TECHNISCHE UNIVERSITEIT DELFT
Faculteit Elektrotechniek, Wiskunde en Informatica

IN4390 – Quantitative Evaluation of Embedded Systems

practice exam, 3 hrs

Mitra Nasri, Koen Langendoen, Lydia Chen, Marco Zuniga

Question:	1	2	3	4	5	6	7	Total
Points:	8	15	10	5	10	10	10	68
Score:								

- This is a closed book exam
- You may use a **simple** calculator only (i.e. graphical calculators are not permitted)
- Write your answers with a black or blue pen, not with a pencil
- **Always justify your answers, unless stated otherwise**

- The exam covers the following material:
  (a) the paper "Basic Concepts and Taxonomy of Dependable and Secure Computing" by A. Avizienis ; J.-C. Laprie ; B. Randell ; C. Landwehr
  (b) chapters 18-20,22-23 (DoE), and 30-33 (Queueing Theory) of the book "The Art of Computer Systems Performance Analysis" by R. Jain
  (c) the paper "Petri nets: Properties, analysis and applications" by T. Murata
  (d) chapters 11.2 (DTMC), and 11.3 (CTMC) of the book "Introduction to probability, statistics, and random processes" by H. Pishro-Nik
  (e) the paper "Exploring Exploration: A Tutorial Introduction to Embedded Systems Design Space Exploration" by A.D. Pimentel

Operational Laws	
Utilization law	$U = XS$
Little's law	$N = XR$
Forced-flow law	$X_k = V_k X$
Bottleneck law	$U_k = D_k X$
Operational Bounds	
Througput	$X \leq \min\left(\frac{1}{D_{max}}, \frac{N}{D+Z}\right)$
Response time	$R \geq \max\left(D, N \times D_{max} - Z\right)$
Queueing Theory M/M/1	
Utilization	$U = XS = \lambda/\mu = \rho$
Probability of n clients in the system	$P_n = \rho^n(1-\rho)$
Mean #clients in the system	$N = \rho/(1-\rho) = \lambda/(\mu-\lambda)$
Mean #clients in the queue	$N_Q = N - \rho$
Mean response time	$R = N/\lambda = 1/(\mu-\lambda)$
Mean waiting time	$W = R - S = \rho/(\mu-\lambda)$
Basic Math	
Geometric series	$\sum_{k=0}^{\infty} r^k = \frac{1}{1-r}$, for $\lvert r \rvert < 1$

ANOVA Table for One Factor Experiments

Compo-nent	Sum of Squares	%Variation	DF	Mean Square	F-Comp.	F-Table
y	SSY=$\sum y_{ij}^2$		ar			
$\bar{y}_{..}$	SS0=$ar\mu^2$		1			
y-$\bar{y}_{..}$	SST=SSY-SS0	100	ar-1			
A	SSA $= r\Sigma\, \alpha_i^2$	$100\left(\frac{\text{SSA}}{\text{SST}}\right)$	a-1	MSA $= \frac{\text{SSA}}{a-1}$	$\frac{\text{MSA}}{\text{MSE}}$	$F_{[1-\alpha; a-1, a(r-1)]}$
e	SSE=SST- SSA	$100\left(\frac{\text{SSE}}{\text{SST}}\right)$	$a(r-1)$	MSE $= \frac{\text{SSE}}{a(r-1)}$		

Washington University in St. Louis CSE567M ©2008 Raj Jain

Question 1 [8 points]

Answer the following set of true/false questions on the special answer sheet that is machine readable. (Do not forget to fill out your credentials on the answer sheet.)

(a) $\boxed{\text{1 point}}$ The 'expected time until next failure' is an example of extrema quantitative measures. (**true**/**false**)

> **Solution: FALSE**: It is an example of reachability quantitative measures.

(b) $\boxed{\text{1 point}}$ Latency is a time delay between the cause and the effect of some physical change in the system being observed. (**true**/**false**)

> **Solution: TRUE**

(c) $\boxed{\text{1 point}}$ A dependable system will never occur any failure (**true**/**false**)

> **Solution: FALSE**: Dependability is about quantifying the severity and frequency of failure that can potentially happen in a system. A dependable system might have a failure, but as long that the severity and frequency of that failure remains below an acceptable threshold, the system remains dependable.

(d) $\boxed{\text{1 point}}$ Reliability is the readiness for correct service. (**true**/**false**)

> **Solution: FALSE**: It is the definition of availability.

(e) $\boxed{\text{1 point}}$ An incorrect internal state of a system is an error. (**true**/**false**)

> **Solution: TRUE**

(f) $\boxed{\text{1 point}}$ Absence of improper system alterations is not an attribute of dependability, rather it is an attribute of security. (**true**/**false**)

> **Solution: FALSE**: Absence of improper system alterations is integrity which is an attribute of both dependability and security

(g) $\boxed{\text{1 point}}$ A point X in the design space of an embedded system is said to Pareto-dominate another point Y if X does better than Y in all (performance/cost) dimensions. (**true**/**false**)

> **Solution: FALSE** as X must only excel in one dimension, and can be on par with Y in all other dimensions.

