

Klausur zur Veranstaltung
“**Industrielle Produktionssysteme**”
im Wintersemester 2015/2016

Hinweise:

- Die Klausur besteht aus **14** Seiten (inkl. Deckblatt). Bitte überprüfen Sie, ob Ihr Exemplar komplett ist und lassen Sie sich ansonsten ein anderes geben.
- Alle Aufgaben in der Klausur sind zu bearbeiten.
- Für jede Aufgabe sind die zu erreichenden Punkte angegeben. Bei einer Klausurdauer von 60 Minuten sind maximal insgesamt 60 Punkte zu erreichen.
- **Der Lösungsweg muss erkennbar sein!** Wenn Sie zur Beantwortung einer Frage eine Formel verwenden, so geben Sie diese zunächst in allgemeiner Form an!
- Als Hilfsmittel sind ein Taschenrechner, ein nicht vernetzter PC mit vorgegebenen Scilab-Programmen und ein beidseitig beschriebenes Hilfsblatt (DIN A4) erlaubt.
- **Wichtig:** Wenn Sie Berechnungen mit Scilab o.ä. durchführen, dann geben Sie bitte auch die **Eingabematrizen und -vektoren** an!!!
- Zur Beantwortung der Fragen finden Sie genügend Platz in der Klausur. Bitte reißen Sie die Klausur nicht auseinander und verwenden Sie kein eigenes Papier.
- Tragen Sie bitte zuerst Ihre persönlichen Daten ein.

Persönliche Daten:

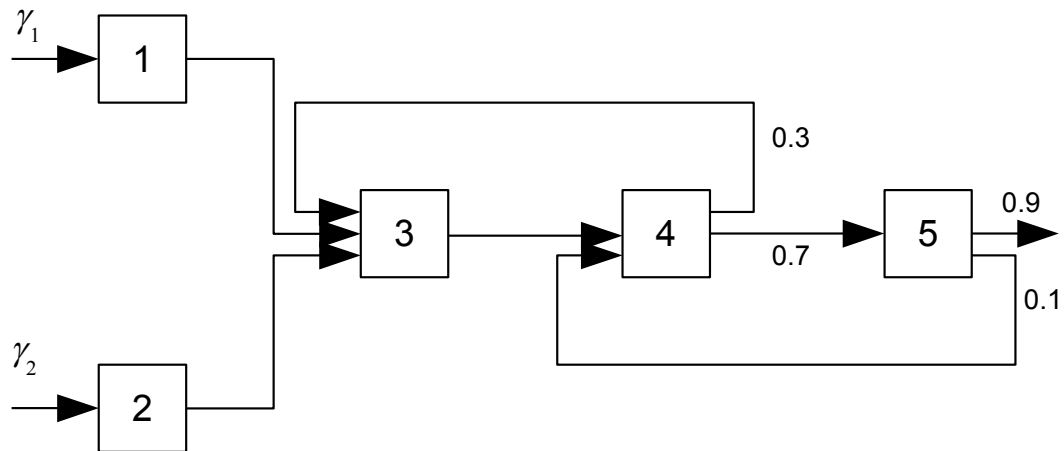
Nachname	Vorname	Matrikelnr.	Studienfach	Semester

Bewertung:

Aufg.	1	2	Summe
Punkte			

1. Leistungsanalyse von offenen mehrstufigen Ein-Produkt-Produktionssystemen mit unbegrenzten Puffern (30 P.)

- (a) **Analyse eines offenen Ein-Produkt-Netzwerks.** Analysieren Sie das in der folgenden Abbildung dargestellte Netzwerk aus Arbeitsstationen mit jeweils einem Server zur Herstellung einer einzigen Produktart.



