Technical Report 2030
September 2013

# A Queueing Model for Supervisory Control of Unmanned Autonomous Vehicles

Joseph DiVita, PhD

Robert L. Morris

Maria Olinda Rodas

**SSC Pacific**



Space and Naval Warfare Systems Center Pacific
San Diego, CA 92152-5001

**SSC Pacific**
**San Diego, California 92152-5001**

| | |
|---|---|
| **J. J. Beel, CAPT, USN** | **C. A. Keeney** |
| **Commanding Officer** | **Executive Director** |


# ADMINISTRATIVE INFORMATION

This report was prepared by the User-Centered Design and Engineering Branch (Code 52621) of the C2 Technology and Experimentation Division (Code 536), Space and Naval Warfare Systems Center Pacific (SSC Pacific.)

| | |
|---|---|
| Released by | Under authority of |
| Deborah Gill-Hesselgrave, Head | Tom Tiernan, Head |
| User-Centered Design and Engineering Branch | C2 Technology and Experimentation Division |

# CONTENTS



# FIGURES

## TABLES

# 1. INTRODUCTION

At Space and Naval Warfare Systems Center Pacific, our research programs have developed quantitative models of operator and system performance that form the basis of a scientific design approach that can be utilized by future Combat System Design Engineers (DiVita, Morris, & Osga, 2004; DiVita, Morris, & Osga, 2005; DiVita, Morris, & Nguyen, 2005; DiVita, Morris, & Osga, 2006). These models were developed for Air Defense and Land Attack combat systems, and were incorporated into prototypes of the future Multimodal Watchstation (MMWS) and Land Attack Weapons Systems (LAWCS) (Osga, Van Orden, Campbell, Kellmeyer, & Lulue, 2002). Several projects (e. g., MMWS, LAWCS, and Combat Supervisory Support Systems) have demonstrated tools that form the foundation for further development of interface concepts that will enable operators to plan and execute complex tasks within dynamic and multiple warfare areas. Our research was designed to answer the growing need to model interface concepts so that future interface designs may evolve in a principled and systematic fashion.

One critical aspect in developing a quantitative model of operator and system performance has been to adopt a task centric approach to interface design that entails an explicit representation, of actions – tasks – that need to be performed by the operator. The representation of work in terms of a task serves as a trace in the system that enables designers to track workload in addition to the task progress and flow of tasks among team members. In supervisory control, we are interested in the flow of tasks (work) through a system that is composed of human servers and automated servers (software agents). Quantitative models and methods that analyze dynamic systems of flow have been developed in the domain of queueing theory. At SSC Pacific, we have applied queueing theory to teams of air defense warfare fighters (DiVita, Morris, & Osga, 2004; DiVita, Morris, & Osga, 2005; DiVita, Morris, & Nguyen, 2005; DiVita, Morris, & Osga, 2006; DiVita, Morris, & Osga, 2007 ) This research may be extended to include supervisory control of unmanned vehicles (UVs). In this analysis, the “customers” to the queue are tasks that must be serviced by the human and software servers. The servers are human operators, software agents, and UVs.

Although the appropriate operator role in supervising autonomous systems has yet to be determined, it is clear that the prevalence of autonomous systems is in an upward trend for the foreseeable future. While some military planners envision truly autonomous systems, most understand that warfighters need to be able to command and control these systems at some level. The future of human factors challenges has undoubtedly shifted from manually controlling individual unmanned systems, to supervisory command of multiple semi-autonomous systems (Kellmeyer & DiVita, 2012). For example, a futuristic Network Centric Operations (NCO) mission scenario is one in which a group of heterogeneous UVs are supervised by a single operator using NCO technology (Rodas, Veronda, & Szatkowski, 2011; Rodas, Szatowski, & Veronda, 2011). In addition to physical platforms, autonomous agents working as virtual team members are prevalent tools for accomplishing naval missions in these NCO scenarios. The coordination of actions and interactions between unmanned autonomous systems, manned systems, and a command group will be essential to accomplishing these future missions (Kellmeyer & DiVita, 2012). In this type of complex command and control (C2 scenario, UV operators will be subjected to vast amounts of information. Therefore, this theory of warfare brings with it a new problem that must be addressed: How to maintain an adequate workload to avoid information overload and resulting loss of situation awareness. It is critical that designers develop predictive models of human and system performance to evaluate the adequacy of a system’s design to satisfy specific mission requirements and ensure adequate operator performance (Rodas, Veronda, & Szatkowski, 2011).

In order to explore the operator’s role in supervising UVs, SSC Pacific acquired the Research Environment for Supervisory Control of Heterogeneous Unmanned Vehicles (RESCHU) developed by the Massachusetts Institute of Technology (MIT). The RESCHU simulator (Nehme, 2009) was developed to test supervisory control tasks such as surveillance and identification. This simulation was modified by adding: 1) a complex mission scenario with an asset to protect and multiple simultaneous enemies to attack, 2) a highly automated system such as mission definition language (MDL), and 3) a highly heterogeneous team that is made of at least three different types of UVs. The new version of the simulation is called RESCHU SP and details of the modification are discussed in Rodas, Veronda, & Szatkowski, 2011.

In the RESCHU SP scenario, a single operator supervises a team of unmanned air vehicles (UAVs), unmanned surface vehicles (USVs), and unmanned underwater vehicles (UUVs). The operator’s task is to deploy the UVs to identify and destroy enemy contacts that are attacking an oil platform at sea. In addition to defending the oil platform, the operator must direct the UVs away from hazardous areas that cause damage to the UVs. The interface for the simulation is depicted in Figure 1. The operator’s display is composed of two main windows placed side by side. In one window, a geo-situational display depicts the spatial position of the UV assets as well as the oil rig, unknown contacts, enemy contacts, and the hazard areas. The second adjacent window is a three-tabbed pane window that contains the following displays: Vehicle Information, The Collaborative Sensing Language (CSL) Editing Controls, and a Payload View.

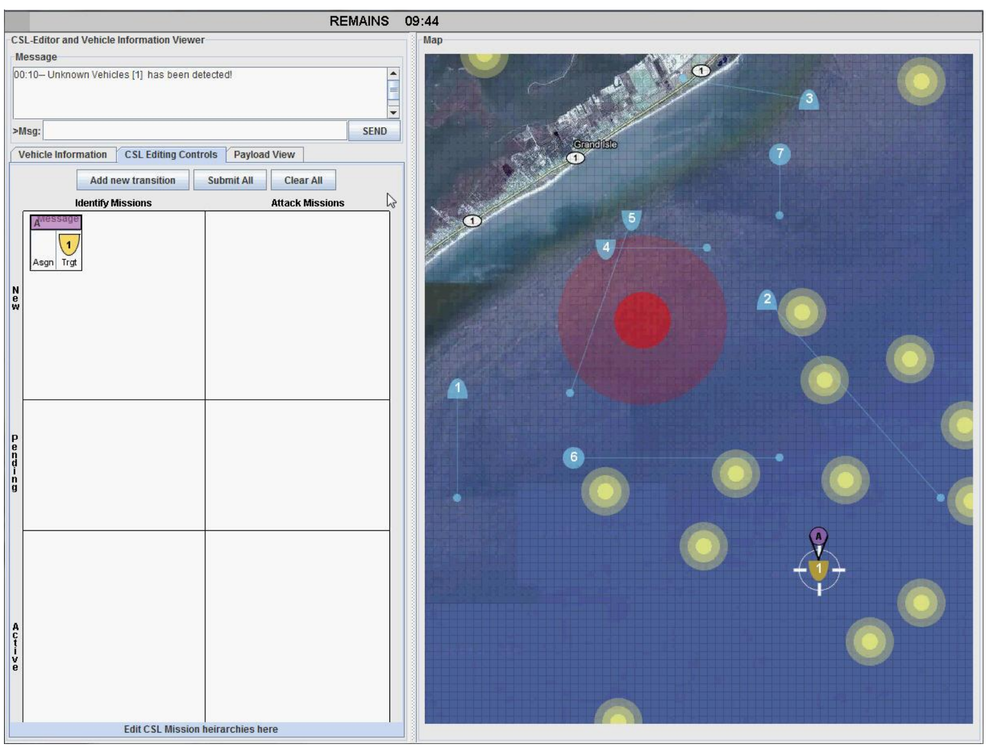


Figure 1. RESCHU SP interface.

In order to protect the oil rig and the UV assets, the operator must engage in five tasks: 1) Assign to identify an unknown contact, 2) Engage to identify an unknown contact, 3) Assign to attack an enemy, 4) Engage to attack an enemy contact, and 5) Hazard avoidance. Figure 2 illustrates the assign to identify task. The scenario is designed such that only USVs can identify unknown contacts and only UUVs can attack enemy contacts. The UAVs fly in predetermined flight patterns that the operator can change to avoid hazardous areas.

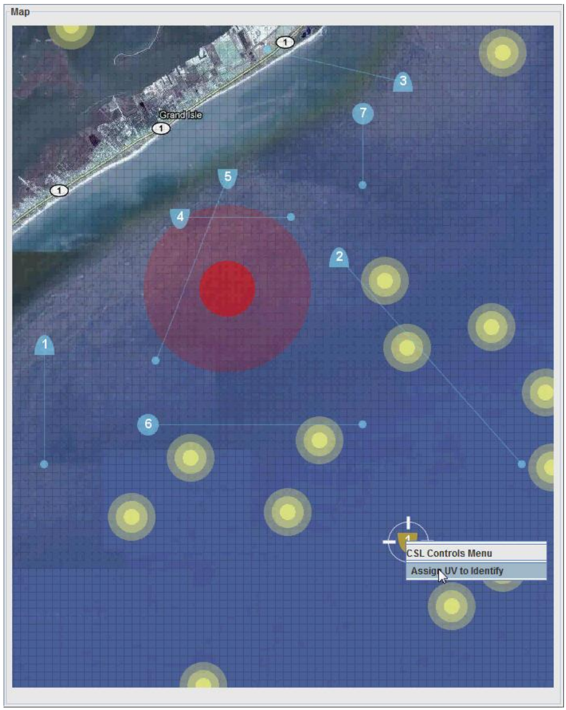


Figure 2. Assign to identify task.