(h) $\boxed{\text{1 point}}$ Multi-objective optimization is a heuristic search method, and can therefore only approximate the true solution (i.e. the Pareto front). (**true**/**false**)

> **Solution: FALSE** as (i) MOO is **not** a heuristic and (ii) heuristics, typically, do well on small problems, and may be fortunate on big problems.

Question 2 [15 points]

Answer the following short questions.

(a) $\boxed{\text{5 points}}$ List all attributes of dependability.

Solution: Availability, reliability, safety, integrity, and maintainability

(b) $\boxed{\text{4 points}}$ List all means of dependability and security.

Solution: Fault prevention, fault removal, fault tolerance, fault forecasting

(c) $\boxed{\text{6 points}}$ Determine the type of "means of dependability and security" for each of the following actions.

(a) debugging the code by local program developers
(b) using a watchdog timer to reset a task when it hangs
(c) following coding standards
(d) using less hazardous programming languages (not like C++)
(e) simplifying the user interface
(f) using three different hardware that run the same program and then use a voter to pick the output that the majority of the computers agree on.

Solution:

(a) fault removal

(b) fault tolerance

(c) fault prevention

(d) fault prevention

(e) fault prevention

(f) fault tolerance

Question 3 [10 points]

A master student conducts a 2-level factorial randomized design. She has then constructed the following incomplete ANOVOA

Source	SS	DF	MS	F
A	350.00	2		
B	300.00		150	
AB	200.00		50	
Error	150.00	18		
Total	1000.00			

(a) 2 points How many levels of factor B did she use in the experiments?

Solution: MSB = SSB/DFB, so DFB = SSB/MSB = 300/150 = 2. Then factor B had 3 levels.

(b) 2 points How many degrees of freedom are associated with the interaction term?

Solution: There are (DFA-1)$\times$(DFB-1) = 2$\times$2 = 4 degrees of freedom

(c) 2 points How many replications has the student performed of each experiment?

Solution: DFE = ab(r-1), so r = 18/(3$\times$3) + 1 = 3 replications.

(d) 2 points What is the error mean square?

Solution: MSE=SSE/DFE = 150/18=8.33

(e) 2 points When the corresponding P values of factor A, B, and AB are 0.02, 0.04, and 0.15, respectively, which factors are significant?

Solution: Using a significance level of 5% (or 10%), factor A and factor B are significant

Question 4 [5 points]

Below is a design for measuring the job latency from three replicates based on 5 factors, A, B, C, D, and E.

A	B	C	D	E	latency		
−	−	−	−	−	7.78	7.78	7.81
+	−	−	+	−	8.15	8.18	7.88
−	+	−	+	−	7.50	7.56	7.50
+	+	−	−	−	7.59	7.56	7.75
−	−	+	+	−	7.54	8.00	7.88
+	−	+	−	−	7.69	8.09	8.06
−	+	+	−	−	7.56	7.52	7.44
+	+	+	+	−	7.56	7.81	7.69
−	−	−	−	+	7.50	7.25	7.12
+	−	−	+	+	7.88	7.88	7.44
−	+	−	+	+	7.50	7.56	7.50
+	+	−	−	+	7.63	7.75	7.56
−	−	+	+	+	7.32	7.44	7.44
+	−	+	−	+	7.56	7.69	7.62
−	+	+	−	+	7.18	7.18	7.25
+	+	+	+	+	7.81	7.50	7.59

(a) $\boxed{\text{4 points}}$ Write out the alias structure for this design. In other words, list all confounding factors.

Solution: Key is finding the generator: I=ABCD, which follows from "seeing" AB=CD (or any other alias).

The complete alias structure is then as follows: A=BCD, B=ACD, C=ABD, D=ABC, E=ABCDE, AB=CD, AC=BD, AD=BC, AE=BCDE, BE=ACDE, CE=ABDE, DE=ABCE, ABE=CDE, ACE=BDE, ADE=BCE

(b) $\boxed{\text{1 point}}$ what is the resolution of the above design?

Solution: The resolution follows from the generator: 4

Question 5 [10 points]

You have designed an embedded system and have run a series of trials to test it. The outcomes of these trials are either success or failure. During your evaluation, you noticed that if the two most recent trials were both successes, the next trial is a success with probability 0.8; otherwise, the chance of success is 0.5. In the long run, what proportion of trials are successes?

Solution:

- Build the state space, which consists of four states: SS, FS, SF, FF.
- Build the transition matrix

$$M = \begin{pmatrix} 0.8 & 0.0 & 0.2 & 0.0 \\ 0.5 & 0.0 & 0.5 & 0.0 \\ 0.0 & 0.5 & 0.0 & 0.5 \\ 0.0 & 0.5 & 0.0 & 0.5 \end{pmatrix}$$

- The matrix is ergodic (irreducible and aperiodic), thus we can use the steady state equations PI*M = PI and Sum PI = 1
- The result is PI_{SS} = 5/11, and 2/11 for the other three states.
- The proportion of successes is: 0.8*5/11 + 0.5*6/11= 7/11

Question 6 [10 points]

You have designed an embedded system that has two main components: A and B. The system has three states:

State 1 [S1]: both components are working.

