MONITORING AND INTERPRETING MULTISTAGE AND MULTICATEGORY PROCESSES

by

RODRIGO IGNACIO DURAN LOPEZ

A dissertation submitted to the

Graduate School-New Brunswick

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Doctor of Philosophy

Graduate Program in Industrial and Systems Engineering

Written under the direction of

Dr. Susan L. Albin

And approved by

_____

_____

_____

_____

New Brunswick, New Jersey

October, 2009

ABSTRACT OF THE DISSERTATION

Monitoring and Interpreting Multistage and Multicategory Processes
By RODRIGO IGNACIO DURAN LOPEZ

Dissertation Director:

Dr. Susan L. Albin

Consider processes where a transaction moves through stages and falls within a category at each stage. For example, in a tax complaint process, the stages are the steps taxpayers follow to resolve a property tax dispute from initial complaint through final resolution. The primary motivation here is customer service, although the transactions could be related to manufacturing applications as well.

The main contribution here is a method to monitor the fractions and numbers of transactions within and across stages of multistage and multicategory processes, a problem that has not been formulated before in the literature. The proposed method not only signals an out-of-control situation, it identifies accurately and easily which stages and categories are causing the disturbance, providing interpretations within and across stages of the process.

The proposed methodology works as follows: If a multinomial distribution fits the number of transactions in each category at every stage, then the process is decomposed into single stages that are monitored separately, and finally into independent binary substages with two categories. Each binary substage is characterized by a conditional probability and monitored with an independent fraction, called a tree fraction. The number of tree fractions that are monitored depends on the number of final categories, i.e., those that do not split in any further categories, not on the number of stages.

Two other contributions, summarized next, address the single stage case. Each is useful by itself, and each contributes to the method for the multistage case as well.

The first is a new two-sided CUSUM Arcsine method to monitor a process with two categories. The second is the *p-tree* method that monitors a multinomial process. The *p-tree* method not only signals an out-of-control situation, it identifies accurately which categories are causing the problem, in contrast to the widely used method in Marcucci (1985).

Future research would cover monitoring other types of multistage processes in service. An application of using probability trees to test and interpret associations in contingency tables is envisioned.

Acknowledgements

There are many people, lucky episodes, and removed roadblocks that are enabled me to pursue doctoral studies. I want to thank Professor Susan Albin for guiding me in this process from developing a research subject to the careful task of writing articles for publication. I am grateful that the Department of Industrial and Systems Engineering hired me as a teaching assistant during several years, providing me financial support, but specially for allowing me to participate in the teaching process. I thank Dr. Jolie Cizewski and Dr. Evelyn Erenrich both of the Graduate School for their key support and encouragement at the beginning of my doctoral studies.

My external committee member, Professor Michael Lahr from the Center for Urban Policy Research has supported me in many ways, opening new research perspectives and encouraging me in the tough times. The members of my dissertation committee, Professor Elsayed Elsayed, and Professor Art Chaovalitwongse have generously given their time and expertise to better my work. I thank them for their contribution and their good-natured support.

I can share a few lessons learned: it is possible to earn a PhD with just a nonprogrammable calculator and even without a computer at home, which was my case sometimes. However, it is not possible to earn a PhD without the guidance of a caring and expert advisor and the example of the Department's professors. Additionally, in my personal case, a network of transformational counselors was a key support in this journey.

I dedicate this dissertation to my family, specially my two loving daughters.

Rodrigo Durán

Table of Contents

List of Appendices

List of Tables

List of Figures

1 Introduction

Consider a transaction process that occurs in one or more stages. For example, in a tax complaint process the stages are the steps taxpayers follow to resolve a property tax dispute from initial complaint through final resolution. The transactions can be related to customer service, as in the tax complaint process, or to manufactured products. However, the primary interest here is in the customer service area. The work is motivated by the previous experience of this student who served as director for the national administration of the property tax in Chile, including management responsibilities over the customer service systems.

One goal of the management of a service organization is monitoring the fraction and number of transactions that fall into multiple categories at each of multiple stages. Often this data is presented to managers in its raw form with some fractions reported. A monitoring system would allow the management to identify and respond to unusual occurrences and also to introduce improved procedures to make the system operate more efficiently by improving training or modifying the IT system or changing staffing.

The main contribution here is a methodology to monitor the fractions and numbers of transactions within and across stages of multistage and multicategory processes, a problem that has not been formulated before in the literature. The method not only signals when the fractions in the multiples stages and categories have changed significantly, it indicates which stages and categories are causing the disturbance allowing management to interpret the signal. Further, the method results in the desired false alarm rate.

Two other contributions in this dissertation address monitoring the fractions in single-stage processes. These are each useful individually and also contribute to the multistage case described above, which is based on decomposing the multiple-stages and categories into single-stages that are monitored separately.

The first single-stage method is proposed for monitoring the fraction in each category in a single-stage with two categories - called here a binary process. This new method is needed because the well-known *p*-chart often does not achieve the desired false alarm rate even when the sample size is very large and one would expect a good normal approximation for the number in each category. The literature does contain other methods to overcome the problems with achieving the desired false alarm rate as described in Section 2.1. However, these methods require complex steps to calculate the control limits including published tables, simulation, or Markov chain analysis. The method for monitoring the binary process is in Duran and Albin (2009b), in print *at Quality and Reliability Engineering International*.

The method developed is the CUSUM Arcsine method in which the data is preprocessed using an arcsine normalizing transformation for a binomial distributed variable and then monitored with a two-sided CUSUM method.

The second single-stage method is for monitoring fractions in processes with three or more categories - called here a multinomial process. The principal advantage of the method developed here, called the *p-tree* method, is its usefulness as a diagnostic tool. The *p-tree* method monitors both nominal and ordinal categorical data and allows any number of categories. This new method is needed because the existing methods are able to signal when the fractions among the categories have changed but they do not indicate

which ones are problematic, i.e., they cannot help to interpret. The method for monitoring a single-stage multicategory process is in Duran and Albin (2009a), in print at *IIE Transactions*.

The *p-tree* method developed here transforms a multinomial process with $K>2$ categories into a binary probability tree with K-1 independent binary substages, in which each substage has two categories. The independence is based on Johnson *et al.* (1996, p. 68) and Kemp and Kemp (1987). Each binary substage is monitored with an independent control chart for binomial distributed data.

The *p-tree* method indicates easily which substages are responsible in case of an out-of-control signal. Each binary substage could be monitored with the familiar and simple to use *p*-chart based on the binomial distribution. However, the *p*-chart often does not result in the desired false alarm rate. To solve this problem, we propose to monitor each binary substage with the proposed CUSUM Arcsine method.

1.1 Overview of property tax complaint process

This work was motivated by the tax complaint process mentioned earlier. The property tax assessment process starts when local offices process input data from municipalities and deeds offices such as construction permits, lot subdivisions permits, occupancy permits, real estate transfers, and taxpayer requests for assessment. An assessor evaluates the property. Then the assessment data is sent to computational processing to update databases. Legislated fiscal price tables are invoked to determine the assessed value. A batch program processes a log of new assessments in order to generate and mails notices to the taxpayers.

A taxpayer with a problem passes through a tax complaint process, which is shown in Figure 1.1 as a multistage and multicategory process. There are four stages. Assessment notices are sent to the taxpayers and taxpayers split into two categories: the taxpayers consult an assessment advisor to discuss whether there is a realistic complaint or do not consult. In stage two, among taxpayers who consult, an assessor gives front desk advice for the case and the taxpayers split into three categories: the assessor may advise taxpayers file an official complaint or not, or that maybe a complaint might be useful. In stage three, some taxpayers will file a complaint and others will not file. In stage four, filed complaints are investigated and the assessors' office makes a final resolution that falls within one of three categories: the resolution is that the assessment is correct, that it is too high or that it is too low. Rafool (2002) contains an overview about this tax in the United States, and The New Jersey Property Tax Assessment Study Commission (1986) describes the methods and makes recommendations, which are still valid, about the administration of this tax in New Jersey.

As a whole, the monitoring system can indicate changes in the quality of the assessment decisions, the performance of the complaint processes, and even help to predict changes in the tax revenues. The assessment process is subject to errors – incorrect assessments against the taxpayer or in favor of the taxpayer. Changes in the error rate affect each stage in the complaint process. Changes in the tax administration responses affect both the front desk and the final resolution stages. Keeping the process in control as well as reducing the errors would lead to a more efficient and equitable tax system.

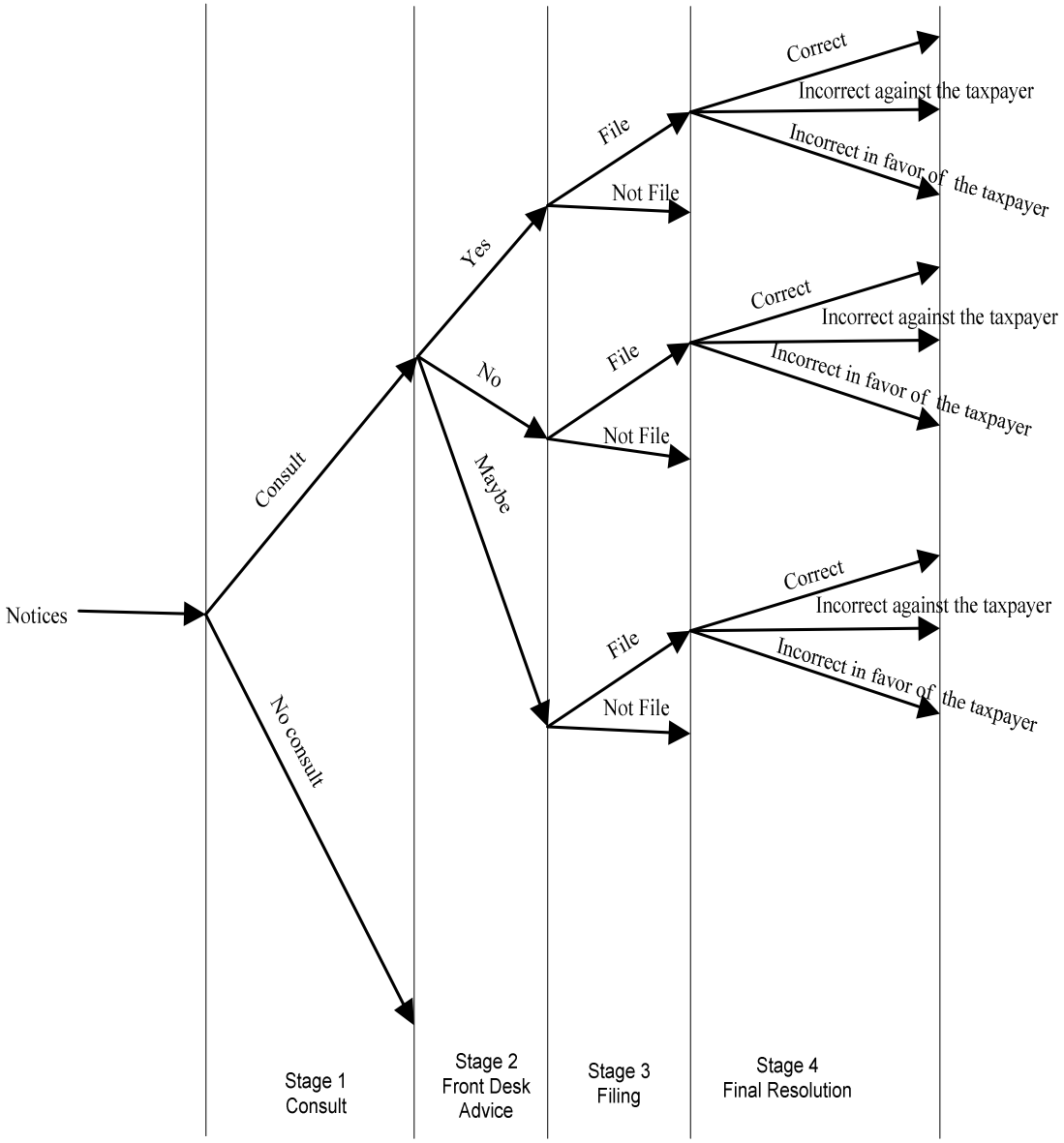

Figure 1.1. Multistage property tax complaint process as a tree diagram

1.2 Additional details about methods proposed

We now give more details about the three methods proposed. We start with the CUSUM Arcsine method for monitoring single-stage processes with two categories. The new method achieves the desired false alarm rate for any sample size N and baseline probability $0.005<p_0<0.995$ such that $E[N]p_0(1-p_0)\geq 3$. The N may be constant or Poisson distributed. This rule works with combinations of N and p_0 in which the normal approximation does not hold, and also in combinations in which the normal approximation should work but fails as explained next.

There exist several rules of thumb to predict when the *p-chart* will achieve the desired false alarm rate based on predicting when the normal approximation to the binomial works well. Schader and Schmid (1989) study two well known rules regarding the normal approximation to the binomial: rule 1 requires that $Np_0(1-p_0)>9$ and rule 2 requires that $Np_0>5$ and also $N(1-p_0)>5$. Table 1.1 shows three examples where both rules of thumb hold (some by a very wide margin) but the resulting *p-charts* do not achieve the desired false alarm rate, or equivalently the desired in-control average run length, ARL_0 (which equals the inverse of the false alarm rate for independent samples). The reason the *p-charts* fail is that the normal approximation performs poorly in the tails of the binomial distribution, as pointed out by Ryan and Schwertman (1997, p. 66). Notice that these sample sizes are very large and we would certainly expect the normal approximation to work.

		Rule 1	Rule 2	
N	p_0	$Np_0(1-p_0)>9$	$Np_0>5$	ARL_0
200	0.1	18>9	20>5	294
600	0.1	54>9	60>5	441
1000	0.01	10>9	10>5	300

Table 1.1. Three-sigma *p*-chart has difficulties achieving desired ARL_0 of 370

There are existing methods that successfully achieve the desired false alarm rate but they are difficult to design because require published tables, simulation, or Markov chain analysis. These include binomial-based EWMA chart in Gan (1990), binomial-based CUSUM chart in Gan (1993), CUSUM *Q*-chart and EWMA *Q*-chart in Quesenberry (1995), and binomial based (modified) CUSUM chart in Reynolds and Stoumbos (2000, 1999). In addition, there exist methods that are easy to design, but these fail to consistently achieve the desired false alarm rate. A thorough review of this literature appears in Section 2.1.

We now give more detail about the *p-tree* method for a single-stage multiple categories case. First we show it can be quite difficult to interpret a signal in a multinomial process that is out of control. Marcucci (1985) gives data where samples of finished bricks are classified into conforming, nonconforming type *A*, and nonconforming type *B* categories with baseline probabilities 0.95, 0.03, and 0.02 respectively. Table 1.2 shows simulated data of two significantly out-of-control samples.