The operator must first select an unidentified contact, symbolized by a yellow icon on the geo-situational display, and then assign a USV to identify it (Figure 2). In the CSL editing control pane, the operator may hit the "Submit All" button to automatically assign the contact an available USV (see Figure 1). Alternatively, the operator may directly assign a USV to identify the unknown contact by dragging the USV's goal point onto the unknown contact. Once the USV arrives at the unknown contact's location, symbolized by a flashing dark yellow icon, the operator may engage the USV to identify the contact (Figure 3).

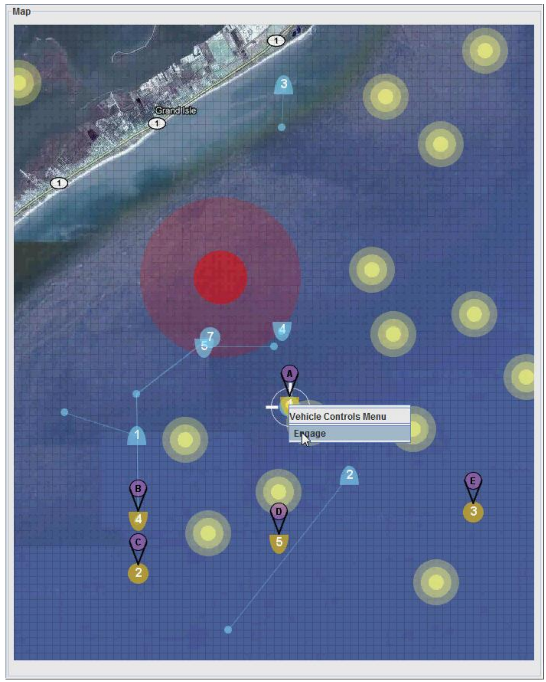


Figure 3. Engage to identify task.

This action brings up a picture in the Payload view that allows the operator to visually identify the contact as a friend or foe (Figure 4). Once identified as an enemy, the contact may be assigned a UUV to attack it.


Figure 4. Image data supplied by USV.

The sequence of actions to attack is analogous to the identification process. An enemy contact is selected to be assigned a UUV for attack. Notice that an enemy contact appears red in the map to indicate that it was identified as the enemy (Figure 5).

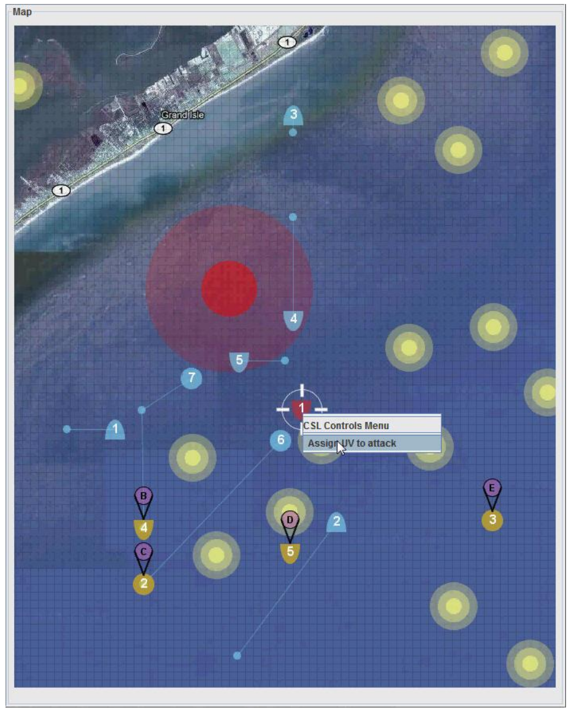


Figure 5. Assign to attack task.

In the CSL editing control panel, the operator hits "Submit All" to automatically assign a UUV for attack. Again, the operator may manually assign a UUV to attack the contact; however, this action is not encouraged in the simulation tutorial. Once the UUV has arrived at the target's location, the red enemy icon flashes and the operator may select the icon to be engaged to attack (Figure 6). Once engaged, the enemy is attacked and eliminated. Its icon disappears from the geo-situational display.

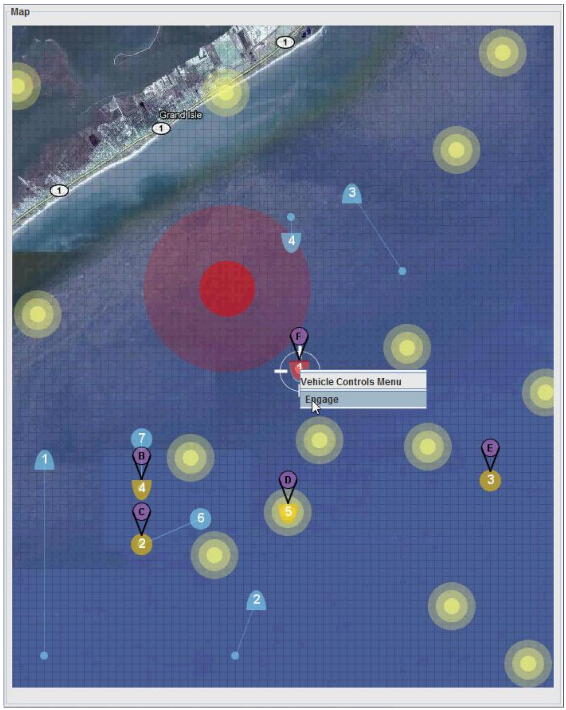


Figure 6. Engage to attack task.

Lastly, in Figure 7, a hazard avoidance task is depicted. The operator manually changes the UV's goal-point to avoid the hazard zone.

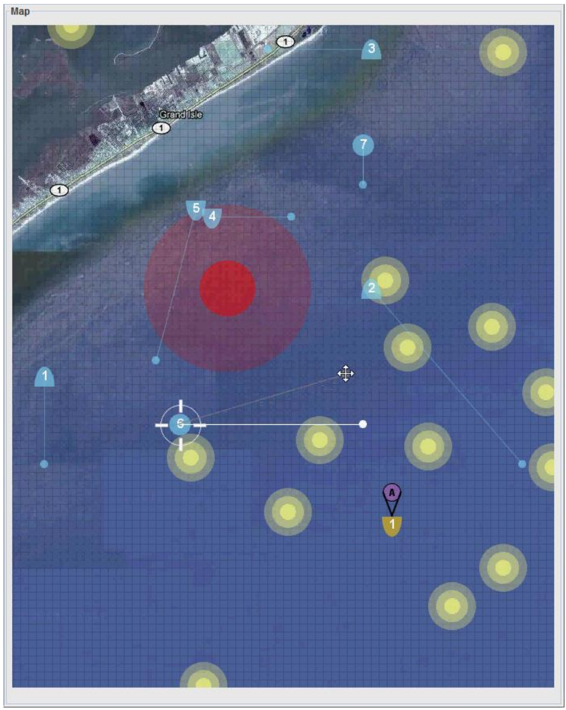


Figure 7. Hazard avoidance task.

The relationships between the operator, the CSL, the UVs and the manner in which they must process tasks may be modeled as a network of interactive queues. Figure 8 shows the general model for a network of queues involving these servers. In an "open" queuing system, "customers" (tasks) arrive at each of the servers. Some tasks are processed and leave the system, but other tasks may be passed from one server to another. Thus, tasks may sequentially arrive at different queues, be waited upon by different servers, and sometimes may "feedback" and return to a previous server before eventually leaving the system. In Figure 8, the circles represent servers: the operator, the CSL, and the UVs all are servers. Tasks from both within the network and outside the network may arrive at a server's queue. The appearance of a new unknown contact in the battle space marks the arrival of a task to the operator to identify the contact. In turn, the servers may pass each other tasks within the network. For instance, the operator may submit to the CSL one or more contacts for identification or attack assignment. Likewise, the CSL requests the operator to submit one or more contacts for identification or attack. The operator may also directly assign UV assets to identify or attack contacts, or may simply redirect the UV's path of motion. Queueing theory provides quantitative tools to analyze the flow of tasks to and from each server. In addition, the overall performance of the network may be analyzed. For this study, we have concentrated only on analyzing the operator in the queueing network (see the Future Work section of this report).

