Land Attack Combat System (LACS) (see Osga et al., 2002). The TM display represents tasks, in the form of icons on a display screen, that the system has determined actionable given the current tactical information and Rules of Engagement (ROE). The posting of tasks to the TM display for operators to perform is analogous to service calls arriving at a Help Desk or calls to any telephone system. Other examples include "jobs" arriving at a computer processing system and customers waiting in line for service, such as at a bank, a post office, or grocery checkout counter. In supervisory control, we are interested in the flow of tasks (work) through a system that is composed of both human servers and automated servers (computers). Quantitative models and methods that analyze dynamic systems of flow have been developed in the domain of queuing theory (Kleinrock 1975 & 1976; Takagi, 1991).

We have demonstrated (DiVita, Morris & Osga, 2005; DiVita, Morris & Osga, 2004) that the performance of a supervisory control operator that must process tasks recommended by a

system task manager is analogous to the performance of a vacationing server, $M/E_r/1$ queue. Thus, we assumed that the input process was Markovian and that service consisted of r- stages of processing each of which was exponentially distributed (Erlangian r-stage, E_r). In addition, we assumed that when there were no tasks in the queue to process, the operator "took a vacation," i.e., performed other duties. The model assumed vacation time was exponentially distributed. We derived the queueing statistics for this system. These statistics included (1) the average number of customers (tasks) in the queue- N, (2) the average time a task spent in the queue - T, and (3) the average waiting time in the queue -W. We extended this model to a two-class priority $M/E_r/1$ vacationing server system. The results of these predictions were compared to actual operator performance over the first 33 minutes of an air defense warfare (ADW) scenario entitled the Sea of Japan test scenario.

Data from an air defense warfare (ADW) team consisting of four operators were collected from a one hour and forty-five minute ADW scenario entitled the Sea of Japan (SOJ). Data from this scenario were analyzed from the viewpoint of queueing theory. In Table 1 we compare the queueing model predictions with actual data from the Team 1 Air Warfare Coordinator (AWC) over the first 33 minutes of the scenario. This time period was selected because the arrival rate of tasks did not change over this portion of the scenario.

Table 1: M/Er/1 Priority Queueing model prediction compared to observed AWC data for first change point interval [0, 1995.1].

$\lambda_1 = 1/332.52$ $\lambda_2 = 1/48.66$ $\mu_1 = 1/16.72$ $\mu_2 = 1/16.79$ $V = 1/17.73$	**N_1** ***Mean number of Class 1 tasks in system***	**N_2** ***Mean number of Class 2 tasks in system***	**N** ***Mean number of tasks in system***	**T_1** ***Mean total time for Class 1 tasks in system***	**T_2** ***Mean total time for Class 2 tasks in system***	**T** ***Mean total time for a task in system***	**W_1** ***Mean waiting time for Class 1 tasks***	**W_2** ***Mean waiting time for Class 2 tasks***	**W** ***Mean waiting time for a task.***
Predicted	0.096	0.867	0.964	32.077	42.198	40.906	15.362	25.406	24.124
Observed	0.098	0.787	0.884	32.467	39.602	38.651	15.752	22.496	21.616
% Error	1.22	9.28	8.23	1.22	6.15	5.51	2.54	11.45	10.39

The service time to perform tasks was modeled with a r-stage Erlangian distribution. The number of stages was varied to find the best fit between the second moment of the observed data and the expected second moment of the theorized distribution of service times. This optimal value of r was discovered to be 6.

The vacation time was computed as follows: A *vacation period* is defined as starting when a task is completed and there are no more tasks in the queue to perform. A vacation period ends when the operator returns from vacation and resumes performing tasks. Vacation times are determined by subtracting the first arrival time of a task during each vacation period from the end time of that vacation period. By assuming that the distribution of vacation times is exponentially distributed, the "memoryless" property of the exponential allows us to claim that the mean vacation time may be estimated by the average of the vacation times (V = 17.73 seconds).