	Fraction		
	conforming	*nonconforming A*	*nonconforming B*
Baseline	.950	.030	.020
Sample 1	.960	.014	.026
Sample 2	.932	.034	.034

Table 1.2. Two out-of-control samples of finished bricks

Table 1.2 demonstrates that is difficult to interpret the results. For sample 1, is the out-of-control condition caused by an increase in the fraction of conforming bricks or is it caused by a decrease in the fraction of nonconforming type *A* bricks? For sample 2, does the decrease in the fraction of conforming bricks cause the out-of-control condition or is it a problem with the ratio of type *A* versus type *B* nonconforming bricks?

The reason because it is difficult to interpret which category is causing the out-of-control is that as one count increases then the sum of the other two decreases, and vice-versa; i.e., the numbers are negatively correlated. A multinomial process can be monitored with a control chart in Marcucci (1985), which has been widely used. Marcucci's method plots a Pearson statistic, and signals when the current sample significantly differs from baseline. However, Marcucci's method does not indicate which categories are causing the disturbance.

For the bricks problem, Marcucci (1985) suggests interpretation as follows: discard one category and construct a modified *p*-chart based on the remaining two. However, it is not clear which category should be discarded and Marcucci (1985, p. 89) suggests that this method is restricted to at most three categories.

In contrast, the *p-tree* method developed here transforms the multinomial process into several independent binary substages and this assists in interpreting where the problem is. For the Marcucci (1985) bricks example, a probability tree with two binary substages is constructed to represent the three categories. The first substage monitors the fraction of conforming bricks out of the total sample. The second substage monitors the fraction of nonconforming type *A* bricks over the nonconforming bricks. The fractions monitored in the two substages are independent. We call these two independent fractions the "tree fractions".

The *p-tree* method helps answer our questions about what caused the out-of-control signals for the samples in Table 1.2. For sample 1, the fraction conforming brick is consistent with baseline, but among nonconforming, type *A* is underrepresented. For

sample 2, the fraction conforming is low compared to baseline and the fraction of type *A* among nonconforming is consistent with baseline.

Simulation studies in Section 3.3 compare the *p-tree* method to the Marcucci method. The *p-tree* method gives accurate interpretations to determine which categories are responsible for the signal, something that the Marcucci cannot do at all. Also, the sensitivity is comparable to that of the Marcucci method.

The method allows the user to order the categories according to their monitoring importance. This is relevant because the way in which we order the categories to transform the single-stage multinomial process into a binary probability tree affects the monitoring capability. Those categories at earlier substages will have larger samples sizes, resulting in better sensitivities and diagnosis accuracies than later substages.

We now discuss the principal contribution of this thesis, the method to monitor and interpret the fractions in multiple categories across multiple stages. Here are the basic steps of the methodology: (1) Construct a multinomial probability tree for the process (for an example, see Figure 1.1) such that there is only one path to reach each category and all splitting of categories are identified. (2) Apply the *p-tree* method to each splitting to convert the multinomial probability tree into a binary probability tree. (3) Identify the binary substages and the "tree fractions", which are the independent fractions that we will actually monitor. (4) Construct CUSUM Arcsine control charts for each tree fraction.

To facilitate the software implementation of the multi-stage multi-category monitoring method, we express the procedure in matrix notation. This is especially critical for larger systems since it is difficult to track, maintain, and update all the data. In Chapter 5, we start with a business process diagram of a call center described in

Mandelbaum *et al.* (2001) and show the equivalent multinomial probability tree, the transformed binary probability tree, and the control charts. The case study also illustrates the matrix representation of the method.

In Chapter 6 future research is described. This includes monitoring multistage multicategory processes with multinomial assumptions that do not hold. An example of an important application of this would be monitoring routing matrices in queuing systems. Another area of future work is in forecasting and in testing and interpreting associations in contingency tables using trees.

The rest of this dissertation is organized as follows: Chapter 2 contains a literature review; Chapter 3 presents monitoring single-stage processes with multiple categories; Chapter 4 presents monitoring a fraction with easy and reliable settings of the false alarm rate; Chapter 5 addresses monitoring multistage and multicategory processes; Chapter 6 describes future research, and Chapter 7 concludes.

2 Literature review

This Chapter reviews the literature related to the following issues:

- Monitoring a fraction in binomial distributed data
- Monitoring processes with multiple categories
- Monitoring processes in the service industry and healthcare
- Monitoring multistage processes

2.1 Literature review about monitoring a fraction in binomial distributed data

The main goal of this review is identify which existing methods might be easily designed to achieve a desired false alarm rate (α) when monitoring fractions of binomial distributed data. Table 2.1 summarizes the methods that have been proposed to monitor a fraction, mostly in manufacturing. This review complements Woodall (1997), which was a comprehensive review about monitoring attribute data. The first column of Table 2.1 identifies the type of method. The second and third columns show the method's name and its reference. The fourth column shows the degree of difficulty of the designing procedure: "not easy" means that the parameters are calculated using any of the following techniques: consulting tables (Tables), combinatorial or enumerative methods (Comb), extensive simulation (Sim), or Markov chain analysis (MC). The label "easy" means that the parameters are calculated in simple steps, without using the latter techniques. The fifth column indicates whether the method is two-sided, i.e., able to monitor both increases and decreases. The last column indicates whether the authors in the reference show that the method achieves a desired α. We also test through simulation whether the methods that have an easy design actually achieve the desired α (Section 4.2).

Regarding the first column of Table 2.1, we distinguish six types of methods for monitoring a count or a fraction:

- *p*-chart, *np*-chart, and modifications: methods that plot the fraction or count of interest.
- Shewhart charts on transformed fraction: plot a normalizing transformation of the fraction with a Shewhart chart for individual observations.
- CUSUM or EWMA on fraction: applied on a fraction or count without transformations.
- CUSUM or EWMA on transformed fraction: preprocess the fraction with a transformation and then monitor it using a CUSUM or a EWMA method.
- Run rules for detecting increases in a fraction in manufacturing.
- Run rules for detecting decreases in a low fraction in manufacturing.

In the rest of this Section, we give additional comments on each type of method. The first type of method includes the modified *p*-chart of Chen (1998) and the modified *np*-chart of Shore (2000). The two latter methods modify the control limits of a *p*-chart or an *np*-chart in order to get similar values to the probability limits that would be obtained using the exact binomial distribution (more details in Appendix E). In this first type of methods we include control charts whose design is combinatorial such as Ryan and Schwertman (1997) and Schwertman and Ryan (1999), which for a given value of p_0, search for values of N and control limits such that the actual ARL_0 gets very close to 370. Acosta-Mejia (1999) and Wu *et al.* (2006) can also be seen as combinatorial procedures.

All the methods classified in the second type have an easy design. The Arcsine transformation (described and used in next Section) is specific for binomial data and is approximately standard normally distributed for any values of N and p_0. The Box-Cox is a power transformation (described in Appendix D) that requires a power parameter in order to minimize the skewness of the transformed data, not requiring knowledge of the data's distribution. The Q transformation (described in Appendix D) is approximately standard normally distributed, and uses the inverse of the cumulative binomial distribution.

Here we comment on methods classified in the third type. The design of a binomial based CUSUM chart is treated initially by Gan (1993). Additionally, Hawkins and Olwell (1998) give complete foundations of the CUSUM method, and propose a CUSUM for binomial data. These CUSUM methods usually require solving a Markov chain in order to calculate their parameters. Reynolds and Stoumbos (1999) propose a modified CUSUM for Bernoulli data, i.e., with N=1 as a sample size. This method requires solving a system of three equations to design the parameters of each side of the CUSUM (upper side and lower side), i.e., a total of six equations for a two-sided method. This method has shown better sensitivity than the binomial based methods, and can work well in manufacturing applications.

Regarding the fourth type of methods, Quesenberry (1995) is the first author that proposes monitoring a fraction using a CUSUM on a normalizing transformation (he uses the Q transformation). He also proposes a EWMA on a Q transformation. Our proposed methods, the CUSUM Arcsine and the CUSUM Box-Cox also fall in this type of methods.

Type of Method	Method	Reference	Design	Two-sided	Achieves α
p-chart, *np*- chart, and modifications	Shewhart *p*-chart	Montgomery (2005)	Easy	Yes	In some cases
	np-chart exact prob limits	Montgomery (2005)	Not easy (Comb)	Yes	In some cases
	Modified *p*-chart	Chen (1998)	Easy	Yes	In some cases
	Modified *np*-chart	Shore (2000)	Easy	Yes	In some cases
	Randomized *np*-chart	Wu *et al.* (2001)	Not easy (Sim)	Yes	Yes
	Modified *p*-chart or *np*-chart	Ryan and Schwertman (1997), Schwertman and Ryan (1999), Acosta-Mejia (1999)	Not easy (Comb)	Yes	In some cases
	Modified *np*-chart	Wu *et al*. (2006)	Not easy (Comb)	Only increases	In some cases
Shewhart charts on transformed Fraction	*Q* chart	Quesenberry (1991)	Easy	Yes	In some cases
	Arcsine chart	Chen (1998)	Easy	Yes	In some cases
	Box Cox chart	Only suggested	Easy	Yes	In some cases
CUSUM or EWMA on Fraction	EWMA for Binomial	Gan (1990)	Not easy (MC)	Yes	Yes (mfg)
	CUSUM for Binomial	Gan (1993), Hawkins and Olwell (1998), Bourke (2001), Reynolds and Stoumbos (2000, 1999)	Not easy (MC/Tables)	Yes	Yes (mfg)
	CUSUM for Bernoulli	Bourke (2001), Reynolds and Stoumbos (2000, 1999)	Not easy (MC)	Yes	Yes (mfg)
CUSUM or EWMA on Transformed Fraction	CUSUM *Q*, EWMA *Q*	Quesenberry (1995)	Not easy (MC)	Yes	Yes (mfg)
	New CUSUM Arcsine	**Proposed here**	Easy	Yes	Yes
	New CUSUM Box-Cox	**Proposed here**	Easy	Yes	Yes
Run rules for increases	Unit and groups-run charts	Wu and Jiao (2007), Gadre and Rattihalli (2005)	Not easy (Tables /MC)	Only increases	In some cases
Run rules for decreases	Based on Negative Binomial or Geometric distributions	Schwertman (2005), Chan *et al.* (2003, 2002), Liu *et al.* (2007, 2006), Lucas *et al.* (2006)	Not easy (Tables /MC/Comb)	Ussually decreases	In some cases

Table 2.1. Methods for monitoring a fraction by type of method and design

(Comb=combination or enumeration, Sim=simulation, MC=Markov chain)

The methods in the fifth type monitor increases in fractions using run rules. The sampling in these methods switches from unit-level inspection (N=1, higher sensitivity) to group-level inspection (N>1, lower sensitivity) and back to the unit-level inspection according to specified rules. The authors of these methods claim that their charts improve the sensitivity when monitoring increases in fractions.

The methods in the sixth type for monitoring a low nonconforming fraction typically consider the geometric or the exponential or the negative binomial distributions. For example, Lucas *et al.* (2006) propose a method based on run rules to detect process improvement when the lower limit of an *np*-chart is zero. When combined with the upper control limit of an *np*-chart can offer a two-sided feature. This method can work well to detect rare events although its design is still "not easy" because it requires enumeration.

In Section 4.2, the identified easy to design methods are compared using simulation. Notice that existing methods that achieve the desired α are not easy to design.

2.2 Literature review about monitoring processes with multiple categories

Table 2.2 summarizes the papers about monitoring single stage processes with multiple categories (ordered in columns by first author). Appendix A contains more details about the papers in Table 2.2. Additionally, the method in Marcucci (1985) is longer discussed in Chapter 3.

The literature also contains Bayesian approaches for monitoring a multinomial process. Laviolette (1995) and Shiau *et al.* (2005) consider that the probability parameters of the multinomial model vary according to a prior distribution, typically the Dirichlet distribution. The method in Laviolette (1995) does not provide interpretation of signals. The method in Shiau *et al.* (2005) provides univariate charts using randomized control limits for negatively correlated fractions, but they do not help accurately to interpret signals as shown in Section 3.3. Further, Laviolette (1995) monitors the cumulative posterior distribution of the probability parameters and detects only increases in nonconforming fractions.

Paper	Duran and Albin	Laviolette (Bayes)	Marcucci (Chi-square)	Shiau *et al.* (Bayes)	Spanos and Chen	Tucker *et al.* (Ordinal)
Year of paper	2008	1995	1985	2005	1997	2002
Type of Categories	General	conforming/non conforming	General	conforming/non conforming	Only non conforming	Ordinal
Applications	Transaction Processes & Marcucci's Bricks	Marcucci Bricks	Bricks Quality	Pass and Multiple Fail modes	Semiconductors	Bricks Quality
Process Distribution	Multinomial and equivalent Classification tree	Dirichlet Compound Multinomial aka. Polya-Eggenberger	Multinomial	Dirichlet Compound Multinomial aka. Polya-Eggenberger	Multinomial, Logistic Regression Model	Multinomial
Variable Monitored	Eech K-1 tree fraction	Posterior Cum dist function of K-1 probab	Chi-square statistic	Fractions of count over sample size	Runs for short term about chosen fraction, Pearson Goodness of Fit for long term	Abs value of location parameter of assumed quality underlying distr.
Distribution of Variable Monitored	Binomial	Based on Dirichlet Compound Multinomial	Chi-square	Marginals based on Dirichlet Compound Multinomial	Chi-square for long term, Geometric for short term run rules	N(0,1)
UCL	Yes	Yes	Yes	Yes, including a randomized one	Yes for long term n/a for short term	Yes
LCL	Yes	No	No	Yes, including a randomized one	Not for long term, n/a for short term	Yes
# of Categories in Example(s)	3 and 6	3	3	5	4	3
Sensitivy	ARLs for different values of shifted tree baseline probabilities	No	No	ARLs for different values of shifted prior probabilities	ARLs only for short term SPC, against shifted prob of one fraction	Yes. Better than Marcucci based chart for quality improvement
Interpretation	Yes, based on idependence of every tree fraction	No	No. Individual np-charts proposed for two critical fractions	No	No	No

Table 2.2. Summary of monitoring single stage processes with multiple categories

Tucker *et al.* (2002) propose a monitoring method for ordinal categorical data. The method assumes that the ordinal characteristic has an underlying continuous distribution. If the user knows the underlying distribution, then the sensitivity of the ordinal chart is better than Marcucci's method. The method in Tucker *et al.* is not applicable to nominal

categorical data. No interpretation of signals is provided. Both the *p-tree* and Marcucci methods monitor either nominal or ordinal categorical data.

Finally, Spanos and Chen (1997) introduce process settings that are treated as covariates of a multinomial logistic model for ordinal categorical data. The method monitors the coefficients using a Marcucci method.