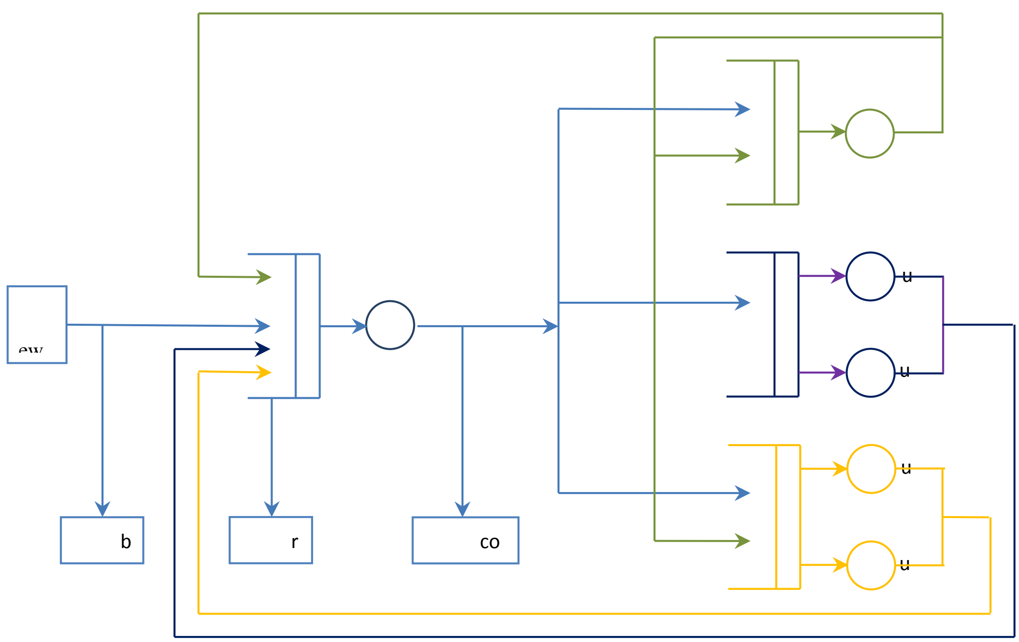


Figure 8. RESCHU SP Queueing Network.

In previous research, to support the multitasking activity associated with supervisory control, a Task Manager (TM) display was incorporated into an operator's interface. (See for example, the Multimodal Watchstation (MMWS) and the Land Attack Combat System (LACS), Osga et al., 2002). As shown in Figure 9, 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. Thus a queue of tasks is clearly represented by the interface. The current display does not have a Task Manager and one may ask the question, "where is the operator's queue?" Nonetheless, the exact arrival time of tasks the operator must perform, as well as the operator's start and completion time for each of the five tasks listed above may be determined from the RESCHU SP simulation log files. Thus, all the components necessary to formulate and analyze a queueing system may be extracted from the RESCHU SP scenario. This leads us to briefly review the elements of a queueing system.

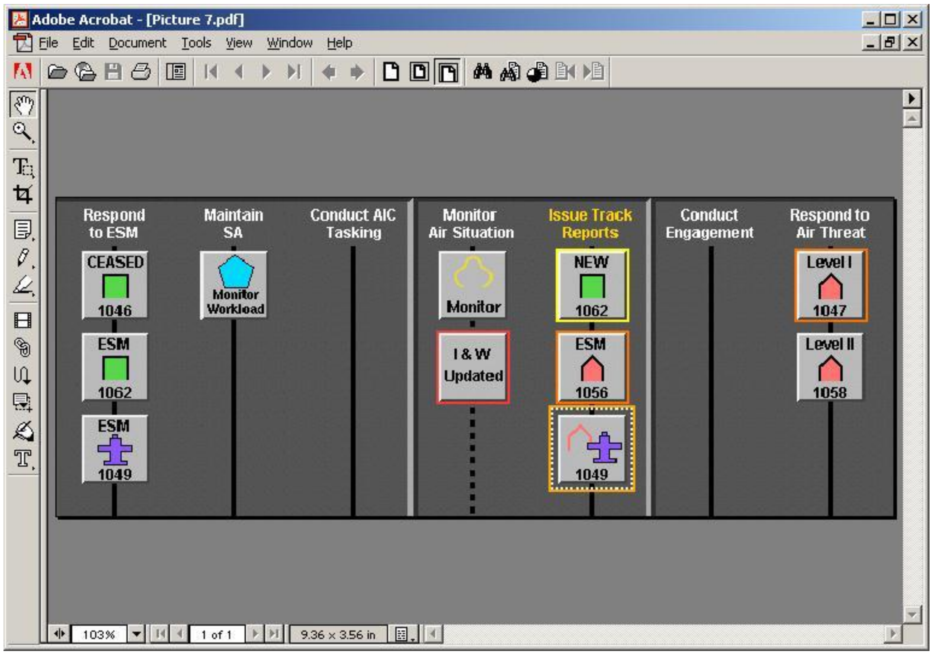


Figure 9. Task manager display.

## 1.1 ELEMENTS OF A QUEUEING SYSTEM

A queueing system is composed of three main components:

1. The input or arrival process. The arrival of customers to a queue is often unpredictable. In this case, arrival is modeled as a random or stochastic process. The arrival process is often assumed to be Poisson in nature, in which case the arrival rate, $\lambda$, is simply the reciprocal of the mean inter-arrival time of customers. For the Possion distribution with parameter $\lambda$, the probability, $P_k$, that $k$ arrivals occur in the time interval $(0,t)$ is given by:

$$P_k(t) = \frac{(\lambda t)^k}{k!} e^{-\lambda t}$$

Previous research revealed changes in the arrival rates of tasks in combat scenarios. In general, the arrival of tasks consisted of a busy phase followed by a slower phase. This ebb and flow of the rate at which tasks appear may be modeled by using the Modulated Markov Poisson Process (MMPP) (see Lucantoni, Meier-Hellstern & Neuts, 1990; Lucantoni & Ramasami, 1985; Hock (1996); DiVita, Morris, & Osga, 2007 for further details).

2. The service mechanism and vacationing servers. Service refers to the number of “servers” and the duration of time the customer’s hold servers. Service time is often modeled by a continuous random variable, *x*, exponentially distributed with parameter $\mu$:

$$f(x) = \mu e^{-\mu x}$$

The servicing of a task may be viewed as a process consisting of several serial stages. The Erlang method of stages was designed to quantify this situation. This method assumes that the distribution of service time for each stage is exponential. The distribution of overall service time is thus given by the convolution of these functions. This distribution is no longer exponential; hence, this method allows a more general service time distribution to be used in formulating predictions. When dealing with convolutions, it is easier to work with Laplace transforms. Thus, if we assume that there are r stages of processing, each with identically distributed service times, the service time distribution is given by the Erlang distribution:

$$b(x) = \frac{r\mu(r\mu x)^{r-1}e^{-r\mu x}}{(r-1)!}$$

The Laplacian, B*(s), for b(x) is given by:

$$B^*(s) = \left(\frac{r\mu}{s + r\mu}\right)^r$$

Further details of this approach are given in Kleinrock, 1975 & 1976. One critical difference between the queueing system described above and an operator engaged in supervisory control is that when no tasks are present, the server is idle; however, one hopes that this is not the case with human operators. When there are no tasks to perform, the operator is still examining the display. Ironically, in the RESCHU SP scenario, the operator is determining if a new task needs to be accomplished. However, the operator is not performing one of the five tasks that we have specified. Thus, there are “tasks” the server may perform that fall outside the flow of tasks represented by the arrival of tasks that have been specified for analysis. These non-specified tasks must be taken into account to quantify system performance because they clearly will have an impact on the queueing statistics. For example, if the operator is evaluating information on a tactical display and one of the five specified tasks arrives, the operator may finish his analysis before realizing there is a new task. The start time for that task is delayed; thus, the waiting time and overall time the task spends in the system will be affected by the operator’s extra activity. Fortunately, a queue with “service vacations” can be adapted to model this situation (Takagi, 1991). The idea behind such queues is as follows: if there are no customers in the queue that need to be served, the server takes a vacation. If upon completion of a vacation, the server returns only to find that there are still no more customers, the server takes another vacation. This process continues until the server returns to find customers waiting to be served.

3. The queueing policy entails the method by which the system selects customers: first-come-first-served (FCFS), last-come-first-served (LCFS), by priority, or at random order service (ROS). Additional considerations for the queueing policy include: Reneging, Balking, and a Finite Queue length. Reneging refers to a customer joining the queue only to leave the queue prior to obtaining service. The probability of a customer reneging is commonly modeled using a negative exponential distribution. Balking refers to the arrival of a customer, but the customer never enters the queue. Associated with balking is a probability distribution conditioned on the number in the queue when the customer arrives; that is, the probability

that a customer balks is a function of the queue length upon arrival. Balking is related to a finite queue size. For a finite queue whose maximum number of customers is K, if a customer arrives to the queue only to find K customers already in the system, that customer balks with 100% probability.

Vital Statistics of a Queue: A queueing model with 1) an exponential arrival process, 2) exponential service process, 3) one server, 4) finite queue size of K, 5) balking and reneging, and 6) a multiple vacation policy, is represented with Kendall's notation as M/M/1/K ( $\alpha\beta$ ) with a vacationing server. Here the M is a Markovian, 'memoryless,' exponential process, $\alpha$ is the renege rate also exponential, and $\beta$ indicates the probability of balking.

Various distributions and statistics can be derived for a queueing system. Most of these expressions involve the quantities $\rho$, $\lambda$, and $\mu$. For example, in analyzing the performance of a system, one may be interested in (1) the average number of customers, N, in the system at any time, (2) the average time a customer spends waiting, W, for service, and (3) the total average time, T, spent in the system by a customer. This time includes both service and waiting time. One useful result that relates several of these quantities is Little's Theorem, which states that the average number of customers to the system, *N*, is equal to the product of the rate of flow of customers, $\lambda$, and the average time spent in the system, *T*:

$$N = \lambda T$$

Another useful characterization of the system is the probability distribution for the number of customers in the queue. For example, what is the probability that a visitor to the queue finds *n* customers in the queue? The equation associated with the M/M/1/K ( $\alpha\beta$ ) are given in the next section.