Die externen Ankunftsraten betragen $\gamma_1 = 6/h$ und $\gamma_2 = 10/h$. Die Zwischenankunftszeiten im Ankunftsstrom 1 sind konstant, jene im Ankunftsstrom 2 sind exponentialverteilt. Die Bearbeitungsdauern an allen Stationen sind gammaverteilt. Die Erwartungswerte $E[T_s(i)]$ und die quadrierten Variationskoeffizienten $c_s^2(i)$ der Bearbeitungszeiten an den einzelnen Stationen entnehmen Sie der folgenden Tabelle:

Station i	$E[T_s(i)]$ [h]	$c_s^2(i)$
1	1/10	0,8
2	1/12	1,2
3	1/25	0,8
4	1/30	0,9
5	1/22	0,2

- i. Berechnen Sie (ggf. unter Verwendung eines geeigneten Scilab-Programms) für jede Station die Ankunftsrate λ_i , den quadrierten Variationskoeffizienten $c_a^2(i)$ der Zwischenankunftszeit, die Durchlaufzeit $CT(i)$ und den Bestand $WIP_s(i)$. Wie groß ist der Durchsatz des Systems? (15 P.)

ii. Geben Sie an, welche Station den Engpass des Systems darstellt und wie stark dieser ausgelastet ist. (2 P.)

(b) **Erläutern** Sie,

i. was man unter einem Erneuerungsprozess versteht (3 P.),

ii. welche Bedeutung das Konzept eines Erneuerungsprozesses für die mathematische Modellierung der Ankunftsströme an den Stationen eines offenen Ein-Produkt-Netzwerkes hat (4 P.),

iii. an welchen Stationen die Ankunftsströme Erneuerungsprozesse darstellen (3 P.)

iv. und welche Konsequenzen dies für die Frage hat, ob die Ergebnisse der Analyse exakt sind oder nicht. (3 P.)

2. Leistungsanalyse von Fließproduktionssystemen mit begrenzten Pufferkapazitäten (30 P.)

Betrachten Sie in dieser Aufgabe Fließproduktionssysteme der Art, die in der folgenden Abbildung dargestellt wird:



Die Quadrate stellen Arbeitsstationen mit exponentialverteilten Bearbeitungszeiten dar, die Kreise symbolisieren Puffer von begrenzter Aufnahmekapazität für die Werkstücke, die das System in Richtung der Pfeile durchlaufen. Die erste Maschine hungert nie, die letzte ist nie blockiert.

Zur Analyse derartiger Systeme wurde Ihnen ein Scilab-Code vorgestellt, den Sie in dem folgenden Listing vorfinden:

```
1  clc
2  clear
3  // Einfacher Dekompositionsansatz
4  // Exponentialverteilte Bearbeitungszeiten, begrenzte Pufferkapazitaeten
5
6  // Autor: Stefan Helber
7  // Datum: 21.1.10
8
9  // Berechnung aller Kenngroessen eines M/M/1/K-Systems
10 function [p0, pK, lambdaeff, rhoeff, ELQ, ELS] = MMIK(lambda, mue, K)
11     rho=lambda/mue;
12     rhoHsum=0;
13     for n=0:K,
14         rhoHsum=rhoHsum+rho^n;
15     end
16     p0=1/rhoHsum;
17     pK=rho^K*p0;
18     ELQ=0;
19     for n=1:K,
20         ELQ=ELQ+(n-1)*rho^n*p0;
21     end
22     lambdaeff=(1-pK)*lambda;
23     ELS=ELQ+lambdaeff/mue;
24     rhoeff=lambdaeff/mue;
25 endfunction
26
27 AnzMasch=10;
28 AnzMasch
29
30 // Bearbeitungsraten aller Maschinen
31 mue=[10 10 10 10 10 10 10 10 10 10];
32 mue
33
34 // Pufferkapazitaeten zwischen den Maschinen
35 %buffcap=[0 0 0 0 0 0 0 0 0];
36 %buffcap=[2 2 2 2 2 2 2 2 2];
37 %buffcap=[5 5 5 5 5 5 5 5 5];
```

```

38 %buffcap=[10 10 10 10 10 10 10 10 10];
39 %buffcap=[100 100 100 100 100 100 100 100 100];
40
41 buffcap=2*[1 1 1 1 1 1 1 1 1];
42 buffcap
43
44 // Bearbeitungsraten der virtuellen Upstream-Maschinen initialisieren
45 mueup=mue(1:AnzMasch-1);
46 mueup
47 // Bearbeitungsraten der virtuellen Downstream-Maschinen initialisieren
48 muedn=mue(2:AnzMasch);
49 muedn
50
51
52 // Initialisierung der Kenngrößen der virtuellen Systeme
53 for n=1:AnzMasch-1,
54 [p0(n), pK(n), PR(n), rhoeff(n), ELQ(n), ELS(n)]=MMIK(mueup(n),muedn(n),buffcap(n)+2);
55 end
56
57 // Nun Dekomposition
58 // dazu Repeat-Anweisung emulieren!
59 Zaehler=0
60 NichtFertig=%T
61 while NichtFertig==%T,
62 // Vorwaertsrechnung
63 for n=2:AnzMasch-1,
64 mueup(n)=1/(1/mue(n)+1/PR(n-1)-1/muedn(n-1));
65 [p0(n), pK(n), PR(n), rhoeff(n), ELQ(n), ELS(n)]=MMIK(mueup(n),muedn(n),buffcap(n)+2);
66 end
67 // Rueckwärtsrechnung
68 for n=AnzMasch-2:-1:1,
69 muedn(n)=1/(1/mue(n+1)+1/PR(n+1)-1/mueup(n+1));
70 [p0(n), pK(n), PR(n), rhoeff(n), ELQ(n), ELS(n)]=MMIK(mueup(n),muedn(n),buffcap(n)+2);
71 end
72 Zaehler=Zaehler+1
73 if abs(PR(1)-PR(AnzMasch-1))<0.000001,
74 NichtFertig=%F
75 end
76
77 end
78 // Durchsatz
79 PR
80 // Bestand je Puffer
81 ELS
82 Zaehler
83 Gesamtbestand=0;
84 for n=1:AnzMasch-1,
85 Gesamtbestand=Gesamtbestand+ELS(n);
86 end
87
88 Gesamtbestand