2.3 Literature review about monitoring processes in the service industry and healthcare

Monitoring service processes has received limited but increasing attention from practitioners and researchers. Montgomery (2005, p. 185-189) and Devor *et al.* (2007, p. 512-521) include examples about univariate applications of nonmanufacturing processes. From a methodological point of view, Montgomery (2005, p 184) points out that a key element about applying statistical process control in nonmanufacturing applications is to focus initial efforts on developing a valid measurement system. Montgomery proposes to use flowcharts and process charts. Table 2.3 summarizes several references in this field, which are commented below.

Regarding existing research papers, Sulek (2004) reviews the use of statistical quality control in the service industry, emphasizing the modeling of the process flow, and mentioning the potential for monitoring multistage service processes. MacCarthy and Wasusri (2002) review the literature between the years 1989 and 2000 about monitoring methods of nonmanufacturing processes.

Here is a list of articles with applications of monitoring methods to the service industry: Andersson *et al.* (2005) monitor cyclical business processes, with applications to financial decisions as well as comparing firms. Pettersson (2004) monitors customers churn (rapid change of carriers) in the telecommunication industry. Sulek *et al.* (2006) approaches monitoring a service process in a retail operation. Jensen and Markland (1996) monitor quality perception among customers. Heimann (1996) proposes monitoring fractions in a telephone service maintenance process with charts based on

individual observations. Gardiner and Mitra (1994) propose to monitor waiting times in a bank.

Woodall (2006) gives a review of the use of control charts in health-care and in public-health surveillance. He shows that the use of attribute data is often found in health-care applications. Chesher and Burnet (1996) use a 3-sigma *p*-chart to monitor technical performance of clinical laboratories. Hutwagner *et al.* (2005) propose CUSUM methods for monitoring the risk of bioterrorism as well as emergency calls. Benneyan (2006, fig. 6) illustrates the use of a EWMA based *p*-chart to detect increases in the use of prescription drugs. In health-care applications, there is usually 100% inspection, so it is not possible to remove special causes to return the process quickly to in-control. Thus, a control chart might continue to signal after its first signal. The latter issue also applies to service processes, where special causes are searched and investigated but the process is not necessarily stopped. Multivariate monitoring methods remain largely unexplored in the area of health care.

Paper	**Andersson *et al.***	**Devor *et al.***	**Jensen and Markland (quality perception)**	**Montgomery**	**Pettersson**	**Sulek**	**Woodall (Health-Care)**
Year	2005	2007	1996	2005	2004	2004, 2006	2006
Applications	Turn in business cycles in industries	Defects in account payable process	Survey on customers	Plans for aerospace job orders	Churn in telecom-munication industry	Multistage service processes	For example, infection rates or waiting times of various sorts
Process Distribution	Any cyclical time series	Binomial	Assumed MVN	Assumed Normal	Binomial	Varied	Attributes related
Monitoring Method	Likelihood based	u-chart & p-chart	Factor Analysis, T^2 Hotelling chart & PCA	Shewhart	one-side p-chart with variable sample size	Shewhart chart for 1st stage. Cause selecting control chart (about residuals) for 2nd stage.	Varied: CUSUM, EWMA for attributes, Risk Adjusted, etc

Table 2.3. Summary of monitoring processes in the service industry

2.4 Literature review about monitoring multistage processes

First, methods for monitoring multistage and multicategory processes are not found in the literature. Monitoring methods exist for other type of multistage processes, mostly in manufacturing applications. These processes are characterized by a sequence of manufacturing stages in series, in which the output of a stage is an input for the next immediate stage.

Sulek *et al.* (2006) is an early and perhaps unique work about monitoring a multistage process in service. That work approaches monitoring a two-stage service process in a retail operation, using a regression adjustment method relating the two stages. Zou and Tsung (2008) suggest the use of multistage methods to monitor service processes in industries such as telecommunications, banking, and health care.

The following are references about monitoring multistage processes in manufacturing, with correlated stages in series: Niaki and Davoodi (2009), Zou and Tsung (2008), Kaya and Engin (2007), Zantek *et al.* (2006), Jearkpaporn *et al.* (2005), Lee *et al.* (2004), Zhou *et al.* (2003), Heredia-Langner *et al.* (2002), Ding *et al.* (2002a and 2002b), Yao and Chen (1999), Lawless *et al.* (1999), and Agrawal *et al.* (1999). These papers approach multiple correlated stages, and most of them fit a linear model between contiguous stages. Propagation of variation across stages and diagnosis capabilities are key issues in this field. Other monitoring techniques include: neural networks in Niaki and Davoodi (2009), multivariate exponentially weighted moving average control chart (MEWMA) in Zou and Tsung (2008), generalized linear models (GLM) in Jearkpaporn *et al.* (2005), partial least squares (PLS) in Lee *et al.* (2004), and analysis of variance using autoregressive models in Lawless *et al.* (1999) and Agrawal *et*

al. (1999). Heredia-Langner *et al.* (2002) as well as Yao and Chen (1999) focus on finding optimal inspection policies in terms of costs.

As mentioned above, many methods for multistage processes fit linear models between stages, and then set a control chart for residuals (also called-cause selecting methods), e.g.: Zantek *et al.* (2006). Regression adjustment methods were proposed for model-based problems before approaching multistage processes. These works include: Wade and Woodall (1993), Hawkins (1993), Hauck *et al.* (1999), Loredo *et al.* (2002), and Shu *et al.* (2004, 2005).

Many of the above methods propose to monitor independent residuals between contiguous stages. However, there are no methods that propose a decomposition into independent quality characteristics as the methods proposed here in Chapters 3 and 5 for either a single stage process or for a multiple stage process with multiple categories.

3 Monitoring and accurately interpreting processes with multiple categories using a probability tree

Here we present a new method that offers an easier way to interpret an out-of-control signal, the *p-tree* method. Consider a process with more than two, say *K*, categories and the numbers of transactions across categories are multinomial distributed. We construct a probability tree with *K*-1 binary substages that is equivalent to the process with *K* categories. We show that the substages are independent and can be monitored with independent *p*-charts. The independence is based on Johnson *et al.* (1996, p. 68) as well as Kemp and Kemp (1987). The *p-tree* method indicates easily which substages are responsible in case of an out-of-control signal and each *p*-chart represents an estimate of a conditional probability that characterizes a substage.

The principal advantage of the *p-tree* method is its usefulness as a diagnostic tool. The *p-tree* method may monitor both nominal and ordinal categorical data. It has a simple implementation because it decomposes a multivariate problem into independent fractions and *p-charts* (assuming the normal approximation holds for every tree fraction), and it allows any number of categories. The Marcucci and *p-tree* methods seem to have comparable sensitivities.

To illustrate the *p-tree* method, let us apply it to the Marcucci (1985) bricks example of Chapter 1. A probability tree with two binary substages is constructed to represent the three categories. The first substage monitors the fraction of conforming bricks out of the total sample. The second substage monitors the fraction of nonconforming type *A* bricks over the nonconforming bricks. Section 3.2 shows that the fractions monitored in the two substages are independent and each is monitored by a *p*-chart.

The *p-tree* method helps answer our questions about what caused the out-of-control signals for the samples in Table 1.2. In Section 3.2 we show that for sample 1 the fraction conforming brick is consistent with baseline, but among nonconforming, type *A* is underrepresented. For sample 2, the fraction conforming is low compared to baseline and the fraction of type *A* among nonconforming is consistent with baseline.

Consider also the following reduced tax complaint process: a taxpayer with a problem passes through two substages: the first is to consult with a front desk assessor and the second is to actually file a complaint. The management, at the end of the deadline, may classify the notices into three categories: not consulted, consulted but not filed, and filed complaints (assuming that consulting the front desk is required before filing a complaint).

Applying the *p-tree* method to the taxpayers' process, we first monitor the fraction of taxpayers that consult in a month with the stage-one *p*-chart. Then we monitor the fraction of filed complaints over the number of taxpayers that consult with the stage-two *p*-chart. An important assumption is that the taxpayers make decisions independently (non autocorrelated) of one another and that the volume of notices sent in a month does not affect their decisions.

Monitoring customer service processes can be somewhat different from monitoring manufacturing processes. Customer transactions are usually continuously scanned through an IT system, in contrast to manufacturing processes where sampling is often used. In customer transactions the monitoring method might continue to provide signals after its first signal because it may not be possible to remove special causes to return the process quickly to in-control. For example, if the *p-tree* method shows that the fraction of complaints over consults in the tax assessment is abnormally large, then management

may introduce corrective actions like better regulations, improved instructions, and changes in the IT system – but such changes may take some time.

We present simulation studies to demonstrate that the *p-tree* method is a helpful tool in two ways. First, it signals an out-of-control condition in a comparable amount of time to the Marcucci method. Second, it gives correct interpretation information on which category is responsible for the signal, something that the Marcucci and other existing methods method cannot do.

Table 1.2 in Chapter 1 shows also that a single stage with three categories can be seen as a multivariate process. In fact, a monitoring system would need to record at least observations of two counts and the sample size. Indeed, the monitoring system would need to track and record n_1 and either n_2 or n_3 (assuming a constant sample size N). For instance, if n_2 is tracked, n_3 can be deduced as $N-n_1-n_2$. So, Marcucci's bricks data can be seen as a multivariate process with three correlated count variables.

The rest of this Chapter is organized as follows: Section 3.1 gives the equivalence between a multinomial process and a probability tree, and explains how a tree is built and characterized. Section 3.2 describes the *p-tree* control chart that it is based on the independence of the tree fractions of equation (3-3). Section 3.3 shows simulation results to illustrate the diagnosis capabilities of the *p-tree* chart and compares its average run length (ARL) performance with Marcucci's method and with the Bayesian method of Shiau *et al.* (2005). Section 3.4 ends with concluding remarks about the *p-tree* chart.

3.1 Equivalence between multinomial process and probability tree

Consider a multinomial with three categories (trinomial) and a sample of size N transactions. Baseline probabilities in category i are equal to p_i, i= 1,2,3, and the numbers in each category are n_i, i = 1, 2, 3. Of course, $p_1 + p_2 + p_3 = 1$ and $n_1 + n_2 + n_3 = N$. This process is depicted in Figure 3.1a. Equivalently, Figure 3.1b depicts this trinomial process as an equivalent probability tree with two substages. In substage 1, f_1 is the baseline probability that transactions are in category 1 resulting in realized number n_1, and 1-f_1 is the probability that transactions are not in category 1, with realized number N-n_1. In substage 2, f_2 is the baseline conditional probability that transactions are in category 2 given that they are not in category 1, with realized number n_2. The conditional probability that transactions are not in category 2 (therefore in category 3) given that they are not in category 1 is 1-f_2, with realized number n_3. Only two conditional probability parameters completely characterize the process, f_1 and f_2.

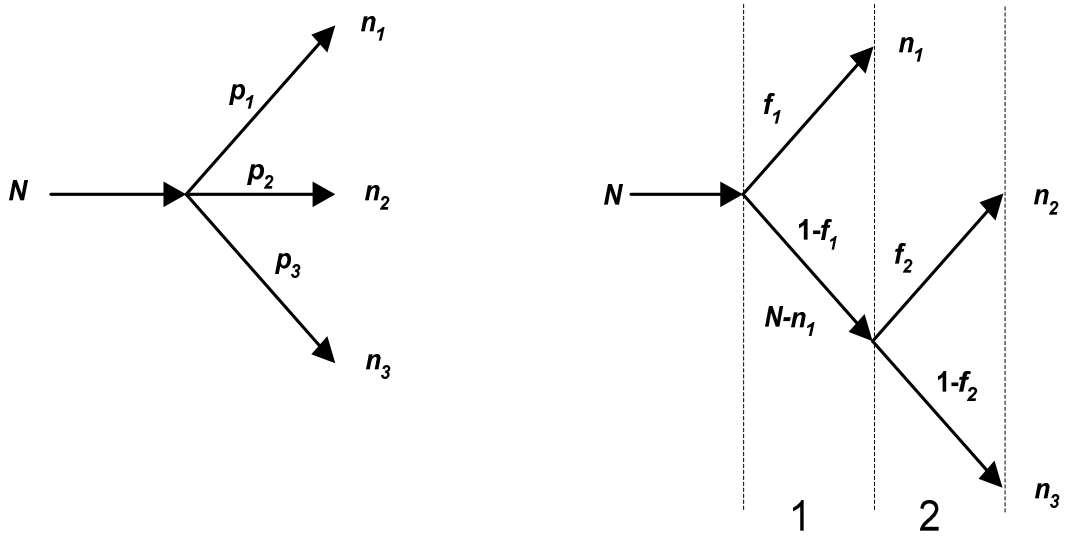

Figure 3.1. (a) A trinomial process and (b) Equivalent probability tree with two substages

Based on the probability multiplicative rule, the p_i may be expressed as a function of the f_i: $p_1 = f_1$, $p_2 = (1-f_1)\cdot f_2$, $p_3 = (1-f_1)\cdot(1-f_2)$. It follows that $f_2 = p_2/(1-p_1)$.

This probability tree representation allows the user to order the categories according to their monitoring importance. In the Marcucci bricks example it is logical to order the substages such that substage 1 discriminates between conforming and nonconforming bricks, with f_1 equal to the probability of conforming bricks, and substage 2 discriminates between nonconforming Type *A* and *B* bricks with f_2 the conditional probability of nonconforming Type *A* given nonconforming. In case categories are of equal or unknown importance, such as transactions type *A*, *B*, or *C*, one can choose an order by default, for example put categories in decreasing order of their in-control probabilities p_i.

The probability tree can also be built as a classification tree: according to Duda *et al.* (2004, Chapter 8), any classification problem can be modeled through a sequence of binary questions that can be answered “yes” or “no”. With *K* categories each transaction

in the sample goes through a sequence of up to *K*-1 questions. The first question is: Classified in category 1? If the response is “yes”, an associated count variable is updated. If the response is “no”, the next question is: Classified in category 2? This procedure stops when a question about category *i* is answered “yes” or when the last question *K*-1 is answered “no” (item is classified in category *K*).

This probability tree (or classification tree) method allows the user to order the categories accordingly to their monitoring importance. In case of equal or unknown importance, choose an order by default, like categories in decreasing order of their in-control probabilities p_i.