Generalizing the Arrival Process

During the course of a scenario, the rate at which tasks appear on the TM display may vary. In general, there are busy periods followed by relatively non-busy periods. This arrival process creates a challenge for queueing theory predictions since, tasks "back-up" during periods of high task flow, but then are completed as the flow of tasks subsides. The impact on performance to time critical events in the context of periods of high and low workload is of particular interest in a command and control situation. Evaluating the manning requirements of combat workstations must take into account the change in workflow. During periods of low workload the system may be over-staffed, but during periods of high workload the system runs the risk of being under-staffed. The Markov-modulated Poisson Process (MMPP) (see Hock, 1996 Chapter 8) may capture the ebb and flow of the task arrivals and its impact on the performance of a queueing system. The MMPP is a doubly stochastic process. In our example, the length of the two periods, rush hour and non-rush hour, randomly varies. This leads to a 2x2 transition rate matrix, Q, where the entries r_i represent the rates at which the underling stochastic process changes. For example, r_1 may represent the rate at which the arrival process changes from non-rush hour to rush hour. In general, q_{ij} represents the rate at which the process changes from state i, to j. q_{ii}, represents the rate at which the process does not change and is set equal to $q_{ii} = -\sum_{j=1, j\neq i}^{m} q_{ij}$ where m is the number of states of the process. In our example, there are only two states, thus:

$$Q = \begin{bmatrix} -r_1 & r_1 \\ r_2 & -r_2 \end{bmatrix}$$

For each state, there is associated a Poisson arrival process with arrival rates λ_i. In our example there is a λ_1 and λ_2 associated with the rush hour arrival rate and the non-rush hour arrival rate. From these λ's the matrix Λ is created:

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

Letting X_k equal the time between the (k-1) and k^{th} arrival and J_k the state of the Markov process at the time of the kth arrival, then it may be shown that the transition probability distribution matrix, F(x), is given by:

$$F(x) = \{I - \exp((Q - \Lambda)x)\}(\Lambda - Q)^{-1}\Lambda$$

where I is the identity matrix and Q and Λ are the matrices described above. The elements of the matrix $F_{ij}(x)$ represent the conditional probabilities:

$$F_{ij}(x) = P\{J_k = j, X_k \le x \mid J_{k-1} = i)$$

Thus $F_{ij}(x)$ represents the probability that the inter-arrival time between the k and k-1 arrival is less than or equal to x, given that that underlying process was in state i for the k-1 arrival and now is in state j for the kth arrival. The transition probability matrix may be derived by differentiating F(x). Thus:

$$f(x) = \frac{d}{dx}F(x) = e^{\{Q-\Lambda)x}\Lambda$$

Taking the Laplace transform of f(x) we obtain:

$$L[f(x)] = (sI - Q + \Lambda)^{-1}\Lambda$$

In matrix notation we have:

$$L\{f(x)] = \left[\begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} - \begin{bmatrix} -r_1 & r_1 \\ r_2 & -r_2 \end{bmatrix} + \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}\right]^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

$$= \frac{1}{\det A}\begin{bmatrix} (s + r_2 + \lambda_2)\lambda_1 & \lambda_2 r_1 \\ \lambda_1 r_2 & (s + r_1 + \lambda_1)\lambda_2 \end{bmatrix}$$

Where det A = $(s+r_1+\lambda_1)(s+r_2+\lambda_2)-r_1r_2$.

Thus the Laplace transform of the unconditional inter-arrival times is:

$$L[X] = \varphi\left(\frac{1}{\det A}\begin{bmatrix} (s + r_2 + \lambda_2)\lambda_1 & \lambda_2 r_1 \\ \lambda_1 r_2 & (s + r_1 + \lambda_1)\lambda_2 \end{bmatrix}\right)\begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$= \frac{s(r_1\lambda_2^2 + r_2\lambda_1^2) + (r_1\lambda_2 + r_2\lambda_1)(r_1\lambda_2 + \lambda_1\lambda_2 + r_2\lambda_1)}{(r_1\lambda_2 + r_2\lambda_1)(s^2 + r_1 + r_2 + \lambda_1 + \lambda_2)(r_1\lambda_2 + \lambda_1\lambda_2 + r_2\lambda_1)}$$

It may be shown that the Laplacian transform for the probability density function (pdf) of inter-arrival times for a generalized arrival function, A(x), must satisfy the equation:

$$\sigma = A^*(m\mu - m\mu\sigma)$$

Where, A*( ) is the Laplacian of the arrival function, A, evaluated at s = $m\mu-m\mu\sigma$, m is the number of servers, and σ is the limit of the ratio of the number of times we find the system in state E_{k+1} to the number of times we find the system in the state E_k. The state E_k signifies a new arrival to the queue finding k customers in the queue. Since in our case m =1, we set σ equal to L[X] and substitute $\mu-\mu\sigma$ for s in the right-hand-side of the equation for L[X] and then we solve

for σ. In the above 2 x 2 example, we are left with solving a cubic equation and must chose the value of σ that is less than 1. The purpose of deriving this σ is that the waiting time, W, for a generalized arrival process may be expressed in terms of this σ, that is:

$$W = \frac{\sigma}{\mu(1-\sigma)}$$

From W, the average waiting time, T, may easily be determined. In order to determine the average number of customers, N, in the queue, we apply Little's Theorem, where λ is derived to be:

$$\lambda = \frac{\lambda_1 r_2 + \lambda_2 r_1}{r_1 + r_2}$$

Thus a set of queueing statistics may be derived for the MMPP/M/1 queue.