### 1.2 EQUATIONS FOR THE M/M/1/K ( $\alpha\beta$ ) VACATIONING SERVER

Figure 10 depicts the steady state transition diagram for the M/M/1/K ( $\alpha\beta$ ) vacationing server. In this diagram, the empty ellipses denote the server is on vacation whereas the filled ellipses signify that the server is busy. The number inside each ellipse denotes the number of customers in the system. The maximum number in the system, K, has be set to 4; thus, there are only 9 possible states for this queue. In the diagram, $\lambda$ is a transition due to an arrival, $\mu$ is a transition created when a service completion occurs, $\alpha$ indicates a renege, and $\eta$ is a transition caused by the completion of a vacation. Note that $b_n$ is the probability of an arrival joining the queue at state $n$ (for more details of this approach see Zhang, Yue, & Yue, 2008).

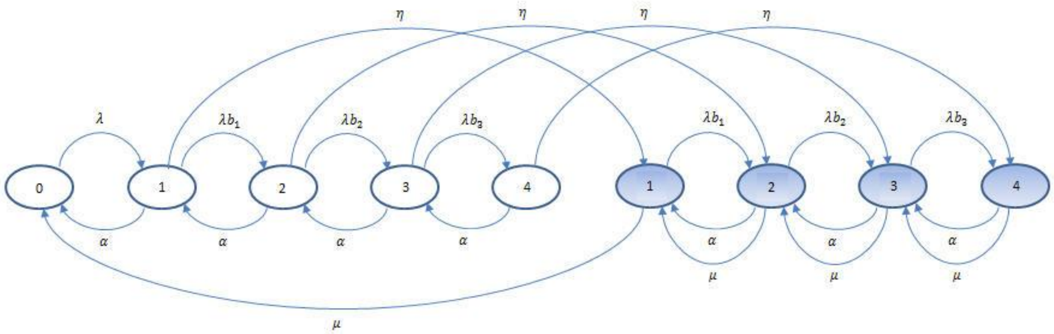


Figure 10. State transition diagram for M/M/1/K (**$\alpha\beta$**) queueing model with a vacationing server. In this illustration, the number inside each state defines the number of tasks in the system. The unfilled state ellipses indicate vacation periods, and the filled states indicate the system is busy. Labeled arrows indicate a transition between states. Here $\boldsymbol{\lambda}$ is a transition due to an arrival, $\boldsymbol{\mu}$ is a transition created when a service completion occurs, $\boldsymbol{\alpha}$ indicates a renege, and $\boldsymbol{\eta}$ is a transition caused by the completion of a vacation. Note that $\boldsymbol{b_n}$ is the probability of an arrival joining the queue at state $\boldsymbol{n}$. This example shows a system where the total capacity $\boldsymbol{K}$ is 4; hence, there are no other states, and tasks arriving to a full system are lost.

Let $p_0(n)$ denote the probability that the server is on vacation and there are n customers in the queue, and let $p_1(n)$ denote that the server is busy and there are n customers in the system. Then the transition equations are given by:

$$\mu p_1(1) + \alpha p_0(1) = \lambda p_0(0)$$

$$\lambda b_{n-1} p_0(n-1) + \alpha p_0(n+1) = (\alpha + \lambda b_n + \eta) p_0(n), n = 1,2,\cdots,N-1$$

$$\lambda b_{N-1} p_0(N-1) = (\alpha + \eta) p_0(N)$$

$$\eta p_0(1) + (\mu + \alpha) p_1(2) = (\lambda b_1 + \mu) p_1(1)$$

$$\lambda b_{n-1} p_1(n-1) + \eta p_0(n) + (\mu + \alpha) p_1(n+1) = [\lambda b_n + \mu + \alpha] p_1(n), n = 2,\cdots,N-1$$

$$\lambda b_{N-1} p_1(N-1) + \eta p_0(N) = [\mu + \alpha] p_1(N)$$

The steady state equations assume that flow into a state is equal to flow out of a state. For example, $\mu p_1(1) + \alpha p_0(1) = \lambda p_0(0)$ states the flow into the state $p_0(0)$ (no customers in the queue and the server is on vacation) can only happen if a service is completed in state $p_1(1)$, [$\mu p_1(1)$] or if there is a renege in state $p_0(1)$ [$\alpha p_0(1)$]. The flow into $p_0(0)$ must be equal to the flow out of $p_0(0)$. Flow out of p0(0) can only happen if there is an arrival when the queue is in state $p_0(0)$, that is, λp0 (0).

Let $P_0 = [p_0(0), \quad p_0(1), \quad \cdots \quad p_0(K)]$,$P_1 = [p_1(1), \quad \cdots \quad p_1(K)]$, and $P = [P_0 \quad P_1]$. If $Q$ is the transition rate matrix of the Markov Process, then the following equations must be satisfied:

$$PQ = 0$$

$$Pe = 1$$

where $e$ is a column vector with each component equal to one. The block form of the matrix Q and its subcomponents are defined as:

$$Q = \begin{bmatrix} B_0 & A \\ C & B_1 \end{bmatrix} \in \mathbb{R}^{(2K+1)\times(2K+1)}$$

$$C = \begin{bmatrix} \mu & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{K\times(K+1)}, A = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ \eta & 0 & \cdots & 0 \\ 0 & \eta & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & \eta \end{bmatrix} \in \mathbb{R}^{(K+1)\times K},$$

$$B_0 = \begin{bmatrix} -\lambda & \lambda & 0 & 0 & \cdots & 0 & 0 & 0 \\ \alpha & -c_1 & \lambda b_1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 2\alpha & -c_2 & \lambda b_2 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & (K-1)\alpha & -c_{K-1} & \lambda b_{K-1} \\ 0 & 0 & 0 & 0 & \cdots & 0 & K\alpha & -(K\alpha+\eta) \end{bmatrix} \in \mathbb{R}^{(K+1)\times(K+1)},$$

$$B_1 = \begin{bmatrix} -d_1 & \lambda b_1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ \mu+\alpha & -d_2 & \lambda b_2 & 0 & \cdots & 0 & 0 & 0 \\ 0 & \mu+2\alpha & -d_3 & \lambda b_3 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \mu+(K-2)\alpha & -d_{K-2} & \lambda b_{K-1} \\ 0 & 0 & 0 & 0 & \cdots & 0 & \mu+(K-1)\alpha & -[\mu+(K-1)\alpha] \end{bmatrix}$$
$$\in \mathbb{R}^{K\times K},$$

$$c_i = i\alpha + \lambda b_i + \eta, d_i = \lambda b_i + \mu + (i-1)\alpha, i = 1, \cdots, K-1.$$

Here the probability of joining the queue depends on the number in the system $n$:

$$b_n = \frac{1}{n+1}, b_0 = 1\ b_n = 0\ for\ n \geq K.$$

Define,

$$Z = I - CB_0^{-1}AB_1^{-1} \in \mathbb{R}^{K\times K}$$

and

$$V = e_1 - CB_0^{-1}e_0 \in \mathbb{R}^{K\times 1},$$

where $e_0 \in \mathbb{R}^{K+1\times 1}$, and $e_1 \in \mathbb{R}^{K\times 1}$ are column vectors whose elements are all ones. Then

$$\tilde{Z} = [V \quad \hat{Z}],$$

where $\hat{Z}$ is $Z$ with the first column removed. The solution to the steady state equations follow:

$$P_1 = g\tilde{Z}^{-1}$$

$$P_0 = -P_1 CB_0^{-1},$$

where $g$ is a row vector of length $K$

$$g = [1 \quad 0 \quad \cdots \quad 0].$$

These probabilities may be used to compute N, the average number of customers in the queue:

$$N = \sum_{i=0}^{1}\sum_{n=0}^{K}(n-i)p_i(\mathrm{n})$$

Once N is determined, Little's theorem can be applied to compute the average time in the queue, T, and the average waiting time in the queue; however, λ must be modified to reflect balking. Thus letting P (Balk) equal the probability of balking and P(Join) equal the probability of joining the queue, we have:

$$\mathrm{P(Balk)} = \sum_{\mathrm{i=0}}^{1}\sum_{\mathrm{n=i}}^{\mathrm{K}}(1-\mathrm{b_n})\mathrm{p_i}\,(\mathrm{n})$$

$$\mathrm{P(Join)} = \sum_{\mathrm{i=0}}^{1}\sum_{\mathrm{n=i}}^{\mathrm{K}}\mathrm{b_n}\mathrm{p_i}\,(\mathrm{n})$$

Let

$$\overline{\lambda} = \lambda\, P(Join)$$

Then the average time in the system T is given by:

$$T = \frac{N}{\overline{\lambda}}$$

The average time spent waiting for service is given by:

$$W = \frac{N_q}{\overline{\lambda}},$$

where $N_q$ is the average number in the system waiting for service and is given by:

$$N_q = \sum_{i=0}^{1}\sum_{n=i}^{K}(n-i)p_i\,(n).$$

In order to verify the equations, a simulation for a M/M/1/K ($\alpha\beta$) Vacationing Server, with K=3, $\lambda = 0.4000$, $\mu = 0.2642$, $\alpha = 0.1000$, $\beta = 0.5140$, $\eta = 0.1000$, was created in Matlab using 100,000 customers arriving at the queue. In Table 1, the predicted queueing statistics are compared to the observed statistics. As can be seen, the error is minor, on the order of 0.25%. In Table 2, the $p_0$ (n) and $p_1$ (n) probabilities are compared, predicted versus obtained. Again the percent error is minor.

Table 1. Simulation observed queueing statistics compared to predicted queueing statistics. The simulation was run with 100,000 customers.