The proposed method shows that a multinomial process with *K* categories can be monitored by a *p-tree* method that consists of *K*-1 independent *p*-control charts as shown in the next Section. The notation for the multinomial process is the following,

p_i = baseline probability that item is in category i , $i=1,2,...,K$

n_i = number of items in category i in a sample

N = sample size

The following relations and properties hold:

$$\sum_{j=1}^{K} n_j = N \qquad \text{and} \qquad \sum_{i=1}^{K} p_j = 1$$

The probability mass function of the multinomial distribution is:

$$\Pr(n_1 = x_1, n_2 = x_2, ..., n_K = x_K) = \begin{cases} \dfrac{N! \prod_{i=1}^{K} p_i^{x_i}}{\prod_{i=1}^{K} x_i!} & \text{for } \sum_{i=1}^{K} x_i = N \\ 0 & \text{otherwise} \end{cases}$$

The expectation, variance, and covariance functions are:

$$E[n_i]=Np_i$$

$$Var[n_i]= Np_i(1- p_i)$$

$$Cov[n_i , n_j]= -Np_i p_j < 0 \text{ for } i\neq j$$

For the probability tree the following notation is used: f_i= baseline conditional probability that item is in category i given that is not in category 1,…,i-1 for i=2,…,K-1. The probability mass functions for each substage are given by the following binomial distributions:

$$\Pr(n_1 = x_1) = \binom{N}{x_1} f_1^{\,x_1} (1- f_1)^{N-x_1}$$

$$\Pr(n_2 = x_2 / n_1 = x_1) = \binom{N - x_1}{x_2} f_2^{\,x_2} (1- f_2)^{N-x_1-x_2}$$

The relationships between the multinomial and the probability tree parameters are based on the total probability rule for multiple events as shown in Montgomery and Runger (2002, p. 44-45):

$$p_1 = f_1 \qquad \text{and} \qquad p_i = [\prod_{j=1}^{i-1} (1- f_j)] \cdot f_i \qquad \text{for } i=2,\ldots,K.$$

It follows that,

$$f_i = \frac{p_i}{1 - \sum_{j=1}^{i-1} p_j} \qquad i=2,\ldots,K \qquad (3\text{-}1)$$

Eqn. (3-1) shows that a shift in a probability p_j produces shifts in every tree probability f_i for $j\leq i\leq K$-1. Similarly, a shift in a tree probability f_i may have been produced by a shift on any probability p_j for $j\leq i \leq K$-1. The last probability f_K always

equals one, meaning that if the item is not in categories 1,2,…,*K*-1, it must be in category *K*.

3.2 The *p-tree* method

The *p-tree* method monitors any multinomial process with K categories using K-1 control charts that monitor shifts in the f_i. We show in this Section that these K-1 charts are independent. Assuming that the baseline probabilities p_i are known, then according to eqn. (3-1) the baseline conditional probabilities f_i, i=1,…,K-1 are also known. Using p-charts, the control limits are:

$$\begin{cases} UCL_1 = f_1 + Z_{(1-\alpha^*/2)} \cdot \sqrt{\dfrac{f_1(1-f_1)}{N}} \\ CL_1 = f_1 \\ LCL_1 = f_1 - Z_{(1-\alpha^*/2)} \cdot \sqrt{\dfrac{f_1(1-f_1)}{N}} \end{cases} \tag{3-2}$$

$$\begin{cases} UCL_i = f_i + Z_{(1-\alpha^*/2)} \cdot \sqrt{\dfrac{f_i(1-f_i)}{N-\sum_{j=1}^{i-1} n_j}} \\ CL_i = f_i \\ LCL_i = f_i - Z_{(1-\alpha^*/2)} \cdot \sqrt{\dfrac{f_i(1-f_i)}{N-\sum_{j=1}^{i-1} n_j}} \end{cases} \quad \text{for } i=2,\ldots, K\text{-}1.$$

where Z_p comes from standard normal distribution such that the upper tail area is p. The sample statistic for each control chart is

$$\hat{f}_1 = \frac{n_1}{N} \qquad \text{and} \qquad \hat{f}_i = \frac{n_i}{N-\sum_{j=1}^{i-1} n_j} \qquad i=2,\ldots,K\text{-}1 \tag{3-3}$$

At any observation time, the process is in-control if all K-1 p-charts have sample statistics within the control limits. The process is out-of-control if any of the p-charts signal, i.e., have sample statistics outside the control limits.

The control limits in eqn. (3-2) are based on the independent binomial distributions of the realized number n_i in category i given the realized numbers in categories 1,…,i-1. This independence result is shown by Johnson *et al.* (1996, p. 68) as well as Kemp and Kemp (1987) as follows:

$$\begin{cases} n_1 \sim \text{Binomial}(N, p_1), \\ n_{\text{i}} \text{ given } n_1, n_2, ..., n_{\text{i-1}} \sim \text{Binomial}(N - \sum_{j=1}^{i-1} n_j, \ \dfrac{p_i}{1 - \sum_{j=1}^{i-1} p_j}) \quad \text{for } i=2,\ldots,K\text{-}1 \end{cases} \tag{3-4}$$

For instance, n_2 comes from an independent binomial with sample size N-n_1 and probability $\frac{p_2}{1-p_1}$, which according to eqn. (3-1) is equivalent to f_2. Thus, every p-chart monitors the independent n_i in category i given the realized numbers in categories 1,…,i-1, using the sample tree fraction $\hat{f}_i$ of substage i, which equals the ratio of the realized n_i to its sample size in its respective independent binomial in eqn. (3-4). The square-root terms in eqn. (3-2) are the standard deviations of the $\hat{f}i$ conditioned on n_1, n_2,…, n_{i-1}, obtained from eqn. (3-4). It can be shown also that those square-root terms represent an approximation for the unconditional standard deviation of $\hat{f}_i$ when N is large and the probability that any count equals zero is negligible.

The *p-tree* control charts have a total false alarm rate α, which is also called family wise error rate. Because of the independence property, 1-α (the probability that the monitoring method does not signal given in-control) is:

$$1 - \alpha = (1 - \alpha^*)^{K-1} \tag{3-5}$$

Therefore,

$$\alpha^* = 1 - (1-\alpha)^{(1/K-1)} \qquad (3\text{-}6)$$

where α* is the exact individual false alarm rate for every *p*-chart in the *p*-tree method. Eqn. (3-4) is based on the independent tree fractions $\hat{f}_i$ and on Montgomery (2005, eqn. (10-2), p. 489).

The *p-tree* method may be implemented using Minitab or any other software that offers the *p*-chart. For example, Figure 3.2 shows a *p*-chart for $\hat{f}_1$ and Figure 3.3 shows a *p*-chart for $\hat{f}_2$ for simulated finished bricks data as in Table 1.2 with a total false alarm rate α=0.05. The first two samples in Figures 2.2 and 2.3 are taken from Table 1.2 and are the only ones outside the control limits. The *p*-charts make it easy to interpret the out-of-control signals: sample 1 reflects a decrease in type *A* bricks relative to the total nonconforming bricks (as shown in Fig. 2b) while sample 2 reflects a decrease in conforming bricks (as shown in Fig. 2a). Note that the control limits in Figure 3.3 vary since the number N-n_1 is a variable sample size for $\hat{f}_2$, as well as its denominator.

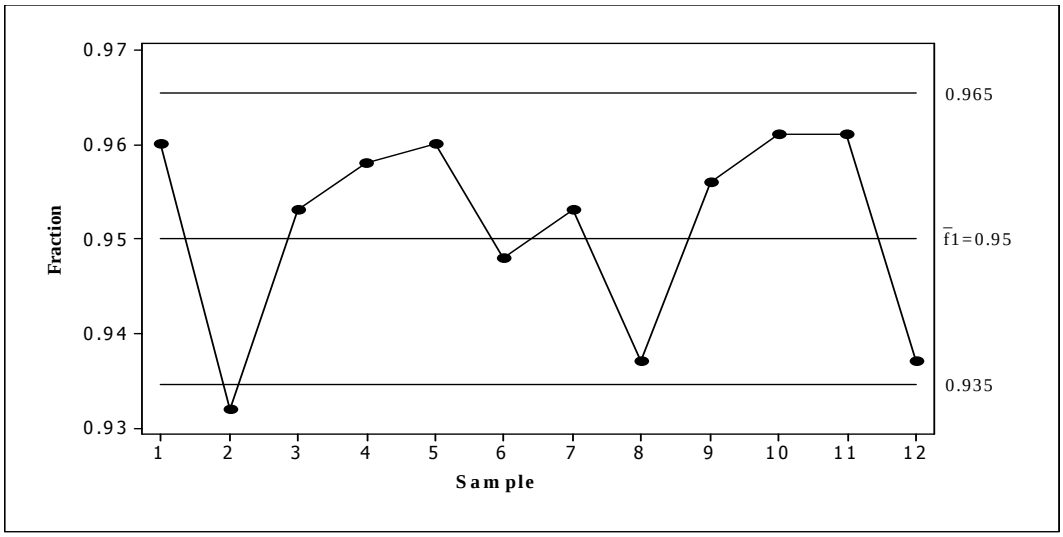

Figure 3.2. *p*-chart for $\hat{f}_1$ (conforming bricks over total)

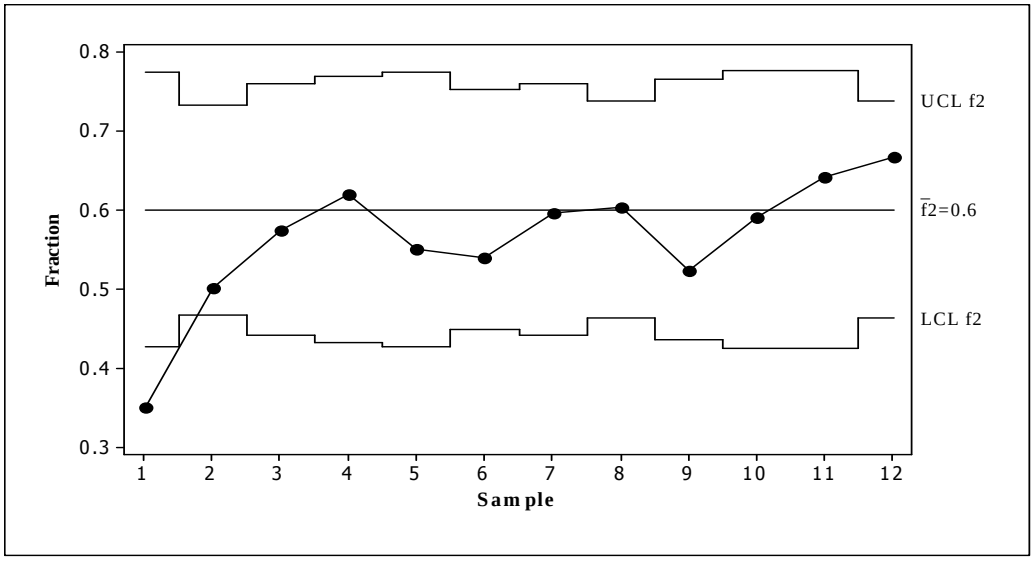

Figure 3.3. *p*-chart for $\hat{f}_2$ (nonconforming Type *A* bricks over all nonconforming bricks)

3.3 Simulation experiments comparing methods

We define the diagnosis accuracy of the *p-tree* method as the fraction of correct signals over the total number of signals given there has been a shift in a tree probability f_i (i=1,2,…,K-1), a concept adapted from Skinner *et al.* (2006). If just the tree probability f_i shifts, the signal is correct only when the *p*-chart for $\hat{f}_i$ signals, and other *p*-charts for $\hat{f}_j$ do not signal, $j \neq i$ and j=1,2,…,K-1.

In this Section we use simulation to assess the diagnosis accuracy of the *p-tree* method under a shift in a tree probability f_i. Note that diagnosis accuracy cannot be computed for the Marcucci method, which does not give information about the specific multinomial probability that has shifted, but only signals whether there has been a shift in any of the probabilities.

Also in this Section, we compare the sensitivity of the *p-tree* and Marcucci methods using ARL, the average number of samples from the time the process shifts until the control chart signals. The number of runs for each simulated condition is determined such that the standard error of every estimated ARL, for both the *p-tree* and Marcucci methods, is less than 0.015 of the estimated ARL. Thus the total number of runs per simulated condition varies between 10,000 and 11,500.

Next, we briefly review the Marcucci control chart. The method uses the Pearson statistic:

$$X^2 = \sum_{i=1}^{K} \frac{(n_i - Np_i)^2}{Np_i} \qquad (3\text{-}7)$$

where n_i is the realized number in category i; N is the sample size; p_i is the known baseline fraction in category i; and Np_i is the expected count in category i. According to

Marcucci (1985), the X^2 statistic in eqn. (3-7) is approximately chi-square distributed with *K*-1 degrees of freedom, assuming the process is in-control, the sample size *N* is greater than 167, and the expected Np_i are not too small. The upper control limit of the Marcucci chart is:

$$\text{UCL} = \chi^2_{(K-1,\alpha)}$$

where α is the false alarm rate and equals the upper tail area of the chi-square distribution with *K*-1 degrees of freedom. There is no lower control limit and the chart only indicates whether the process is in-control or not. It does not indicate whether the probabilities are too high or too low in any particular category.

3.3.1 Diagnosis accuracy and sensitivity for processes with three categories

We conduct a simulation experiment for processes with three categories. The experimental design is summarized in Table 3.1. Two cases of processes are simulated: the first scenario is called the Brick case, which has baseline multinomial probabilities that follow Marcucci's example and the sample size is 1000. The second scenario is called the Customer case, which has probabilities more evenly distributed across the categories and the sample size is 300. Sample size is selected to guarantee a normal approximation to the binomial distribution, the chi-square approximation to the statistic X^2, as well as positive lower control limits, and upper control limits less than one for the individual *p*-charts in the *p-tree* method.

The second factor is the ARL_0, which is the average run length to a false alarm, either 20 or 200, corresponding to choices more common in service and manufacturing, respectively. Since successive observations are independent, the ARL_0 is equal to 1/α.

These levels imply exact α* for the separate *p*-charts equal to 0.0253 and 0.0025 respectively as in eqn. (3-6). Coleman et *al.* (2001) present control charts with ARL_0 of approximately 22 and 370 when monitoring business processes. Marcucci (1985) only uses an ARL_0 of 20 when monitoring samples sizes over 200 bricks.

Scenarios	Levels
Case	Brick p_1=0.95, p_2=0.03, and p_3=0.02 or f_1=0.95 and f_2=0.6, with N=1000 Customer p_1=0.5, p_2=0.25, and p_3=0.25 or f_1=0.5 and f_2=0.5, with N=300
ARL_0	20 and 200
f_1	Brick case: from 0.95 to 0.945, 0.94, 0.935, and 0.93 Customer case: from 0.5 to 0.52, 0.54, 0.56, 0.58, and 0.6
f_2	Brick case: from 0.6 to 0.56, 0.52, 0.48, 0.44, and 0.4 Customer case: from 0.5 to 0.52, 0.54, 0.56, 0.58, and 0.6

Table 3.1. Experimental design for examples with three categories

For each case (Brick and Customer) and ARL_0, we consider shifts on f_1 and f_2. There are between five and six levels each, representing no shift and going up to approximately three standard deviations of each tree fraction. First we simulate the baseline scenario with no shifts. Then we hold f_2 at the baseline value and simulate shifts in f_1 to the levels shown in Table 3.1. Then we hold f_1 at the baseline value and simulate shifts in f_2 to the levels shown in Table 3.1. The baseline values for f_1 and f_2 and the shift sizes are different for the Brick and Customer cases. Also, the shifts are negative for the Brick case and positive for the Customer case.