Results

In Figure 1, the running average of the arrival rate of tasks is presented. A change point analysis (Chen & Gupta, 2000) was calculated on the inter-arrival times of tasks to determine changes in the arrival rate. This analysis revealed 3 periods of relatively heavy task flow that lasted, 524.1 sec (8.73 min), 375.7 sec (6.26 min), 376.6 sec (6.28 min). During these periods, the average inter-arrival time of tasks was 15.4 sec, 20.9 sec, and 16.4 sec. Likewise, there were 3 slow periods that lasted 1995.1 sec (33.25 min), 1381.3 sec (23.02 min), and 1656.9 sec (27.62 min). The average inter-arrival time for tasks during these periods was 42.4 sec, 125.6 sec and 61.4 sec. From the inter-arrival times, the average rate of task arrivals, λ_1 and λ_2, can be computed for the slow and busy periods. We assumed that task arrival during these periods was Poisson (with different rates). The average durations for these periods were also assumed to be exponentially distributed. r_1 and r_2 represent the average time spent in low and high activity period, respectively.

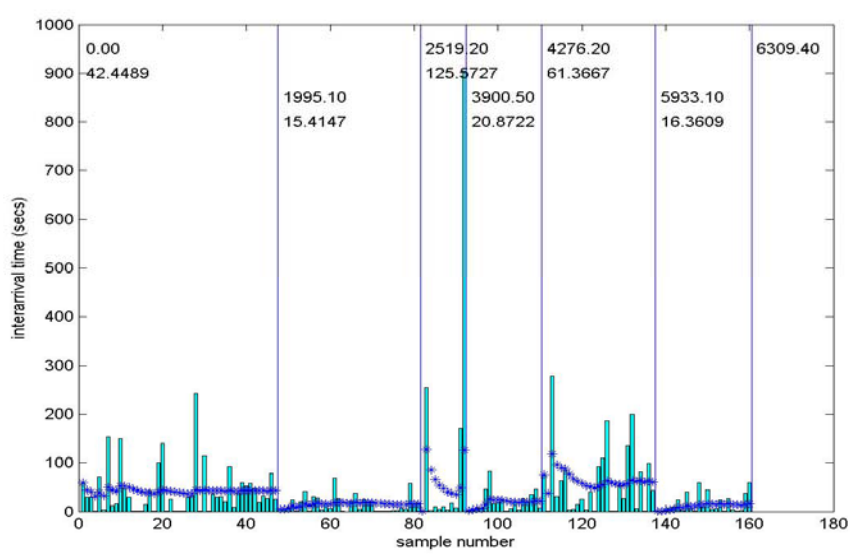

Figure 1: Change point analysis based on the inter-arrival time between AWC tasks for the entire scenario [0, 6309.4]. Asterisks represent the running average.

In Table 2, the results of the M/M/1 queueing predictions are compared to the actual data for the Team 1 AWC operator for the entire scenario. The server was assumed to vacationing. The percent error runs in the 15% - 25% range for the three key statistics – mean number in the system -N, mean total time -T, and mean waiting time in the system -W. Task prioritization was removed from this analysis. The M/M/1 underestimates these statistics. This is because the M/M/1 cannot handle the way in which the tasks pile-up during a burst of rush hour activity.

$\lambda = 1/39.43$ $\mu = 1/17.50$ $V = 1/17.73$	**N** ***Mean number of tasks in the system***	**T** ***Mean total time for a task in the system***	**W** ***Mean waiting time for a task.***
Predicted	1.247	49.190	31.690
Observed	1.443	57.256	39.708
% Error	15.67	16.40	25.30

Table 2: M/M/1 Queueing model prediction compared to observed AWC data for the entire scenario [0, 6309.4], λ represents overall λ.

In Table 3, the results of the MMPP/M/1 queueing predictions are compared to the actual data. The server was assumed to be vacationing. The percent error between observed and predicted data now drops down to the 4%- 8% range, and the effect on performance due to the ebb and flow task arrival rate is better predicted.