M/M/1/K ($\alpha\beta$) Queue with Vacationing Server

$K = 3,\ \lambda = 0.4000,\ \mu = 0.2642,\ \alpha = 0.1000,\ \beta = 0.5140,\ \eta = 0.1000$

| | **N** | **T** | **W** |
|---|---|---|---|
| **Observed** | 1.329230 | 6.822919 | 4.910755 |
| **Predicted** | 1.331202 | 6.847554 | 4.927672 |
| **% Error** | 0.15 | 0.36 | 0.34 |

Table 2. Simulation observed transition probabilities compared to predicted transition probabilities (100,000 customers).

| | **$p_0(0)$** | **$p_0(1)$** | **$p_0(2)$** | **$p_0(3)$** | **$p_1(1)$** | **$p_1(2)$** | **$p_1(3)$** |
|---|---|---|---|---|---|---|---|
| **Observed** | 0.176741 | 0.250199 | 0.150256 | 0.050282 | 0.170865 | 0.148172 | 0.053485 |
| **Predicted** | 0.175494 | 0.250706 | 0.150423 | 0.050141 | 0.170806 | 0.148868 | 0.053561 |
| **% Error** | 0.71 | 0.20 | 0.11 | 0.28 | 0.03 | 0.47 | 0.14 |

Figures 11 and 12 show the change in average number in the queue N, when the reneging and vacation parameters systematically vary between 0 and 1. There were two cases for balking. In the first case, "canbalk = 0," $b_n = 1$, for $n < K$ and $b_n = 0$ for $n \geq k$. In the second case, "canbalk = 1," $b_0 = 1$, $b_n = 1/(n+1)$, for $n < K$ and $b_n = 0$ for $n \geq k$.

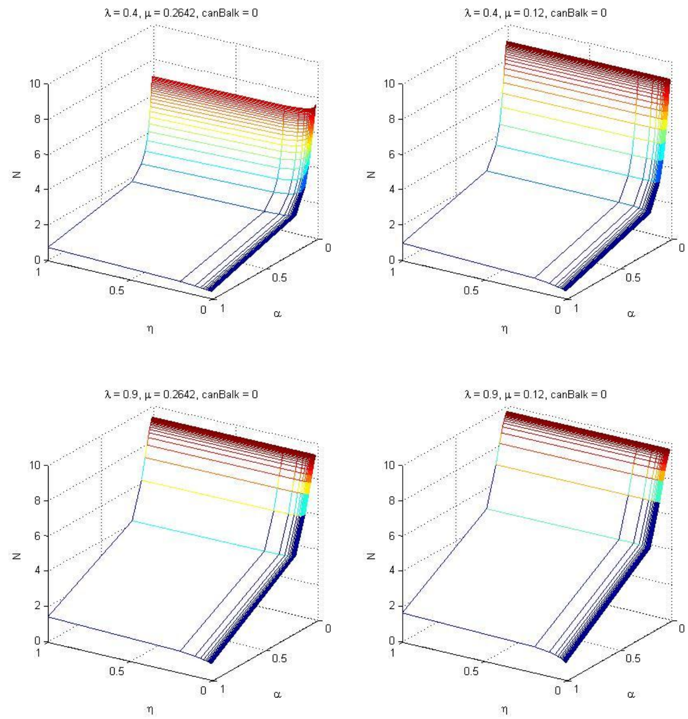


Figure 11. Sensitivity analysis when queueing system is not allowed to balk with the number in the system less than K.

In Figure 11, no balking is allowed for $n < K$, the maximum number the system can hold. In the upper panel of Figure 11, $\lambda$ has been held constant, (0.4) while the service rate has changed from relatively fast to slow ($\mu = 0.2642$, as opposed to $\mu = 0.12$). Not surprisingly, the decrease in service rate forces N to increase. Figure 11 demonstrates the rate of this increase and how it interacts with the $\alpha$ and $\eta$ parameters of the queueing model. The lower half of Figure 11 increases the arrival rate $\lambda$. Again N increases; however, the $\alpha$ and $\eta$ parameters have a great effect on the rate of this increase. Figure 12 is similar to Figure 11, only now balking is allowed for $n < K$. With balking allowed, the N greatly decreases.

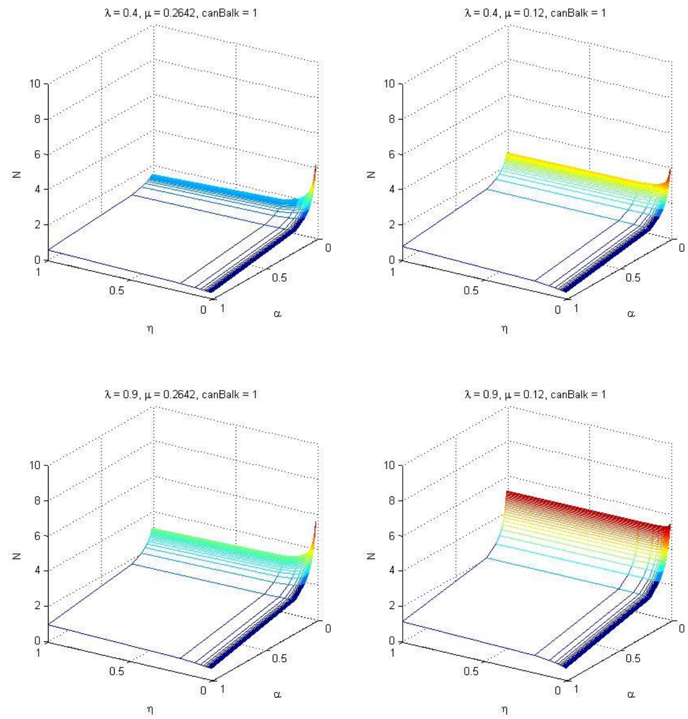


Figure 12. Sensitivity analysis when queueing system is allowed to balk with the number in the system less than K.

## 2. RESULTS

The output of the RESCHU SP simulation is a log file containing all events that occur during the course of a subject interacting with the simulation. One log file is generated per mission with a single operator. A small set of mission simulation output log files were analyzed manually to determine the cues which specified the arrival of a task, the setup time prior to service of the task (see below), the beginning of service performed by the operator, and the end of service. It was discovered that there were many occurrences where a task disappeared prior to the beginning of service. To handle the problem, a task renege time cue was determined for all tasks. The renege time is defined as the projected time that a task would disappear in the event the operator never performs service on the task.

A Perl script was created to automate the parsing of each simulation output log file. The Perl script produces a list of tasks discovered during a simulation run. This task list includes 1) task arrival time, 2) task setup time, 3) begin service time, 4) end service time, and 5) task renege time. The task list was sorted in ascending order on the arrival time.

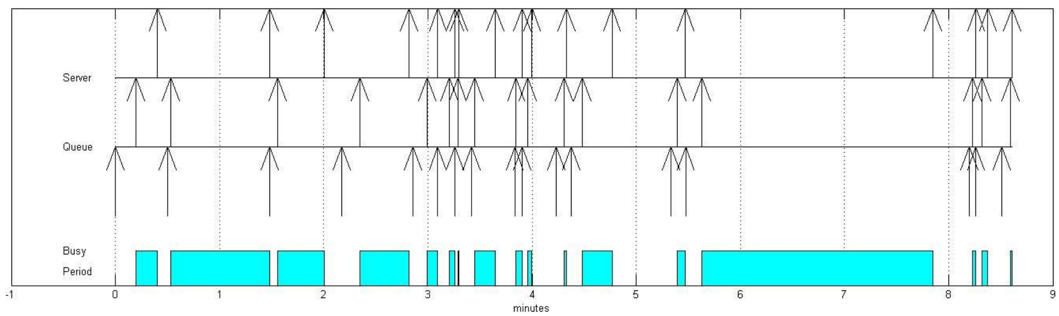


Figure 13. Lower arrows indicate first arrival during a vacation period, the middle arrows indicate the begin service time, which starts a busy period, and the upper arrows show the departure time, which ends a busy period. The busy period is represented by the shaded boxes at the bottom of the figure.

When an operator completes a task, and a second task is waiting for service, the operator does not immediately start the second task. It was decided that this behavior could partially be explained by a setup time performed by the operator to regain situational awareness. The setup time was combined with the service time in a two stage Erlang-r service process. The first stage of service corresponds to activity required to gain situational awareness. The second stage of service corresponds to the time it takes the operator to complete the task.

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.

The inter-arrival durations, combined service/setup durations, renege durations, and vacation durations were analyzed to determine their underlying stochastic distributions. For each observed distribution of times, an average was calculated and a $\chi^2$ test of the observed times was compared to an exponential distribution with the same mean. For the arrival process, the service process, the renege, and vacation durations, analyses are depicted in Figure 14, Figure 15, Figure 16, and Figure 17, respectively.

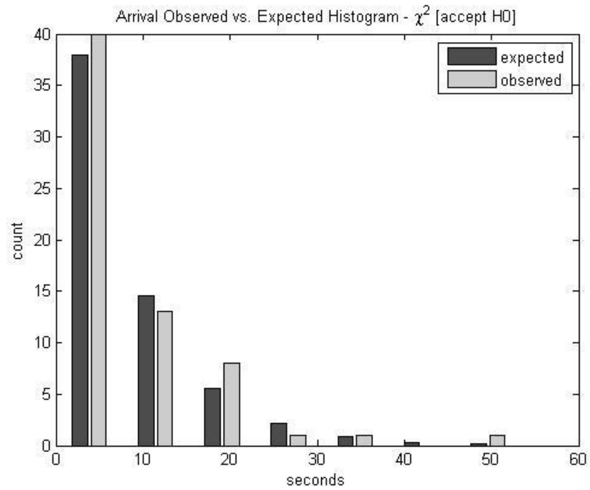


Figure 14. Inter-arrival duration histogram shows that the observed data matches an Erlang-r (r = 1) which is equivalent to an exponential distribution. A $\chi^2$ test failed to reject the hypothesis that the observed matches the expected.