The independence property of the tree fractions $\hat{f}_1$ and $\hat{f}_2$ is confirmed by the Kendall nonparametric tests of independence (Kendall & Gibbons, 1990, p. 66). The p-values are 0.42 and 0.57 for the Brick and Customer cases respectively, so the null hypotheses of independence are clearly not rejected.

		ARL_0=20			ARL_0=200		
Tree Prob.	Tree Prob. Value	Diagnosis accuracy of *p-tree*	*p-tree* ARL	Marcucci ARL	Diagnosis accuracy of *p-tree*	*p-tree* ARL	Marcucci ARL
	0.95		20.8	20.9		188.3	189.4
	0.945	0.76	10.3	9.0	0.88	46.6	42.4
f_1	0.94	0.90	4.1	3.8	0.97	12.1	12.2
	0.935	0.95	2.1	2.0	0.99	4.3	4.5
	0.93	0.97	1.4	1.4	1.00	2.1	2.3
	0.60		20.7	20.6		188.3	189.3
	0.56	0.69	12.8	12.6	0.74	77.0	76.8
f_2	0.52	0.85	6.0	6.0	0.92	24.2	25.4
	0.48	0.92	3.0	3.1	0.98	8.7	9.6
	0.44	0.95	1.8	1.9	0.99	3.8	4.3
	0.40	0.97	1.3	1.4	0.99	2.1	2.4

Table 3.2. Diagnosis Accuracy (correct signals over the total signals) and ARL performances for Brick case, K=3

		ARL_0=20			ARL_0=200		
Tree Prob.	Tree Prob. Value	Diagnosis accuracy of *p-tree*	*p-tree* ARL	Marcucci ARL	Diagnosis accuracy of *p-tree*	*p-tree* ARL	Marcucci ARL
	0.50		21.1	20.6		214.6	208.1
	0.52	0.71	11.7	11.9	0.80	87.0	94.1
f_1	0.54	0.89	4.7	4.8	0.95	21.1	22.8
	0.56	0.94	2.2	2.3	0.98	6.0	6.6
	0.58	0.97	1.4	1.4	0.99	2.6	2.8
	0.60	0.97	1.1	1.1	1.00	1.5	1.6
	0.50		21.1	20.6		214.6	208.1
	0.52	0.63	14.9	14.6	0.71	123.1	121.6
f_2	0.54	0.81	8.0	7.9	0.90	45.7	45.1
	0.56	0.90	4.1	4.1	0.97	16.1	16.7
	0.58	0.94	2.4	2.5	0.99	6.9	7.3
	0.60	0.96	1.7	1.7	0.99	3.5	3.8

Table 3.3. Diagnosis Accuracy and ARL performances for Customer case, K=3

Tables 3.2 and 3.3 show the diagnosis accuracy results of the *p-tree* method and the ARLs for the *p-tree* and Marcucci methods on the Brick and Customer cases respectively. Diagnosis accuracy of the *p-tree* method and ARLs are measured as an average from all its signaling runs. For both cases, the larger the shift from baseline, the better the diagnosis accuracy. The larger the ARL_0, the better the diagnosis accuracy for the same shift. For example, Table 3.3 shows that if f_1 shifts from 0.5 to 0.52 and ARL_0=20, the *p-tree* method signals correctly for 0.71 of the signals. If f_1 shifts from 0.5 to 0.6 and ARL_0=20, the *p-tree* method give the correct signal for 0.97 of the signals.

If the diagnosis accuracy measure is divided by the ARL, we get the fraction of correct signals over the total of samples. This fraction combines sensitivity with diagnosis accuracy and estimates the probability that the *p-tree* method signals correctly at any sample for a process that is out-of-control. For instances, for the Customer case in

Table 3.3 if f_1 shifts to 0.6 and ARL_0=20, the number of correct signals over the total number of samples in a run is 0.88.

As expected, the *p-tree* method diagnosis accuracy is better if the tree probability that shifts has a lower index, or comes first as a substage, because the substage's sample size is larger. For example, Table 3.3 shows that the diagnosis accuracy for shifts in f_1 are better than for shifts in f_2.

In terms of ARL comparisons, Table 3.2 shows that the Marcucci method is slightly more sensitive than the *p-tree* method when monitoring shifts on f_1 in the Brick case, particularly when ARL_0=20, which is the case developed in Marcucci (1985). Table 3.3 shows that the *p-tree* method is slightly more sensitive than the Marcucci method when monitoring shifts on f_1 in the Customer case. In general, Tables 3.2 and 3.3 show that the differences between the ARLs of both control charts are quite small. The significant contribution of the *p-tree* method is its value as a diagnosis tool.

Figure 3.4 shows a graph comparing the ARL performances when f_1 shifts in the Customer case, for a desired ARL_0=200. Although other graphs are not shown, this is a typical plot. In general, it is visually difficult to distinguish between the methods.

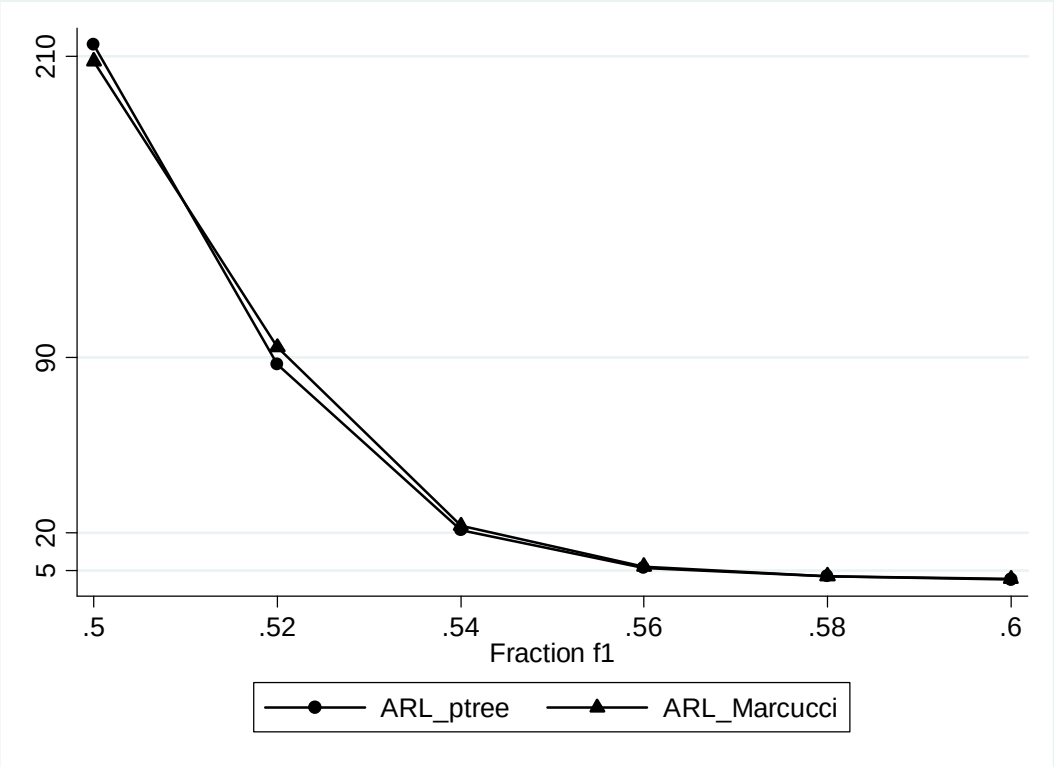

Figure 3.4. ARL comparison for shifts on f_1 in Customer case, K=3, desired ARL_0=200

3.3.2 Diagnosis accuracy and sensitivity for a process with six categories

In this Section all simulations have six categories, and the experimental design is summarized in Table 3.4. Only a Customer case is simulated, with all tree baseline probabilities f_i equal to 0.5, i=1,2,…,5, and sample size is 1000. The ARL_0 equals 20 or 200.

Scenarios	**Levels**
Baseline Probabilities	p_1=0.5, p_2=0.25, p_3=0.125, p_4=0.0625, p_5=p_6=0.03125 or equivalently f_i=0.5, for i=1,…,5, with N=1000
ARL_0	20 and 200
f_1	from 0.5 to 0.51, 0.52,…,0.55
f_2	from 0.5 to 0.51, 0.52,…,0.56
f_3	from 0.5 to 0.52, 0.54,…,0.60
f_4	from 0.5 to 0.52, 0.54,…,0.60
f_5	from 0.5 to 0.53, 0.56,…,0.68

Table 3.4. Experimental design for example with six categories

We consider shifts on f_i for i=1,2,…,5. There are between five and six levels each, representing no shift and going up to approximately three standard deviations of each tree fraction. First we simulate the baseline scenario with no shifts. Then we hold f_j (j=2,…5) at the baseline values and simulate positive shifts in f_1 to the levels shown in Table 3.4. Then we hold f_j (j=1 or j=3,4,5) at the baseline values and simulate positive shifts in f_2 to the levels shown in Table 3.4, so on and so forth.

The exact α* for each p-chart of $\hat{f}i$, i=1, 2,.., 5, obtained from eqn. (3-6), are 0.0102 and 0.001 for ARL_0 of 20 and 200 respectively. The independence property among the tree fractions $\hat{f}i$'s is confirmed by Kendall nonparametric tests of independence among the $\hat{f}i$'s for an in-control data set with 20,000 samples p-values over 0.1 that lead to not rejecting the null hypotheses of independence

Table 3.5 shows that the *p-tree*'s diagnosis accuracy is better if the shift size and/or ARL_0 are larger, and if the tree probability that shifts has a lower index. These results are similar to those in Tables 3.3 and 3.4 for the three-category case. The number of categories does not limit these advantageous features of the *p-tree* method.

		ARL_0=20			ARL_0=200		
Tree Prob.	Tree Prob. Value	Diagnosis accuracy of *p-tree*	*p-tree* ARL	Marcucci ARL	Diagnosis accuracy of *p-tree*	*p-tree* ARL	Marcucci ARL
	0.50		20.0	19.9		204.7	192.3
	0.51	0.40	15.5	16.0	0.49	133.5	144.0
f_1	0.52	0.71	7.6	8.4	0.83	43.4	54.4
	0.53	0.86	3.5	4.1	0.96	12.7	17.4
	0.54	0.92	2.0	2.3	0.98	4.6	6.4
	0.55	0.95	1.4	1.5	0.99	2.3	3.0
	0.50		20.0	19.9		204.7	192.3
	0.51	0.31	17.9	18.8	0.38	156.4	178.7
	0.52	0.53	12.0	12.8	0.72	89.9	103.9
f_2	0.53	0.73	7.0	7.7	0.87	34.0	43.9
	0.54	0.84	4.1	4.6	0.94	14.5	20.9
	0.55	0.90	2.5	3.0	0.97	6.8	10.1
	0.56	0.93	1.8	2.0	0.98	3.7	5.3
	0.50		20.0	19.9		204.7	192.3
	0.52	0.39	15.5	16.7	0.50	127.9	145.8
f_3	0.54	0.70	7.7	8.5	0.85	41.4	52.6
	0.56	0.86	3.6	4.2	0.95	12.1	17.3
	0.58	0.92	2.0	2.3	0.98	4.5	6.5
	0.60	0.95	1.4	1.5	0.99	2.2	3.0
	0.50		20.0	19.9		204.7	192.3
	0.52	0.30	18.0	18.1	0.36	169.9	162.5
f_4	0.54	0.54	12.0	12.4	0.67	90.3	89.8
	0.56	0.73	7.0	7.6	0.86	36.3	42.2
	0.58	0.84	4.0	4.5	0.94	15.1	19.8
	0.60	0.90	2.5	2.9	0.97	7.0	9.6
	0.50		20.0	19.9		204.7	192.3
	0.53	0.31	17.7	17.5	0.37	158.6	140.5
f_5	0.56	0.55	11.5	11.0	0.68	79.1	69.2
	0.59	0.75	6.5	6.4	0.88	31.0	31.0
	0.62	0.85	3.6	3.8	0.95	12.2	14.1
	0.65	0.91	2.2	2.5	0.98	5.6	7.2
	0.68	0.93	1.6	1.8	0.99	3.0	4.0

Table 3.5. Diagnosis Accuracy and ARL performance for Customer case, *K*=6 categories

Table 3.5 also shows for ARL comparisons, that the *p-tree* method is slightly better than the Marcucci method when monitoring shifts on f_1, f_2 or f_3. However, the Marcucci method is slightly more sensitive when monitoring small shifts on f_4 and f_5, particularly

when ARL_0=200. Notice that the smaller the index of the fraction monitored, the larger the substage's sample size in the *p-tree* method, and the better the ARL performance of the *p-tree* method over the Marcucci method.

Figure 3.5 shows a graph comparing the ARL performances when f_4 shifts, for a desired ARL_0=200. This is a typical plot that shows the closeness between the ARLs of both methods.