$\lambda_1 = 1/59.22$ $\lambda_2 = 1/17.01$ $r_1 = 1/1677.77$ $r_2 = 1/425.37$ $\mu = 1/17.50$ $V = 1/17.73$	**N** ***Mean number of tasks in the system***	**T** ***Mean total time for a task in the system***	**W** ***Mean waiting time for a task.***
Predicted	1.376	54.245	36.745
Observed	1.443	57.256	39.708
% Error	4.89	5.55	8.06

Table 3: Two state MMPP/M/1 Queueing model predictions compared to observed AWC data for the entire scenario [0, 6309.4].

The MMPP may also be used to handle correlated arrivals of tasks. For example, in our test there were a total of 4 operators that made up the ADW team. In the course of the scenario

we observed correlations between tasks that appeared on the Task Manager display, that is, tasks arriving from "outside" the queueing network, and tasks passed between operators – task arriving from "inside" the network. Unfortunately, the total number of tasks for the auxiliary operators was rather small. (The AWC was the main "player" in this scenario). In order to aid our modeling we simulated task arrival and the passing of tasks between operators. Completion rates of tasks was also simulated. In Table 4 we present the results of comparing the predictions for an M/M/1 queue to that of a Matlab simulation of 100,000 tasks presented to the Information Quality Control-1

$\lambda_{outside} = \lambda_{inside} = 1/152.4$ $\mu = 1/15.0$ $V = 1/13.3$	**N** ***Mean number of tasks in the system***	**T** ***Mean total time for a task in the system***	**W** ***Mean waiting time for a task***
Predicted Queueing	0.420	31.976	16.976
Observed IQC1	0.455	34.464	19.430
% Error	8.41	7.78	14.45

Table 4: M/M/1 Queueing model predictions compared to observed IQC1 correlated arrival simulation.

Operator (IQC1) in our queueing network. As can be seen, the predictions are rather poor considering the large number of trials. This is because, in addition to tasks presented on the TM display for the IQC1 ("outside tasks"), he is also passed tasks to do from the AWC ("inside tasks"). There is correlation between the outside tasks appearing on the TM display and the inside passed tasks that causes the tasks to "pile-up" in a manner analogous to the rush hour effect demonstrated in Tables 2 and 3. In order to handle this arrival process an MMPP was applied to the data and the results are listed in Table 5. The MMPP model handles the correlation among the arriving task as is demonstrated by the models accurate predictions.

$\lambda_{outside} = \lambda_{inside} = 1/152.4$ $\mu = 1/15.0$ $V = 1/13.3$	**N** ***Mean number of tasks in the system***	**T** ***Mean total time for a task in the system***	**W** ***Mean waiting time for a task***
Predicted Queueing	0.455	34.478	19.478
Observed IQC1	0.455	34.464	19.430
% Error	0.03	0.04	0.25

Table 5: MMPP/M/1 Queueing model predictions compared to observed IQC1 correlated arrival simulation.

Conclusion

The Markov-Modulated Poisson process (MMPP) generalizes the arrival process of tasks to the operator. In this manner, the slow down in through put can be effectively measured when

a burst of tasks arrive that require the operator to take action. The MMPP/M/1 vacationing server predicts operator performance better than the simpler M/M/1 vacationing server under conditions where task flow varies. This predictive capability is vital to predicting performance to time critical tasks in CIC workstations.

Future research will focus on the following topics: 1) the MMPP models can be extended to handle task priority. 2) the service process may be generalized to be Erlangian, thus predictions for a MMPP/Er/1 need to be generated, 3) the probability distribution of the time a critical task may spend in the system due to a back-log of tasks awaiting service may be calculated. These estimates may then be evaluated by system experts to see if they fall within an acceptable range.

References

Chen, J. & Gupta, A.K. (2000) Parametric Change Point Analysis. Birkhauser: Boston.

DiVita, J., Osga, G., & Morris, R. (2004) Modeling Team Performance In the Air Defense Warfare (ADW) Domain. Command and Control Research and Technology Symposium, San Diego, CA.

DiVita, J & Morris, R. (2005). The Vactioning Server: Queueing Models for Supervisory Control. Space and Naval Warfare Report, San Diego Systems Center (In press).

Hock, N.C. (1996). Queuing Modeling Fundamentals. John Wiley and Sons. New York.

KleinRock, L. (1975). Queueing Systems, Volume I: Theory; Wiley-Interscience, New York.

Osga, G., Van Orden, K., Campbell, N., Kellmeyer, D., & Lulu, D. (2002) Design and Evaluation of Warfighter Task Support Methods in a Multi-Modal Watchstation. Space & Naval Warfare Center San Diego Tech Report 1874.

Takagi, H. (1991). Queueing Analysis. A Foundation of Performance Evaluation, Volume 1: Vacation and Priority Systems, Part 1. Elsevier Science Publishing Company Inc., New York.