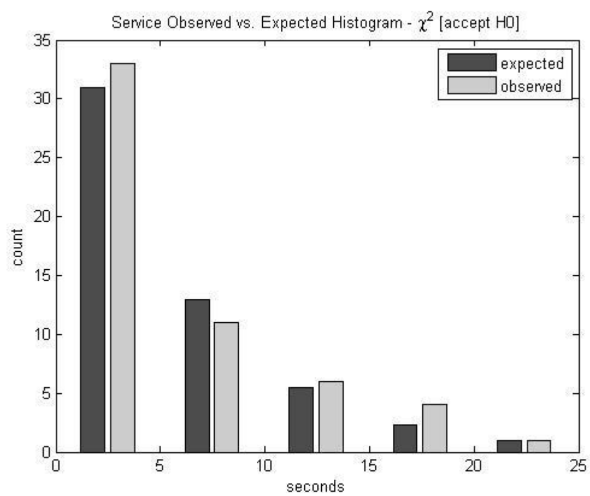


Figure 15. Service duration histogram shows that the observed data matches an Erlang-r (r = 1) which is equivalent to an exponential distribution. A χ^2 test failed to reject the hypothesis that the observed matches the expected.

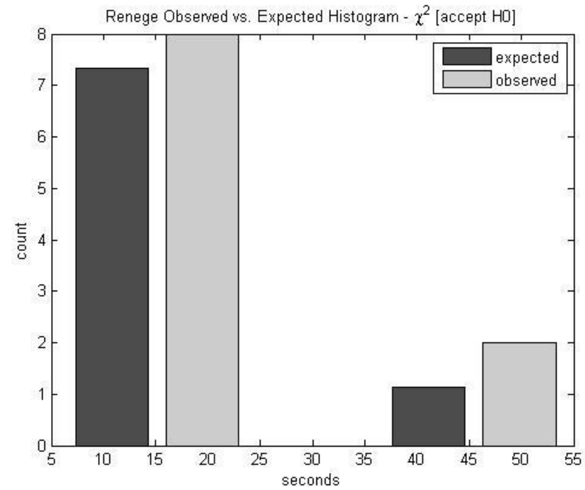


Figure 16. Renege duration histogram shows that the observed data matches an Erlang-r (r = 1) which is equivalent to an exponential distribution. A χ^2 test failed to reject the hypothesis that the observed matches the expected. However, only 10 of the 65 tasks reneged prior to the task receiving service.

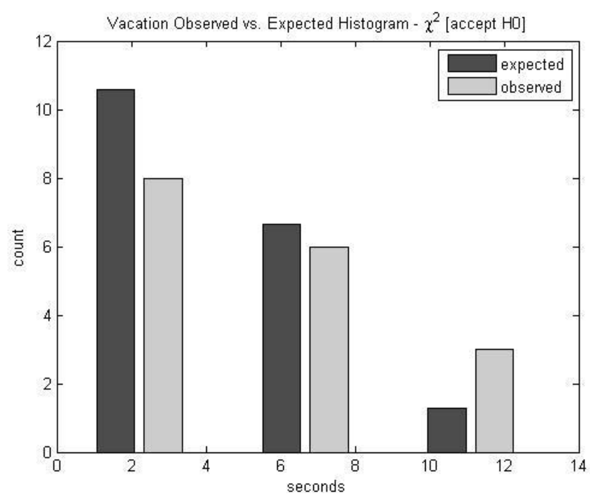


Figure 17. Vacation duration histogram shows that the observed data matches a three stage Erlang-r distribution. A χ^2 test failed to reject the hypothesis that the observed matches the expected.

In each case, the binning of the data was optimized to compare the expected number of observations from the exponential distribution to the observed data. It was determined that the inter-arrival process, service process, and renege process failed to reject the null hypothesis of an exponential distribution. However, the vacation process rejected the exponential hypothesis. In a second analysis, the second moment of the observed data was compared to the second moment of the

Erlang–r distribution, for r = 1,…, 6, for the arrival process, the service process, the renege and vacation durations. These analyses are given in Table 3, Table 4, Table 5, and Table 6, respectively.

Table 3. Minimizing the error of the observed vs. expected second moment to find the best fitting r-stage Erlang distribution indicates that a single exponential stage best fits the arrival process.

| r | $M_2$ Expected | $M_2$ Observed | Error |
|---|---|---|---|
| 1 | 127.2511 | 138.8925 | 11.6414 |
| 2 | 95.4383 | 138.8925 | 43.4542 |
| 3 | 84.8341 | 138.8925 | 54.0585 |
| 4 | 79.5319 | 138.8925 | 59.3606 |
| 5 | 76.3507 | 138.8925 | 62.5419 |
| 6 | 74.2298 | 138.8925 | 64.6627 |

Table 4. Minimizing the error of the observed vs. expected second moment to find the best fitting r-stage Erlang distribution indicates that a single exponential stage best fits the service process.

| r | M2 Expected | M2 Observed | Error |
|---|---|---|---|
| 1 | 65.4368 | 61.5602 | 3.8766 |
| 2 | 49.0776 | 61.5602 | 12.4826 |
| 3 | 43.6245 | 61.5602 | 17.9356 |
| 4 | 40.8980 | 61.5602 | 20.6622 |
| 5 | 39.2621 | 61.5602 | 22.2981 |
| 6 | 38.1715 | 61.5602 | 23.3887 |

Table 5. Minimizing the error of the observed vs. expected second moment to find the best fitting r-stage Erlang distribution indicates that a single exponential stage best fits the renege process.

| r | $M_2$ Expected | $M_2$ Observed | Error |
|---|---|---|---|
| 1 | 527.1504 | 578.9733 | 51.8228 |
| 2 | 395.3628 | 578.9733 | 183.6104 |
| 3 | 351.4336 | 578.9733 | 227.5396 |
| 4 | 329.4690 | 578.9733 | 249.5042 |
| 5 | 316.2903 | 578.9733 | 262.6830 |
| 6 | 307.5044 | 578.9733 | 271.4688 |

Table 6. Minimizing the error of the observed vs. expected second moment to find the best fitting r-stage Erlang distribution indicates that a three stage best fits the vacation process.

| r | $M_2$ Expected | $M_2$ Observed | Error |
|---|---|---|---|
| 1 | 50.2356 | 35.2024 | 15.0332 |
| 2 | 37.6767 | 35.2024 | 2.4743 |
| 3 | 33.4904 | 35.2024 | 1.7120 |
| 4 | 31.3972 | 35.2024 | 3.8051 |
| 5 | 30.1413 | 35.2024 | 5.0610 |
| 6 | 29.3041 | 35.2024 | 5.8983 |

As can be seen from these tables, the inter-arrival times, the service times, and the renege durations, a value of r =1 minimized the error. This is equivalent to an exponential distribution. For the vacation durations, r = 3 minimizes the error. Since there were a small number of vacation durations that were estimated, an exponential distribution will be assumed for all processes. The data are summarized are in Table 7.

Table 7. Minimizing the error of the observed vs. expected second moment to find the best fitting r-stage Erlang distribution indicates that the arrival, service, and renege processes are best fit with a single exponential stage. However, the vacation process is best fit with a three stage Erlang distribution.

| Process | Best r | $M_2$ Expected | $M_2$ Observed | Percent Error |
|---|---|---|---|---|
| Arrival | 1 | 127.2511 | 138.8925 | 9.1484 |
| Service | 1 | 65.4368 | 61.5602 | 5.9242 |
| Renege | 1 | 527.1504 | 578.9733 | 9.8307 |
| Vacation | 3 | 33.4904 | 35.2024 | 5.1118 |

For the inter-arrival times of tasks, two more analyses were performed. In Figure 18, 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 a single arrival phase. In Figure 19, a MMPP fitting algorithm developed by Meier-Hellstern (1987) was performed on the data. This analysis corroborated the first change point analysis and only a single phase was found. Note that the scenario contained two waves of enemy vehicles. In each wave, 50 millisecond inter-arrival times were discovered. These short inter-arrival durations can be seen in the figures at task number 3, 5-7, 40-43.

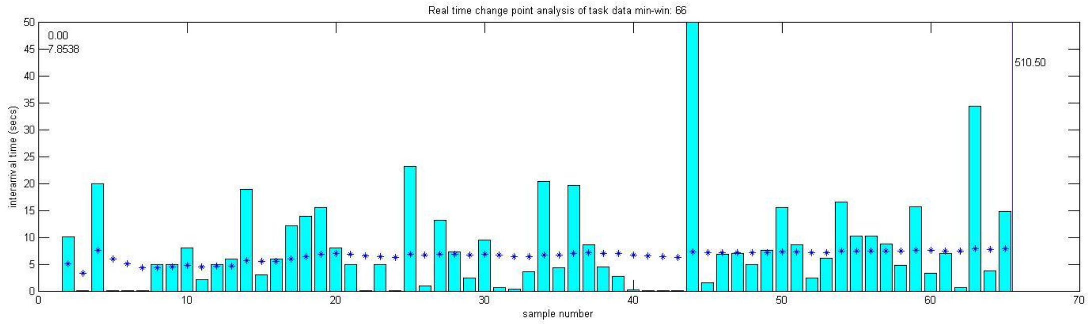


Figure 18. Change point analysis of arrival distribution indicates single phase of average duration 7.8538.