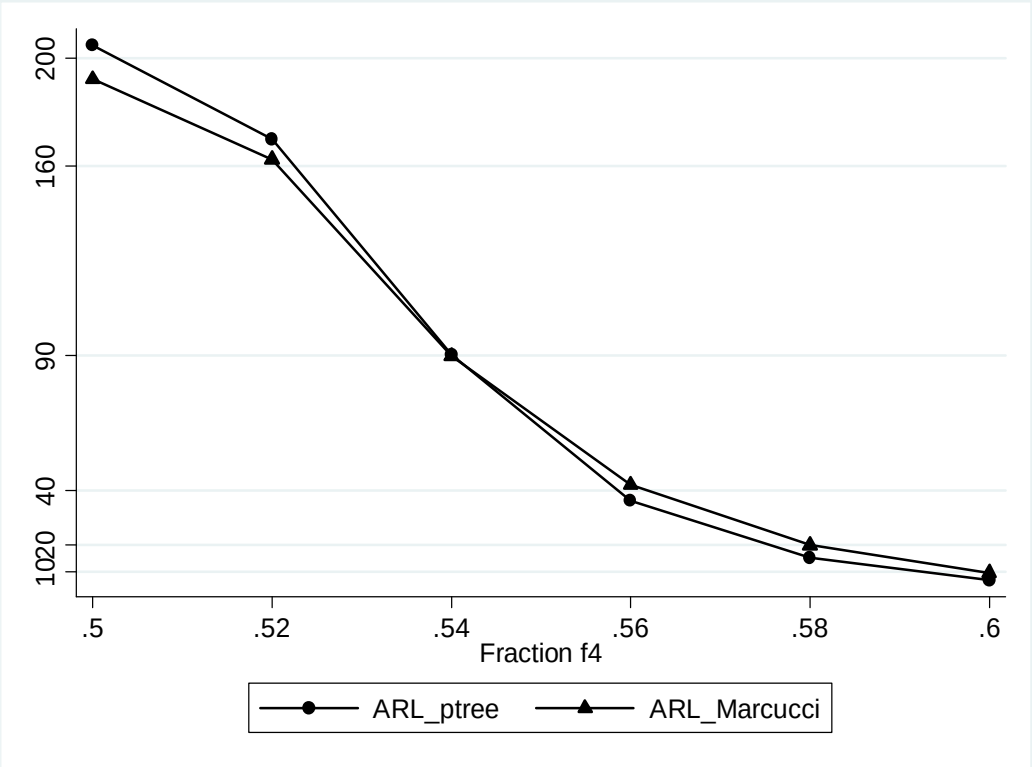

Figure 3.5. ARL comparison for shifts on f_4, K=6, desired ARL_0=200

3.3.3 Diagnosis accuracy and sensitivity for Bayesian method

We show that the Shiau *et al.* (2005) method has little diagnosis accuracy compared to the *p*-tree method. We expected this because the Shiau *et al.* (2005) method was designed to detect an out-of-control process and not to provide interpretations.

Consider the customer case with K=3 as shown on Table 3.1. The process has in-control probabilities p_1=0.5, p_2=p_3=0.25, or equivalently f_1=f_2=0.5, with N=300. We set the control limits using the Shiau *et al.* (2005) procedure (for constant probability parameters). Thus, for an ARL_0=200, the UCL and LCL limits for n_1 are 178 and 122 respectively. If n_1 falls exactly over the control limits, then the sample is out-of-control with probabilities 0.93 and 0.97 respectively. The UCL and LCL limits for n_2 (and n_3) are 100 and 52 respectively. If n_2 (or n_3) falls exactly over UCL or LCL, then the sample is out-of-control with probabilities 0.84 and 0.76 respectively.

Assume that the process goes out-of-control such that p_2 shifts positively, p_3 shifts negatively, and p_1 does not change. In terms of tree probabilities, f_2 shifts positively and f_1 does not change. Diagnosis accuracy is the number of correct signals over the total number of signals. To find the total number of signals in the Shiau *et al.* (2005) method, a signal is counted when any chart for any category signals. A sample has a correct signal when the chart for n_1 does not signal, the chart for n_2 signals, and the chart for n_3 signals. Table 3.6 shows the diagnosis accuracy and ARLs for the simulated in-control situation and for two different shifts. The results of the *p-tree* match those results of Table 3.5.

Process probabilities	Diagnosis accuracy (SE) of *p-tree*	*p-tree* ARL (SE)	Diagnosis accuracy (SE) of *Shiau et al.*	*Shiau et al.* ARL (SE)
In Control: p_1=0.5, p_2=p_3=0.25; or f_1=f_2=0.5		214.6 (2.0)		230.7 (2.4)
p_1=0.5, p_2=0.26, p_3=0.24;or f_1=0.5, f_2=0.52	0.71 (0.0)	123.1 (0.9)	0.01 (0.0)	137.6 (1.0)
p_1=0.5, p_2=0.3, p_3=0.2; or f_1=0.5, f_2=0.6	0.99 (0.0)	3.5 (0.0)	0.15 (0.0)	4.7 (0.0)

Table 3.6. Bayesian Method of Shiau *et al.* (2005) compared with *p*-tree, Customer case, K=3, desired ARL_0=200

As expected, Table 3.6 shows that the Shiau *et al.* (2005) method has a diagnosis accuracy significantly lower than the *p*-tree method. For example, when p_1=0.5, p_2=0.3, p_3=0.2, f_1=0.5, f_2=0.6, the Shiau *et al.* (2005) method has a diagnosis accuracy of 0.15, roughly one correct signal out of seven signals, in contrast with the *p*-tree method which has a diagnosis accuracy of 0.99, (proportion of signals in which only the *p*-chart for $\hat{f}_2$ signals). Thus, the Shiau *et al.* (2005) method should be used only to detect whether the process is in-control or not. It should not be used for interpretations, because its univariate charts are negatively correlated, so its diagnosis accuracy is low. The Shiau *et al.* (2005) method also has lower sensitivity than the *p*-tree method.

The inadequacy of using univariate charts for correlated variables is also noted in Montgomery (2005, p. 487-488 and p. 499) and Lowry *et al.* (1992, p. 52). In contrast, the *p-tree* method uses a decomposition into independent tree fractions, which allows direct and accurate interpretations.

3.4 Concluding remarks about *p-tree* method

Processes in service or manufacturing with multiple categories can be sampled and modeled as multinomial processes. We propose to monitor any multinomial process by decomposition into independent binary substages using a probability tree. If the stages really exist or if the process can be naturally represented in substages as the brick problem, the proposed method allows accurate interpretation of the signals across the substages, as proved theoretically and using simulation.

If a conditional probability related to a tree's substage shifts, the *p-tree* method accurately detects it. As shown on the results, the larger the shift size, or the larger the ARL_0, or the lower the index of the tree's substage that shifts, the better the diagnosis accuracy of the *p-tree*'s signal. By contrast, a signal in the Marcucci method may not be interpreted or measured, because the counts (n_i) across the categories are negatively correlated.

Simulated comparisons of ARL results of the *p-tree* and Marcucci methods with three and six categories show that the *p-tree* method has similar sensitivity (or even slightly better) than the Marcucci method. The *p-tree* method has slightly better sensitivity than the Marcucci's method when monitoring evenly distributed probability trees, i.e., f_i 's close to 0.5, and large shift sizes. The Marcucci's method tends to be more sensitive when monitoring non-evenly distributed tree structures, and small shift sizes. We show also that the method in Shiau *et al.* (2005) has low diagnosis accuracy because of the negative correlation among categories. The decomposition into independent binary stages proposed by the *p-tree* method can be applied to any multinomial process. However, the *p*-chart chart that is used for monitoring each tree fraction requires that

these fractions are normally distributed. In Chapter 4, we propose another univariate method to monitor nonnormal tree fractions, which can be inserted into the *p-tree* method instead of the *p*-chart.

Monitoring fractions in cases in which the *p*-chart may not achieve the desired false alarm rate is addressed in Chapter 4. Other future research issues are: ordering of categories in the tree, and simultaneous shifts on several fractions. Large systems with multiple stages and many categories could be addressed by a decomposition method as developed in Chapter 5. This would be useful in monitoring complex processes involving customers and organizations either in the private or public sectors.

4 Monitoring a fraction with easy and reliable settings of the false alarm rate

Nonconforming fractions have been extensively monitored in manufacturing applications. We also consider monitoring fractions in service applications, which has been covered by several studies. For example, Gans *et al.* (2003. p. 89) show that one measure of the quality of service in call centers is the ratio of the number of customer inquiries that are solved in one contact ("one and done" calls) to the number of the calls that require additional efforts to be solved ("rework" calls). This problem is relevant because processes with multiple categories can be monitored through independent fractions as shown in Chapters 3 and 5.

There are differences between monitoring fractions in service processes and in manufacturing processes. For service, the in-control fraction of interest may vary between 0 and 1, while the in-control fraction in manufacturing tends to be between 0 and 0.1. The process may be continuously monitored using an information system, which either supports the operation of the process in service or collects data from sensors in manufacturing. In service systems, management may monitor based on periodic reports (e.g., monthly), which can be disaggregated into weeks, hours of the day, etc. In manufacturing, reports usually consider small subgroups, which reflect short time operating conditions.

The fraction of interest can be monitored with the familiar *p*-chart based on the binomial distribution, which is simple to run and interpret. However, the *p*-chart may not achieve the desired false alarm rate α as shown in Chapter 1. Notice that achieving α is equivalent to achieving $ARL_{0,}$ which is the in-control average run length until the next false alarm, and equals $1/\alpha$ in case of independent trials.

Here we present a two-sided CUSUM method for monitoring a fraction either in manufacturing or in service processes. The principal advantage of the proposed method is that is easy to use and design from the point of view of the user. We test the proposed method via simulation and find that it achieves a desired α for any N and p_0 such that,

$$E[N]p_0(1-p_0)\geq 3 \qquad (4\text{-}1)$$

where p_0 is between 0.005 and 0.995, and N is constant or Poisson distributed. The proposed method is easy to design because it does not use any of the following techniques: consulting published tables (Tables), combinatorial or enumerative methods (Comb), extensive simulation (Sim), or Markov chain analysis (MC).

Specifically, we propose a CUSUM Arcsine method in which the data is preprocessed using an arcsine normalizing transformation for a binomial distributed variable and then monitored with a two-sided CUSUM method. The parameters of the CUSUM method are set adapting a procedure of Rogerson (2006). The user only needs to determine the control limit as a function of the desired two-sided ARL_0, a formula that can be easily set into a calculator or spreadsheet.

The new two-sided CUSUM Arcsine achieves large desired α such as 1/20, which is typical in service applications, and small α such as 1/200, which may be applied in manufacturing applications. The proposed method is two-sided, i.e., detects either increases or decreases in the in-control p_0.

We show using simulation that other existing easily designed methods do not achieve the desired α. Notice that existing methods that do achieve the desired α are not easy to design, such as: a binomial based EWMA chart in Gan (1990), a binomial based

CUSUM chart in Gan (1993), EWMA *Q*-chart in Quesenberry (1995), and a binomial based (modified) CUSUM chart in Reynolds and Stoumbos (1999, 2000).

The rest of this Chapter is organized as follows: Section 4.2 proposes a new CUSUM Arcsine and a new CUSUM Box-Cox; Section 4.3 shows through simulation experiments that the new CUSUM Arcsine chart consistently achieves a desired α, and has better sensitivity than other easily designed existing methods; Section 4.4 shows an example illustrating an application to a service process.

4.1 New CUSUM Arcsine chart and new CUSUM Box-Cox chart

We propose a method that preprocesses the data with a normalizing transformation, and then monitors it with a CUSUM method that is easily designed. Specifically, we propose the new two-sided CUSUM Arcsine and the new two-sided CUSUM Box-Cox for a fraction. Before, we show how a two-sided CUSUM chart is set into any normalizing transformation (y_t) of the count x_t, where t is the index of the each independent sample, and N_t is the sample size or volume of units at sample t:

1) Preprocess the raw data with a normalization transformation to determine y_t.

2) Determine y_0 and σ_y, which are the in-control mean and the standard deviation of the transformation y_t respectively.

3) Monitor the transformation with a two-sided CUSUM as:

$$C_t^{+} = \max\{0, y_t - (y_0 + K) + C_{t\text{-}1}^{+}\} \tag{4-2}$$

$$C_t^{-} = \max\{0, (y_0 - K) - y_t + C_{t\text{-}1}^{-}\}$$

where the CUSUM is initialized as $C_0^{+} = C_0^{-} = 0$, and K is known as the slack value.

If δ is the shift size (as a multiple of σ_y) considered to be detected quickly, then following recommendations of Woodall and Adams (1993), K is given by:

$$K = \frac{\delta}{2}\sigma_y \tag{4-3}$$

4) The new two-sided CUSUM method signals when either C_t^{+} or C_t^{-} are over the control limit H. We propose to get H as if y_t were normally distributed, which adapting a result from Rogerson (2006) is easily determined as:

$$H \approx ((\frac{\delta^2 ARL_0 + 2}{\delta^2 ARL_0 + 1}) \frac{\ln(\delta^2 ARL_0 + 1)}{\delta} - 1.166)\sigma_y \quad (4\text{-}4)$$

We propose to set δ=1, and eqn. (4-4) reduces only to a function of the two-sided ARL_0 and σ_y:

$$H \approx \left[(\frac{ARL_0 + 2}{ARL_0 + 1}) \ln(ARL_0 + 1) - 1.166 \right] \sigma_y \quad (4\text{-}5)$$

We then use the above procedure to propose a new CUSUM Arcsine method and a new CUSUM Box-Cox method for a fraction. Table 4.1 summarizes the formulation of both methods. The second row shows the transformation, either the Arcsine or the Box-Cox. The Arcsine transformation is specific for binomial distributed data as described by Johnson *et al.* (2005, p. 123) and Chen (1998), and is approximately standard normally distributed. Both transformations may be calculated using just a spreadsheet, but the Arcsine does not require determining a power parameter (L) as does the Box-Cox transformation (described in Appendix D). The third and fourth rows have the in-control mean and standard deviation of each transformation. The following rows give the expressions for each proposed CUSUM.

Symbol	CUSUM Arcsine	CUSUM Box-Cox
Transformed y_t	$y_t = 2\sqrt{N_t}\left[\sin^{-1}\left(\sqrt{\frac{x_t+3/8}{N_t+3/4}}\right)-\sin^{-1}\left(\sqrt{p_0}\right)\right]$	$y_t = \frac{\left(\frac{x_t}{N_t}\right)^L - 1}{L}$
Mean y_0	0	bc_0
Standard deviation σ_y	1	σ_{bc}
Slack K	0.5	$0.5\sigma_{bc}$
Upper side CUSUM C_t^+	$\max\{0, y_t - 0.5 + C_{t-1}^+\}$	$\max\{0, y_t - (bc_0 + 0.5\sigma_{bc}) + C_{t-1}^+\}$
Lower side CUSUM C_t^-	$\max\{0, -y_t - 0.5 + C_{t-1}^-\}$	$\max\{0, (bc_0 + 0.5\sigma_{bc}) - y_t + C_{t-1}^-\}$
Control limit H	$(\frac{1+2\alpha}{1+\alpha})\ln(\frac{1}{\alpha}+1) - 1.166$	$\left[(\frac{1+2\alpha}{1+\alpha})\ln(\frac{1}{\alpha}+1) - 1.166\right]\sigma_{bc}$

Table 4.1. New CUSUM Arcsine method and new CUSUM Box-Cox method for a fraction

4.2 Comparison of easily designed methods for a fraction

We conduct a simulation experiment to study which easily designed methods for monitoring a fraction achieve a desired α. We also compare sensitivity of the selected methods.

The experimental design is summarized in Table 4.2. The first factor corresponds to the evaluated methods, i.e., those methods that have an easy design as shown in Table 2.1. The first method evaluated is the *p*-chart. The following three methods are Shewhart charts on transformations (their implementations are briefly described in Appendix D). The following two methods are the modified *p*-chart of Chen (1998), and the modified *np*-chart of Shore (2000), both described in Appendix E. The following method is the CUSUM *Q* of Quesenberry (1995), which was originally designed using Markov chain analysis. For purposes of experimentation, we try our easy design procedure of Section 3 on that method. The (easy) CUSUM *Q* is obtained similarly to the CUSUM Arcsine. The only change is that the Arcsine transformation shown in the second column of Table 4.1 is replaced by the expression of Q_t as shown in Appendix D. We also evaluate the new CUSUM Box-Cox and the new CUSUM Arcsine.