State 2 [S2]: only component A is working.

State 3 [S3]: only component B is working.

There are no self-loops. From S1 you can go to either S2 (component A fails) or S3 (component B fails). From S2 and S3, you can only go back to S1 (i.e. the respective component is repaired).

(a) 4 points (Discrete Time Markov Chains) Assuming the transition probabilities given below, can we obtain a steady-state distribution? If your answer is Yes, provide the steady-state probabilities. If your answer is No, explain why.

S1→S2 = S1→S3 = 0.5
S2→S1 = S3→S1 = 1.0

Solution: The DTMC is irreducible but periodic, thus, it does not have a steady-state.

(b) 6 points (Continuous Time Markov Chains) In CTMCs, the chain does **not** need to be aperiodic to have steady-state probabilities. Considering that aperiodicity is not needed and assuming the rates given below, can we obtain a steady-state distribution? If your answer is Yes, provide the steady-state probabilities. If your answer is No, explain why.

S1→S2 = S1→S3 = λ_1
S2→S1 = S3→S1 = λ_2

Solution: For the CTMC, since periodicity does not matter only irreducibility, one can use PI*G=0 and Sum PI = 1, which leads to

$$S1 = \lambda_2/(2 * \lambda_1 + \lambda_2)$$

$$S2 = S3 = \lambda_1/(2 * \lambda_1 + \lambda_2)$$

Question 7 [10 points]

Operational laws are a crude, but useful tool to do back-of-the-envelope calculations about the performance benefits of adding hardware resources to a computing system. Consider a database server consisting of a CPU and two disks, whose performance characteristics are as follows:

t (measurement time)	650 seconds
C (#completions)	200 queries
B_{CPU} (busy time)	400 seconds
$B_{slow-disk}$ (busy time)	100 seconds
$B_{fast-disk}$ (busy time)	600 seconds
C_{CPU} (#completions)	22,200 jobs
$C_{slow-disk}$ (#completions)	2,000 jobs
$C_{gast-disk}$ (#completions)	20,000 jobs

(a) $\boxed{\text{3 points}}$ Compute the performance benefit of adding a second CPU (i.e., how many more database queries can be handled per second?)

Solution: Let's compute the demands using the bottleneck law:

- $D_{CPU} = B_{CPU}/C$ = 400 sec/200 queries = 2.0 sec/query
- $D_{slowdisk} = B_{slowdisk}/C$ = 100 sec/200 queries = 0.5 sec/query
- $D_{fastdisk} = B_{fastdisk}/C$ = 600 sec/200 queries = 3.0 sec/query

Thus the fastdisk is the bottleneck, hence, adding a second CPU has no effect!

(b) $\boxed{\text{5 points}}$ Compute the maximum performance benefit that can be gained by balancing the load of the slow and fast disk by shifting database files from one to the other.

Solution: This requires more work as we need to see how often a query visits each disk, and what the respective service times are:

- $E[V_{slowdisk}] = C_{slowdisk}/C$ = 2,000 visits/200 queries = 10 visits/query
- $E[S_{slowdisk}] = B_{slowdisk}/C_{slowdisk}$ = 100 sec/2,000 visits = 0.05 sec/visit
- $E[V_{fastdisk}] = C_{fastdisk}/C$ = 20,000 visits/200 queries = 100 visits/query
- $E[S_{fastdisk}] = B_{fastdisk}/C_{fastdisk}$ = 600 sec/20,000 visits = 0.03 sec/visit

Now we want to balance the visits over the disks to equalize the times spent at each disk:

- $V_{slowdisk} + V_{fastdisk} = 110$
- $S_{slowdisk} \times V_{slowdisk} = S_{fastdisk} \times V_{fastdisk}$

Solving the above set of equations leads to $V_{slowdisk} = 41.25$ and $V_{fastdisk} = 68.75$. Thus equalizing the demand at $D_{slowdisk} = D_{fastdisk} = 2.06$, showing that balancing the disks pays off with a factor of 1.45 (= 3/2.06).

(c) $\boxed{\text{2 points}}$ Consider the alternative approach of adding a second fast disk. How much can performance be improved this way?

Solution: Adding a second fast disk is (at best) doubling the throughput of the combo, thus reducing the demand to $D_{fast1} = D_{fast2}$ = 3.0 / 2 = 1.5 sec/query. The result is that the CPU now becomes the bottleneck, so D_{max} reduces from 3.0 (bottleneck = fastdisk) to 2.0 (bottleneck = CPU) or, in other words, performance is increased with a factor 1.5.