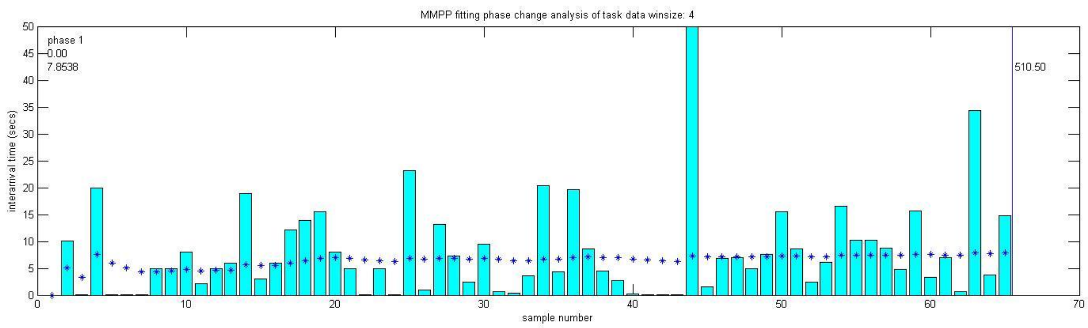


Figure 19. An MMPP fitting algorithm supports the change point analysis which also indicates only a single phase.

From examining the log files, it is clear that the operator does not use a FCFS policy. This is not surprising because the interface does not keep track of when tasks arrived. Logically, some tasks should have a higher priority than others, although further analysis of the data must be conducted to determine if there was any prioritization of tasks on the part of the operator. For the current study, we will assume that the operator selected tasks from the queue randomly. It may be shown that the key averages for queueing statistics such as the average number of customers in the system, N, the average time spent in the system, T, and average time waiting W, are equivalent for the FCFS and ROS service policies (Flatto, 1997). Future research will examine task prioritization.

In Table 8, the actual values observed from the RESCHU SP data are given for the arrival parameter, $\lambda$, the service parameter, $\mu$, the renege parameter, $\alpha$, and the vacation parameter, $\eta$. The finite system size, K was observed to be six. A Matlab simulation of the M/M/1/K ($\alpha\beta$) with vacationing server model was run using the observed parameters in Table 8. Using 100,000 tasks, this simulation observed a balking rate, $\beta$, also listed in Table 8. In Table 9, the predicted value for N, W, and T computed from the queueing equations are compared to the observed values from the Matlab simulation. As can be seen, the errors between the predicted and observed values are less than 1% (approximately, 0.2%).

Table 8. Actual parameters observed in RESCHU SP mission simulation.

| $K$ | $\lambda$ | $\mu$ | $\alpha$ | $\beta$ | $\eta$ |
|---|---|---|---|---|---|
| 6 | 1/7.976563 | 1/5.720000 | 1/16.235000 | 0.0025 | 1/5.011765 |
| 6 | 0.125367 | 0.174825 | 0.061595 | 0.0025 | 0.199531 |

Table 9. Using the RESCHU SP actual parameters, the observed queueing simulation is compared to the predicted results.

M/M/1/K ($\alpha\beta$) Queue with Vacationing Server
$K = 6$, $\lambda =$ 1/7.976563, $\mu =$ 1/5.720000, $\alpha =$ 1/16.235000,
$\beta = 0.0025$, $\eta =$ 1/5.011765

| | N | T | W |
|---|---|---|---|
| **Simulation Observed** | 1.158548 | 9.254135 | 5.484697 |
| **Predicted** | 1.160517 | 9.279735 | 5.496177 |
| **% Error** | 0.17 | 0.28 | 0.21 |

In Table 10, the predicted values of N, W, and T are compared to the actual values calculated from the RESCHU SP scenario for one operator. The error between the predicted and actual values is on the order of 20%. This is not surprising considering 1) the operator's vacation times were not exponentially distributed and 2) the operator's queuing policy was not strictly ROS.

Table 10. Actual observed queueing parameters calculated from the RESCHU SP logfile are compared to the predicted queueing parameters.

M/M/1/K ($\alpha\beta$) Queue with Vacationing Server
$K = 6$, $\lambda =$ 1/7.976563, $\mu =$ 1/5.720000, $\alpha =$ 1/16.235000, $\beta = 0.0025$,
$\eta =$ 1/5.011765

| | **N** | **T** | **W** |
|---|---|---|---|
| **Actual Observed** | 1.418861 | 11.272308 | 6.432308 |
| **Predicted** | 1.160517 | 9.279735 | 5.496177 |
| **% Error** | 22.26 | 21.47 | 17.03 |

# 3. CONCLUSIONS

The RESCHU SP simulation queueing model project is an effort to apply queueing theory to a UV simulation where an operator is faced with the task of defending an offshore oil rig. It was determined that an M/M/1/K ($\alpha\beta$) queue with vacationing server was a good first approximation to the data. The data were analyzed and the observed queueing parameters were used to predict, N, the average number of tasks in the queue, T, the average time spent in the queue (both waiting and service) and W, the average waiting time in the queue. The error between the predicted values and the obtained values were on the order of 20%. This error is due to several factors: 1) the operator did not use an assumed ROS queueing policy, and 2) the distribution of the operator's vacation times was not exponential. Future research (see next section) will evolve and generalize the queueing model to refine the predictions.

# 4. FUTURE RESEARCH

## 4.1 EVOLVING THE QUEUEING MODEL

The following research areas were identified to evolve and generalize the queueing model. The aim of targeting these areas is to improve the model predictions.

1. Analyzing Each Node of the Network

   Figure 8 depicts the entire queuing network composed of human operator, CSL, and UVs. The research discussed in this paper only analyzed the operator node. Each node of the network may be analyzed from the point of view of queueing theory, That is, for each node an arrival process, a service process, and a queueing policy may be specified and all the queueing statistics (e.g., N, T, W) can be computed. For example, the load to each node, $\lambda_i$, is the sum of customers arriving from "outside" the queue and customers passed between servers. Thus,

   $$\lambda_i = \gamma_i + \sum_{j=1}^{M} \lambda_j p_{ji},$$

   where $\gamma_i$ is the rate of flow of customers arriving from outside the queue, $\lambda_j$ is the total flow of customers to the $j_{th}$ server, $p_{ji}$ is the probability that customers arriving to the $j_{th}$ server will be passed on to the $i_{th}$ server, and $M$ equals the total number of servers. Likewise, each server would have its unique service distribution(s) and queueing policy. Most importantly, by modeling the entire network, bottlenecks would be revealed. For example, there may be scenarios where the number of UV assets must be increased in order to successfully complete the mission, or the performance and service of the CSL must be improved. The current analysis only deals with the operator node, which may not prove to be the critical bottleneck in some scenarios. Thus, increasing the number of operators will not improve system performance if there are too few UVs to identify or attack unknown and enemy contacts.

2. Modeling the Network as a Hybrid Open and Closed Queue

   The network may also be modeled as a hybrid open and closed queue. That is, some aspects of the network are indicative of an open network, as was discussed above, but other aspects are indicative of a closed network. A closed queue is one in which customers never enter or leave the queue; instead, they continuously cycle through the queue being passed from server to server. There are aspects of the operator's tasks that suggest a closed queueing system. For example, we have stated that when the operator no longer had any of the five analyzed tasks to perform, they went off and "took a vacation," hence the vacationing server parameters were introduced into the analysis. A more accurate characterization of the operator's behavior would be to say that when there are no other tasks to perform, the operator serves the ever present task of gaining situational awareness. In fact, with the current RESCHU SP interface, there is no task manager display; that is, there is no list of tasks the operator is instructed to perform. Thus the operator must always be determining what task to do next. From this perspective, the task – "figure out what to do next" – is always present in a closed queue fashion. After the operator performs a task, a new "customer" immediately appears in the form of the task – "figure out what to do next." From this perspective, the setup time that is observed in the current model could be viewed in terms of service time to perform this closed-queue task as opposed to adding this setup time to the next task the operator finally decides to perform. This perspective would lead to a new set of queueing equations that may better describe the network's performance.

3. Expanding the RESCHU SP Scenario

   The current scenario is relatively brief and only lasts approximately 10 minutes. A longer C2 scenario would not only be more realistic but it would also illuminate various aspects of the queueing system that are not accessible from a brief scenario. For example, a longer C2 scenario would likely show an ebb and flow to the arrival of tasks that the system is responsible to process. System performance would then be evaluated for both heavy and small loads to the system. In particular, the system's ability to handle the transition between heavy and low workloads would also be evaluated.

4. Generalizing the Arrival Process, the Service Process, and the Queueing Policy

   Each aspect of the queueing model, arrival, service, and queueing policy may be generalized to better represent the underlying process at work in the RESCHU SP scenario. With each modification a new queueing model may be generated that leads to different predictions. All of these predictions may be compared to the obtained data to determine what factors best quantify system and operator performance.

5. Generalizing the Arrival Process

   As discussed above, previous analysis of C2 scenarios revealed an ebb and flow to tasks that the operator must perform. This arrival process is best modeled by the MMPP (DiVita, Morris, & Osga, 2007). Thus the queueing model would have to be modified to handle the MMPP arrival process.