The second factor is the in-control p_0 with levels: 0.005, 0.01, 0.1, 0.2, 0.3, 0.4, and 0.5. If a process has an in-control probability over 0.5, p_0 could be defined in this experimental design as one minus the in-control probability.

The third factor is the volume *N* which is constant or a zero-truncated Poisson distributed ($N \geq 1$). A constant volume represents the typical sampling of a process and the zero-truncated Poisson distributed volume represents a process that is continuously

monitored. The levels of E[*N*] are obtained such that E[*N*] is the smallest integer such that E[*N*]p_0(1-p_0)≥3.

Factors	**Levels**
Methods evaluated (9 easy)	*p*-chart, Box-Cox chart, *Q*-chart, Arcsine chart, Chen *p*-chart, Shore *np*-chart, CUSUM *Q*, new CUSUM Box-Cox, and new CUSUM Arcsine
p_0 and E[*N*]	0.005 & 604, 0.01 & 304, 0.1 & 34 , 0.2 & 19, 0.3 & 15, 0.4 & 13, and 0.5 & 12
Volume type (*N*)	Constant or Poisson distributed
Desired ARL_0	20 and 200
Shifts in p_0	Cover up to $\pm 3\sigma$
For p_0=0.005	-1.5σ, -1σ, -0.5σ,…, 3σ; σ=0.0029
For p_0=0.01	-1.5σ, -1σ, -0.5σ,…, 3σ; σ=0.0057
For p_0=0.1	-1.5σ, -1σ, -0.5σ,…, 3σ; σ=0.0514
For p_0=0.2	-2σ, -1.5σ, -1σ,…, 3σ; σ=0.0918
For p_0=0.3	-2.5σ, -2σ, -1.5σ,…, 3σ; σ=0.1183
For p_0=0.4	-2.5σ, -2σ, -1.5σ,…, 3σ; σ=0.1359
For p_0=0.5	-3σ, -2.5σ, -2σ,…, 3σ; σ=0.1443

Table 4.2. Experimental design for evaluating easily designed methods for a fraction

The fourth factor is the desired ARL_0, which equals 20 or 200. The fifth factor corresponds to the shifts on p_0 in multiples of σ, which is the standard deviation of the in-control fraction of interest $\frac{x_t}{N_t}$. Thus, p_0 shifts to $p_0 \pm \delta\sigma$, where δ is the shift size in multiples of σ and multiples of 0.5. For the cases in which *N* is constant, $\sigma = \sqrt{\frac{p_0(1-p_0)}{N}}$ (values shown on Table 4.2). For the cases in which *N* is zero-truncated Poisson distributed, it can be shown that $\sigma \approx \sqrt{\frac{p_0(1-p_0)}{(\theta-1)(1-e^{-\theta})}}$, where θ is the

parameter of the Poisson distribution ($\theta > 1$). However, both standard deviations are very close for the levels of E[N] and p_0 used in the experiment.

We consider two performances measures. First, we measure the actual ARL_0. We propose that an acceptable method is such that its actual two-sided ARL_0 is between 18 and 25 or between 180 and 250 for desired ARL_0 of 20 or 200 respectively. These ranges are due to the fact that the parameter estimation of p_0 has a significant effect on the final actual ARL_0, as shown for example in Chakraborti and Human (2006). A method that detects increases but not decreases in p_0 is not considered acceptable.

The second performance is sensitivity, measured as the actual two-sided ARL. We follow a convention found in Lucas *et al* (2006), Ryan and Schwertman (1997), and Quesenberry (1991, 1995) in which a signal represents a sample falling outside the control limits, not on or within these limits. The numbers of runs in the simulations are determined such that the standard error of every estimated ARL is less than 0.02 of its estimated ARL.

4.2.1 Comparison of actual ARL_0

The results show that the new CUSUM Arcsine is the only method that achieves an acceptable actual ARL_0 in all cases for both desired ARL_0 of 20 and 200. In other words, the new CUSUM Arcsine is the only method that in all cases gets an actual two-sided ARL_0 between 18 and 25 or between 180 and 250 for desired ARL_0 of 20 or 200 respectively. The new CUSUM Box-Cox is acceptable for a desired ARL_0 of 20, but not for a desired ARL_0 of 200. These conclusions are supported by the results shown in Tables 4.3 and 4.4.

Table 4.3 summarizes which methods achieve an acceptable ARL_0 for both types of volume, i.e., constant and Poisson distributed. For example, the cell that corresponds to the new CUSUM Box-Cox and to the column E[*N*]=604 and p_0=0.005 shows a 20 because that method gets an acceptable actual ARL_0 for both types of volume. As expected, the *p*-chart, and modified *p*-charts of Chen (1998) and of Shore (2000) do not get acceptable actual ARL_0. The Shewhart charts such as the Arcsine of Chen (1998), the Box-Cox, and *Q* chart of Quesenberry (1991) also do not get acceptable actual ARL_0. These results are consistent with the research found in Ryan and Schwertman (1997), Chen (1998), and Acosta-Mejia (1999).

Table 4.3 shows that the new CUSUM Arcsine achieves an acceptable ARL_0 in all cases for both desired ARL_0 of 20 and 200. The new CUSUM Box-Cox achieves an acceptable ARL_0 in 6 out of 7 cases for desired ARL_0 of 20, but only in 2 out of 7 cases for desired ARL_0 of 200.

		Achieved ARL_0 by cases of E[N] & p_0						
Type	**Method**	**604 & 0.005**	**304 & 0.01**	**34 & 0.1**	**19 & 0.2**	**15 & 0.3**	**13 & 0.4**	**12 & 0.5**
Shewhart Methods	p-chart	-	-	-	-	-	20	-
	Box-Cox chart	20	-	-	-	20	20	20
	Q-chart	-	-	-	-	20	-	-
	Arcsine chart	-	-	-	-	20	20	20
Approx to probability limits	Chen-p-chart	-	-	-	-	20	20	20
	Shore-np- chart	-	-	-	-	-	-	-
CUSUMs	CUSUM-Q	-	-	-	-	-	-	-
	New CUSUM-Box-Cox	20	20	20	20	20	200	20 200
	New CUSUM-Arcsine	**20** **200**	**20** **200**	**20** **200**	**20** **200**	**20** **200**	**20** **200**	**20** **200**

Table 4.3. Acceptable actual ARL_0 by case and method

Table 4.4 shows the average absolute % error of ARL_0 and its standard error across cases of E[N] and p_0 by desired ARL_0, volume type, and method. For each case, i.e., for each combination of E[N] and p_0, the absolute % error of ARL_0 is measured as $\left|\frac{\text{Actual } ARL_0 - \text{Desired } ARL_0}{\text{Desired } ARL_0}\right| \cdot 100$.

Type of Method	Method	For ARL_0=20		For ARL_0=200	
		Vol Constant	Vol Poisson	Vol Constant	Vol Poisson
Shewhart Methods	*p* -chart	28.9 (6.2)	26.0 (4.3)	56.0 (34.8)	64.8 (16.6)
	Box-Cox chart	20.2 (5.4)	21.8 (7.3)	63.4 (18.2)	62.2 (9.8)
	Q -chart	26.8 (9.9)	11.9 (3.6)	44.3 (2.7)	28.3 (8.9)
	Arcsine chart	19.8 (5.1)	8.5 (2.8)	63.5 (22.9)	22.7 (9.1)
Approx to probability limits	Chen-*p* -chart	23.8 (4.3)	8.9 (3.3)	72.9 (52.3)	66.7 (21.4)
	Shore-*np*- chart	48.4 (6.0)	45.8 (3.7)	44.5 (11.8)	44.4 (5.7)
CUSUM on Transformations	CUSUM-*Q*	22.7 (2.1)	24.3 (1.4)	65.4 (0.9)	66.1 (0.3)
	New CUSUM-Box-Cox	9.9 (1.9)	8.6 (3.5)	9.4 (3.6)	17.0 (4.2)
	New CUSUM-Arcsine	**9.6 (2.6)**	**4.7 (1.3)**	**5.5 (1.1)**	**5.8 (1.1)**
	Average across methods	23.3 (4.8)	17.8 (3.5)	47.2 (16.5)	42.0 (8.6)

Table 4.4. Average absolute % error of ARL_0 (SE) by desired ARL_0, volume type, and method

The new CUSUM Arcsine has the lowest average absolute % error of ARL_0 for both desired ARL_0, and for each type of volume. Table 4.4 also shows that the average absolute % error of ARL_0 for a Poisson distributed volume tends to be less than the average absolute % error of ARL_0 for a constant volume (7 out of 9 methods and 5 out of 9 methods, for desired ARL_0 of 20 or 200 respectively). This phenomenon could be due to the fact that a Poisson random variable can take large values. Those large sample sizes tend to improve the methods' approximation in the tails of the binomial distribution. This topic will not be further developed in this paper, but could lead to some future investigation.

In summary, we propose the new CUSUM Arcsine as the best easily designed method that achieves a desired ARL_0 of 20 or 200, for volumes that are constant or Poisson distributed, in which $E[N]p_0(1-p_0)\geq 3$, and p_0 is between 0 and 1.

4.2.2 Comparison of sensitivity

The results show that the new CUSUM Arcsine method has similar sensitivity compared to the new CUSUM Box-Cox method. Only those methods that obtain an acceptable actual ARL_0 (as shown on Table 4.3) are compared. For a desired ARL_0 of 20, both new CUSUM methods have better sensitivity than other easily designed methods when detecting shifts of size up to 1.5σ, and have similar but not better sensitivities than other methods when detecting shifts of size between 2σ and 3σ. For a desired ARL_0 of 200, only the sensitivity of the new CUSUM methods can be compared.

The above summary is supported by the results shown in Tables 4.5 and 4.6, which show the average of ARLs across cases of E[*N*] and p_0 by shift size in multiples of σ, desired ARL_0, and method. Tables 4.5 and 4.6 show results for increases and decreases of p_0 respectively.

The new CUSUM Arcsine and the new CUSUM Box-Cox have similar sensitivities, particularly for shifts of size greater or equal than 1σ. For shifts of size 0.5σ, Table 4.5 shows that the new CUSUM Arcsine has better sensitivity than the new CUSUM Box-Cox for positive shifts of size 0.5σ, either for ARL_0 of 20 or 200. In contrast, Table 4.6 shows that the new CUSUM Box-Cox has better sensitivity than the new CUSUM Arcsine for negative shifts of size 0.5σ, either for ARL_0 of 20 and 200.

	Shifts in Multiples of σ											
	For ARL_0=20						For ARL_0=200					
Method	**0.5σ**	**1σ**	**1.5σ**	**2σ**	**2.5σ**	**3σ**	**0.5σ**	**1σ**	**1.5σ**	**2σ**	**2.5σ**	**3σ**
p-chart	12.5	5.9	3.1	1.9	1.4	1.1	-	-	-	-	-	-
Box-Cox chart	16.7	8.4	4.4	2.6	1.8	1.4	-	-	-	-	-	-
Arcsine chart	14.0	6.4	3.4	2.0	1.4	1.2	-	-	-	-	-	-
Chen-*p*-chart	13.9	6.4	3.3	2.0	1.4	1.2	-	-	-	-	-	-
New CUSUM-Box-Cox	12.1	5.8	3.7	2.7	2.2	1.8	31.5	9.3	5.2	3.6	2.8	2.3
New CUSUM-Arcsine	9.6	4.7	3.0	2.2	1.8	1.5	27.5	9.2	5.4	3.9	3.0	2.6
Average	13.1	6.3	3.5	2.2	1.7	1.4	29.5	9.3	5.3	3.8	2.9	2.5

Table 4.5. Average of ARLs by positive shift sizes and by ARL_0 and method

	Shifts in Multiples of σ											
	For ARL_0=20						For ARL_0=200					
Method	**-0.5σ**	**-1σ**	**-1.5σ**	**-2σ**	**-2.5σ**	**-3σ**	**-0.5σ**	**-1σ**	**-1.5σ**	**-2σ**	**-2.5σ**	**-3σ**
p-chart	20.5	11.9	4.0	2.0	1.2	-	-	-	-	-	-	-
Box-Cox chart	13.9	6.3	2.8	1.7	1.2	1.2	-	-	-	-	-	-
Arcsine chart	14.7	6.5	3.1	1.7	1.2	1.2	-	-	-	-	-	-
Chen-*p*-chart	14.9	6.6	3.1	1.7	1.2	1.2	-	-	-	-	-	-
New CUSUM-Box-Cox	10.9	4.6	2.1	1.7	1.3	1.3	30.1	8.8	4.8	3.3	2.5	2.1
New CUSUM-Arcsine	12.3	4.9	2.4	1.7	1.3	1.3	41.8	9.4	4.3	2.9	2.3	2.0
Average	14.5	6.8	2.9	1.8	1.2	1.2	36.0	9.1	4.6	3.1	2.4	2.1

Table 4.6. Average of ARLs by negative shift sizes and by ARL_0 and method

4.2.3 Example of a process service

Consider a simplified property tax complaint process as described in Duran and Albin (2009a). Notices are sent monthly in batches to taxpayers to communicate changes in the assessment. The assessment process may generate errors that the taxpayer and the property tax system must resolve. A taxpayer with a problem consults first with a front

desk assessor and then decides to file a complaint or not. The taxpayers may have one month to file a complaint. The management, at the end of the deadline, wishes to monitor the fraction of filed complaints over the number of taxpayers that consult.

Assume that the number of taxpayers that consult in a month (N) in one local office is Poisson distributed with E[N]=34. Assume that during the first 10 months the in-control probability (p_0) that a taxpayer that consults files a complaint is 0.1, and p_0 shifts to 0.16 in month 11, which is equivalent to a shift size of 1.15σ (small in multiples of σ, but large as a magnitude). Simulated data is shown in Table 4.7.

Month (t)	Number of consults (N_t)	Number of complaints (x_t)	Fraction (f_t)
1	43	5	0.116
2	33	2	0.061
3	41	3	0.073
4	37	6	0.162
5	35	3	0.086
6	28	3	0.107
7	33	4	0.121
8	31	0	0.000
9	50	9	0.180
10	32	2	0.063
11	27	6	0.222
12	28	7	0.250
13	34	4	0.118
14	34	4	0.118
15	39	9	0.231
16	41	9	0.220
17	33	5	0.152
18	26	2	0.077
19	33	6	0.182
20	33	5	0.152

Table 4.7. Property tax complaint data

Figure 4.1 shows a run chart of the fraction ($f_t=x_t/N_t$). It is difficult to check by eye whether the process is out-of-control or not. Both CUSUM Arcsine charts are constructed

from the data of Table 4.7, with a desired ARL_0 of 20 (the control limit H=2.02 is obtained using the last row of Table 4.1), are shown in Figure 4.2. The chart corresponding to the $CUSUM^+$ (for detecting increases in p_0) shows that the process is indeed out-of-control at months 12 and 16.