```

- (a) Erläutern Sie zunächst anhand einer Skizze die Grundidee des in der Vorlesung dargestellten Dekompositionsverfahren zur Leistungsanalyse derartiger Systeme!
(6 P.)

- (b) Erläutern und begründen Sie die verschiedenen in dem Dekompositionsansatz verwendeten Gleichungen (6 P.)

- (c) Erläutern Sie auf dieser Basis den Algorithmus in dem oben angegebenen Scilab-Code und verweisen Sie dabei zur Erläuterung der einzelnen Schritte auf die jeweiligen Nummern links neben dem Programmcode. (6 P.)

- (d) Im folgenden sind Ihnen einige Systemkonfigurationen derartiger Systeme gegeben, jeweils unter Angabe der (isolierten) Bearbeitungsrate μ_i der Stationen $i = 1, \dots, I$ und der Kapazität des Puffers C_i hinter der Station. Erläutern Sie **anhand prinzipieller Überlegungen** jeweils, aus welchem Grund Sie die jeweilige Pufferallokation für wirtschaftlich sinnvoll halten oder nicht. (Berechnungen mit Scilab sind nicht erforderlich!) (10 P.)

- System A: (2 P.)

Station i	μ_i	C_i
1	10	6
2	10	5
3	10	4
4	10	6
5	10	6
6	10	-

- System B: (2 P.)

Station i	μ_i	C_i
1	10	1
2	10	3
3	10	6
4	10	9
5	10	12
6	10	-

- System C: (2 P.)

Station i	μ_i	C_i
1	10	4
2	11	4
3	12	4
4	12	4
5	11	4
6	10	-

- System D: (2 P.)

Station i	μ_i	C_i
1	8	10
2	10	6
3	10	3
4	10	3
5	10	3
6	10	-

- System E: (2 P.)

Station i	μ_i	C_i
1	10	2
2	13	8
3	14	8
4	16	8
5	15	2
6	11	-

- (e) Stellen Sie sich nun vor, die Stationen würden sich hinsichtlich des Variationskoeffizienten der Bearbeitungszeiten unterscheiden. Welchen Effekt hätte dann *ceteris paribus* ein Anstieg des Variationskoeffizienten der Bearbeitungszeit an einer einzelnen Station i auf die Struktur der optimalen Pufferallokation bei einer gegebenen Anzahl von Pufferplätzen? (2 P.)