   In the current analysis, the operator is responsible for handling five tasks. If the scenario was expanded, there would be sufficient data to determine a unique arrival rate for each of these tasks. Likewise, if the queueing policy was expanded to include priority classes (see below), the arrival rate of each class must be considered separately. Lastly, in the current scenario there is evidence of bulk arrivals; that is, several tasks arrive simultaneously. The queueing model could be modified to handle bulk arrival.

6. Generalizing the Service Process

   As discussed above, the Erlang-r is an effective method with which to model operator's service time of tasks. The Erlang-r effectively represents the stages of task completion. It is very likely that each of the five tasks considered for analysis in this project has a unique service-time mean and distribution. A more fine grain analysis of operator performance and more data for each of the operator's task would reveal these differences. These different service times would then be incorporated into a more detailed queueing model. Such detail would also be necessary if the queueing policy allowed for the prioritization of tasks.

There is evidence of bulk service in the current scenario. That is, n customers are served at once. This appears to be the case with the CSL when an operator submits a number of contacts to the CSL to be identified or attacked.

7. Generalizing the Queueing Policy – Prioritization

   The current analysis assumed an ROS queueing policy. Clearly the operator did not service tasks in a random order. There is some evidence of prioritization of tasks and some evidence for FCFS. The problem with the current display is that it lacks a task manager display that captures the arrival time of a task. One modification to the display would be to introduce a task manager. This task manager may also prioritize tasks. For example, the distance from the oil rig and the nature of the task (identification vs. attack vs. hazard avoidance) may be taken into account to determine its importance relative to other tasks that have to be performed. System performance could then be tested with and without a task manager display, with particular emphasis on how the system handles high-priority tasks. In previous research, we have modeled a prioritization queueing policy and it does increase the queuing model's predictive capability. The problem with human servers is that they are often inconsistent with prioritization, thus research concerning a "fuzzy" prioritization queueing policy should also be explored.

8. Balking and Reneging

   If operator team size is allowed to vary, then balking becomes an experimental factor in that an operator's queue is a finite size. Any additional (potentially lost) customers could then be sent to other operators. Likewise, reneging could be handled differently. At present, a task reneges and does not get serviced. In the current scenario, this meant that a UV did not avoid a hazard zone and suffered damage.

### 4.2 DECISION NETWORK MODEL

At SSC Pacific, a decision network model has been developed to explore the team size of in the RESCHU SP simulation (Rodas, Veronda, & Szatkowski, 2011). The queueing statistics determined in the current study can be used as data to this decision network. A decision network allows researchers to capture qualitative aspects of the system as well as the quantitative measures captured by the queueing model. For example, the quality of service is never directly addressed in the queueing metrics, but the quality of service could be a factor in a decision network. Thus, various queueing statistics such as average number and time in the system may be made dependent on levels of quality of service. Future research is required to blend the quantitative measures of system performance derived from queueing theory with the qualitative measures of system performance that can be represented in a decision network. The combination of quantitative and qualitative measures would provide for a more powerful and comprehensive analysis of system performance.

### 4.3 CONCLUSION

The current research project and data analysis suggests that there are several interesting and important areas of research that can be explored in order to improve the predictability of system performance using queueing theory.

# 5. REFERENCES


Chen, J. & Gupta, A. K. (2000). Parametric change point analysis. Birkhauser: Boston.

DiVita, J. & Morris, R., & Nguyen, H. T, (2005). The vacationing server: a queueing model for supervisory control. Space & Naval Warfare Systems Center Pacific, Technical Report 1920.

DiVita, J., Morris, R., & Osga, G. (2004). Modeling team performance in the air defense warfare (ADW) domain. Paper presented at the Command and Control Research and Technology Symposium, San Diego, CA.

DiVita, J., Morris, R., & Osga, G. (2005). Modeling supervisory control in the air defense warfare domain with queueing theory. Paper presented at the 10th International Command and Control Research and Technology Symposium, Tysons Corner, VA.

DiVita, J., Morris, R., & Osga, G. (2006). Modeling supervisory control and team performance in the air defense warfare domain with queueing theory, part II, paper presented at the Command and Control Research and Technology Symposium, 2006, San Diego CA.

DiVita, J., Morris, R., & Osga, G. (2007). A modeled based approach to interface design and usability testing: decision support systems & modeling for intelligent mission managing and monitoring. FYO7 ONR Progam Plan summary.

Flatto, L. (1997). The waiting time distribution for the random order service M/M/1 queue. *The Annals of Applied Probabilty*, Vol 7, No.2, 382-409.

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

Kellmeyer, D. & DiVita, J. (2012). Risk based attention management for supervisory controllers of multiple autonomous systems, SSC Pacific, white paper.

Kleinrock, L. (1975). *Queueing Systems, Volume I: Theory*. Wiley-Interscience, New York, NY.

Kleinrock, L. (1976). *Queueing Systems, Volume 2: Computer Applications*. Wiley-Interscience, New York, NY.

Lucantoni, D., Meier-Hellstern, K. S., & Neuts, M. F. (1990). A single server queue with server vacations and a class of non-renewal arrival processes. *Advanced Applied Probability*. **22**(2) 676-705

Lucantoni, D., & Ramaswami, V. (1985). Efficient algorithms for solving the non-linear matrix equations arising in phase type queues. *Communication Statistical Stochastic Models*, 29-51.

Meier-Hellstern, M. S. (1987). A fitting algorithm for Markov-modulated Poisson processes having two arrival rates. *European Journal of Operations Research*, **29** 370-377.

Nehme, C. E. (2009) Modeling supervisory control in heterogeneous unmanned vehicle systems. Ph D. Thesis, MIT Dept. of Aeronautics and Astronautics, Cambridge, Ma.

Osga, G., Van Orden, K., Campbell, N., Kellmeyer, D., & Lulue, D. 2002. Design and evaluation of warfighter task support methods in a multi-modal watchstation. Space and Naval Warfare Systems Center, Pacific. Technical Report 1874. San Diego, CA.

Rodas, M. O., Szatkowski, C. X., & Veronda, M. C. (2011). evaluating unmanned system's command and control technologies under realistic assumptions, paper presented at the 16th International Command and Control Research and Technology Symposium (ICCTRS) Conference, Quebec City, Canada.

Rodas, M.O., Veronda, M. C., & Szatkowski, C.X. (2011). "Developing a Decision Tool to Evaluate Unmanned System's Command and Control Technologies in a Network Centric

Operations Environment," paper presented at the IEEE IARIA Cognitive Conference, Rome, Italy.

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, NY.

Zhang, Y.,Yue, D. & Yue, W. (2008) Analysis of an M/M/1?N Queue with Balking, Reneging and Server Vacations. http://www.aporc.org/LNOR/5/ISORA2005F04.pdf

## REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 0704-01-0188*



| **1. REPORT DATE** *(DD-MM-YYYY)* | **2. REPORT TYPE** | **3. DATES COVERED** *(From - To)* |
|---|---|---|
| 09–2013 | Final | |

**4. TITLE AND SUBTITLE**

A Queueing Model for Supervisory Control of Unmanned Autonomous Vehicles

**5a. CONTRACT NUMBER**

**5b. GRANT NUMBER**

**5c. PROGRAM ELEMENT NUMBER**

**6. AUTHORS**

Joseph DiVita, PhD
Robert L. Morris
Maria Olinda Rodas

**5d. PROJECT NUMBER**

**5e. TASK NUMBER**

**5f. WORK UNIT NUMBER**

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**

Space and Naval Warfare Systems Center Pacific (SSC Pacific)
San Diego, CA 92152–5001

**8. PERFORMING ORGANIZATION REPORT NUMBER**

TR 2030

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**

Space and Naval Warfare Systems Center Pacific (SSC Pacific)
San Diego, CA 92152–5001

**10. SPONSOR/MONITOR'S ACRONYM(S)**

SSC Pacific

**11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

**12. DISTRIBUTION/AVAILABILITY STATEMENT**

Approved for public release.

**13. SUPPLEMENTARY NOTES**

**14. ABSTRACT**

The coordination of actions and interactions between unmanned autonomous systems, manned systems, and a command group will be essential to accomplishing future missions. In this type of complex command and control scenario, operators will be subjected to vast amounts of information, resulting in information overload and loss of situation awareness. It is critical that designers develop predictive models of human and system performance to evaluate the adequacy of a system's design to satisfy specific mission requirements and ensure adequate operator performance. The effort described here applies queueing theory to a complex unmanned simulation. Data analysis suggests that there are several interesting and important areas of research that can be explored in order to improve the predictability of system performance using queueing theory.

**15. SUBJECT TERMS**

Mission Area: Command and Control,
Queueing Model; Supervisory Control; Unmanned Autonomous Vehicles

| **16. SECURITY CLASSIFICATION OF:** | | | **17. LIMITATION OF ABSTRACT** | **18. NUMBER OF PAGES** | **19a. NAME OF RESPONSIBLE PERSON** |
|---|---|---|---|---|---|
| **a. REPORT** | **b. ABSTRACT** | **c. THIS PAGE** | | | M. O. Rodas |
| | | | | | **19B. TELEPHONE NUMBER** *(Include area code)* |
| U | U | U | U | 38 | (619) 553–3646 |

# INITIAL DISTRIBUTION

| | | |
|---|---|---|
| 853 | Archive/Stock | (1) |
| 843 | Library | (2) |
| 53621 | M. O. Rodas | (6) |

| | |
|---|---|
| Defense Technical Information Center<br>Fort Belvoir, VA 22060–6218 | (1) |

Approved for public release.

Space and Naval Warfare Systems Center Pacific
San Diego, CA 92152-5001