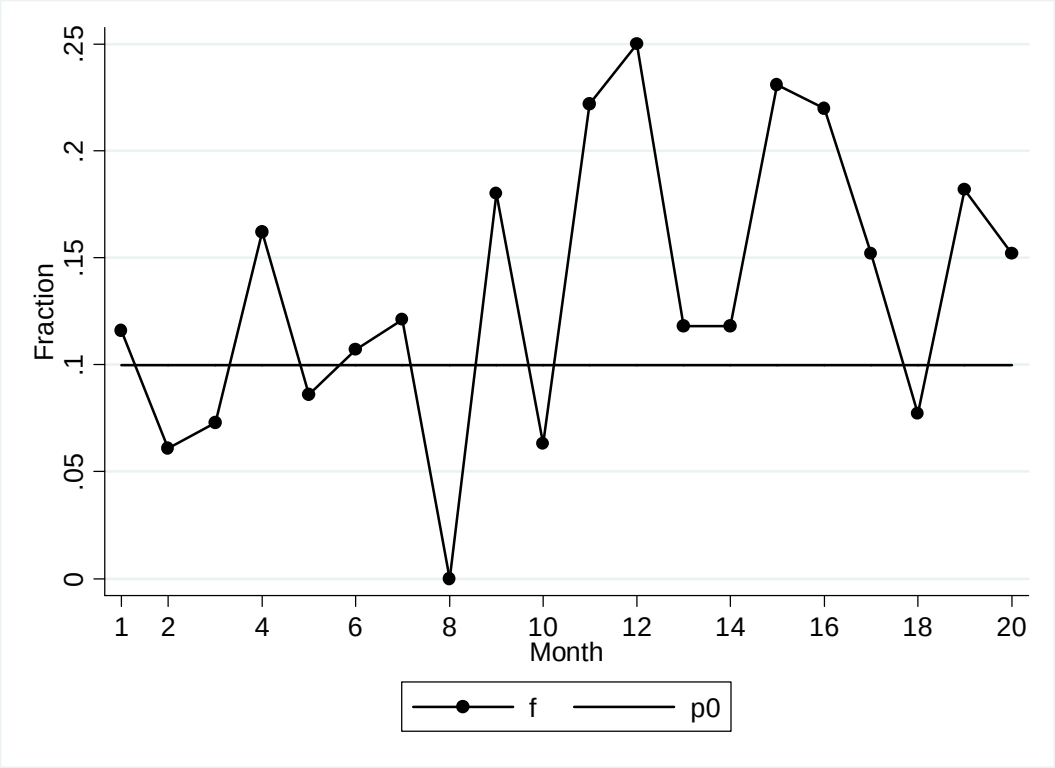

Figure 4.1. Run chart of number of complaints over number of consults

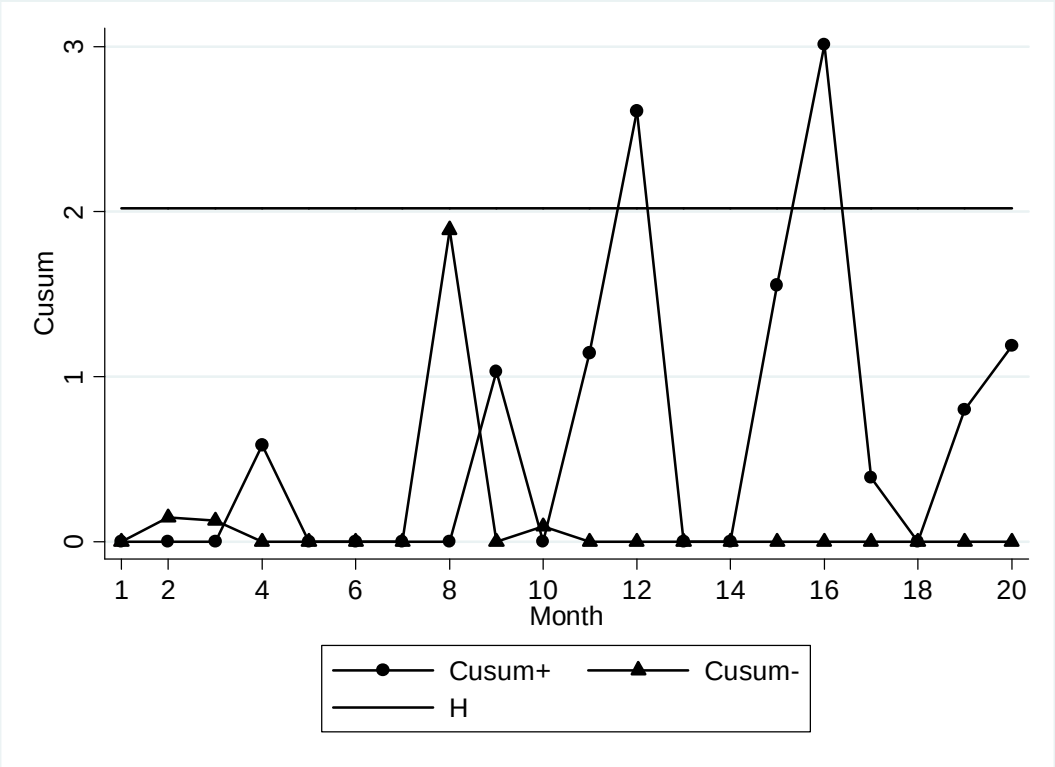

Figure 4.2. Two-sided CUSUM Arcsine charts for fractions of complaints

These charts would be set for every local office of a jurisdiction. For a service process like this, it is not possible to remove special causes to return the process quickly to in-control. However, management may analyze these out-of-control situations and introduce corrective actions like better regulations, improved instructions, and changes in the IT system – but such changes may take some time.

4.3 Concluding remarks about CUSUM for a fraction method

We propose a new CUSUM Arcsine method to monitor a fraction both in service and in manufacturing applications. The proposed method is easy to use and design. The CUSUM Arcsine method achieves small and large α such as $\frac{1}{200}$ and $\frac{1}{20}$, for volumes that are constant and Poisson distributed, in which $E[N]p_0(1-p_0)\geq 3$, and p_0 is between 0 and 1. The proposed method gets better sensitivity than other easily designed methods in all shifts for a desired α of $\frac{1}{200}$. The proposed method gets better sensitivity than other easily designed methods for shifts of size up to 2σ and similar sensitivities in simulations for shifts of size over 2σ for a desired α of $\frac{1}{20}$. The new CUSUM Box-Cox achieves large desired α such as $\frac{1}{20}$, but not always small desired α such as $\frac{1}{200}$, and has similar sensitivity compared to new CUSUM Arcsine.

Future research may explore developing easily designed methods for monitoring processes in which the rule $E[N]p_0(1-p_0)\geq 3$ may not be fulfilled. The proposed CUSUM Arcsine might also be subject to optimization studies of its sensitivity features. Processes in which the volumes are large and the data does not fit the binomial distribution might be investigated too because they can represent complex customer service processes.

5 Monitoring multistage and multicategory processes

We propose to monitor and interpret multistage and multicategory processes (MSMC) using a decomposition methodology. The proposed methodology decomposes stages with multiple categories into binary substages, and also describes the relations among the stages of the process.

The methodology for a single stage and multicategory process proposed in Chapter 3, and in Duran and Albin (2009a), is extended to the multistage and multicategory process. If the multinomial model fits every stage of the process, then a number of independent fractions ─ called tree fractions ─ are used to monitor and provide full interpretations within and across stages of the process. Additionally, the initial volume of customers can be monitored too.

The methodology proposes to monitor every tree fraction corresponding to every binary substage of the MSMC. Each tree fraction is monitored using the CUSUM Arcsine method proposed in Chapter 4, and in Duran and Albin (2009b). The CUSUM Arcsine method has the advantage that is easily designed and achieves a desired false alarm rate when monitoring a fraction, especially those with small sample sizes. The CUSUM Arcsine method also has good sensitivity properties to detect shifts of different sizes.

Similarly to the single stage case, the order of the stages and categories matters in terms of describing the process and in terms of monitoring properties such as sensitivity and achieving a desired false alarm rate. Thus, the user may order the categories according to their monitoring importance within each stage. The user could also reorder the stages and get a new multinomial probability tree as long as the new tree makes sense describing the process.

The rest of this Chapter is organized as follows: Section 5.2 proposes a methodology to monitor multistage and multicategory processes; Section 5.3 develops a case study about a call center; Section 5.4 concludes.

5.1 Methodology to monitor multistage and multicategory processes

Consider a multistage and multicategory process (MSMC) with the following notation and assumptions:

i. Process starts at stage 0 with an initial volume of $N^{(t)}$ customers at sample t.

ii. Process has additional M stages.

iii. Every stage j has a total of K_j categories, $j=0,1,2,\ldots,M$. By definition, $K_0=1$ and $K_j \geq 2$ for $j \geq 1$.

iv. $n^{(t)}_{(i,j)}$ = realized number of customers that fall in category i of stage j at sample t, for $i=1,2,\ldots,K_j$ and $j=0,2,\ldots, M$. By definition, $n^{(t)}_{(1,0)}=N^{(t)}$.

v. Customers are classified in one exclusive category at each stage.

vi. Realized numbers at every stage are multinomial distributed.

vii. Volume of customers does not affect their customers decisions.

viii. Customers move forward through stages. There are no loops in the process.

ix. There is only one path to reach a category as shown on Figure 5.1.

Figure 5.1. One path to reach a category

The algorithm to monitor MSMC is first summarized in words, and then formally explained using matrix representation. The algorithm consists of two procedures: an input procedure and a monitoring procedure. Here is a summary of the input procedure:

1. Visualize the MSMC process using a multinomial probability (e.g., Figure 1.1).

2. Identify splitting processes across MSMC process:

   2.1. There is only one splitting process at stage 1, with a total of K_1 categories, and sample size equal to initial volume $N^{(t)}$.

   2.2. Identify all splitting processes and root categories at stage j, for j=2,…,M: a root category within a stage splits into several offspring categories at the next stage. The realized number in a root category is the sample size for its splitting process in the next stage. The first root category in a MSMC is category 1 (unique) at stage 0 with realized number $N^{(t)}$, which is the sample size at sample t for the splitting process in stage 1.

Here the monitoring procedure is described:

1. Build binary probability tree of MSMC process: each splitting process with more than two categories in step 2 of the input procedure is decomposed into binary substages using the probability tree decomposition method for a single stage process with multiple categories as explained in Chapter 3.
2. Determine the tree fractions: the tree fractions monitor every binary substage obtained in the step 1. If each splitting process can be modeled with a multinomial distribution, then the tree binary substages and the tree fractions are independent.
3. Check the multinomial distribution assumption. This is done testing whether the volume $N^{(t)}$ and the sample tree fractions determined in step 2 are independent and non-correlated random variables. Two different tests of hypothesis can be used: the Kendall nonparametric test of independence in Kendall & Gibbons (1990, p. 66) and/or the test of null pairwise correlation between fractions as shown in

Montgomery and Runger (2002, p. 402). In both cases, not rejecting the null hypothesis is a sign of independence.

4. Monitoring MSMC process:

   4.1. If $N^{(t)}$ and the sample tree fractions are independent: monitor and interpret each tree fraction with a univariate control chart. We recommend the CUSUM Arcsine method of Chapter 4. Additionally, monitor the volume $N^{(t)}$ if needed.

   4.2. If $N^{(t)}$ and the sample tree fractions are not independent: the method proposed here cannot be applied. A further multivariate method needs to be investigated.

It will be show below (eqn. (5-8)) that there is total of K_F-1 tree fractions, where K_F is the number of final categories, i.e., those that are not split in any further categories. Notice that this number of sufficient fractions K_F-1 is independent of the number of stages, and of the number of categories in intermediate stages, and only dependent of the number of final categories.

5.1.1 Methodology to monitor MSMC processes using matrices

The algorithm to monitor MSMC processes summarized above is explained here using matrix representation. This representation is useful for a potential development of software for the algorithm. In the literature, Beygelzimer *et al.* (2005) and Kaplan (1982) approach the representation of trees using matrices, either for diagnosis or for risk analysis. An example about applying matrix representation is presented in Section 5.2.

For the input procedure:

The first step is representing the multinomial probability tree using the following matrices: $\underline{\mathbf{L_j}}$= linking indicator matrix from stage *j*-1 into stage *j*, for *j*=1,2,…,*M*. There are *M* matrices $\underline{\mathbf{L_j}}$, each one having K_{j-1} rows and K_j columns, i.e., a dimension (K_{j-1} x K_j).

The elements of the $\underline{\mathbf{L_j}}$ are given by:

$$\underline{\mathbf{L_j}}\,[l, m] = \begin{cases} 1 & \text{if category } m \text{ in stage } j \text{ has a root in category } l \text{ at stage } j-1 \\ 0 & \text{otherwise} \end{cases} \tag{5-1}$$

, for l=1,2,…,K_{j-1} and m=1,2,…,K_j.

The rows of $\underline{\mathbf{L_j}}$ to be denoted $\underline{\mathbf{L_j}}[l,\cdot]$ are an array of linking indicators to relate root category *l* in stage *j*-1 with its splitting categories in stage *j*. Those elements that take the value 1 are called active cells, with a category in stage *j* having a root category equal to *l* in stage *j*-1. If the elements of a row vector $\underline{\mathbf{L_j}}[l,\cdot]$ are summed up, the result equals the number of categories in which the root category *l* of stage *j*-1 splits into stage *j*. This sum of elements across a vector can be expressed as Manhattan distance (MD) as shown by Duda *et al.* (2001, p. 188) because of the streets and elevators distance analogy in three dimensions. Thus the number of categories into which category *l* of stage *j*-1 splits into stage *j* is:

$$\mathrm{MD}(\underline{\mathbf{L_j}}[l,\cdot]) = \sum_{i=1}^{Kj} \underline{\mathbf{L_j}}[l,i] \tag{5-2}$$

We adopt the convention that any final category *l* in an intermediate stage *j*-1 has a (virtual) splitting into one category in stage *j*, i.e., MD($\underline{\mathbf{L_j}}[l,\cdot]$)=1, and (virtually) into one category in each successive stage.

The second step is recording the realized numbers in every category at sample *t* using the following matrices: $\underline{\mathbf{S_j}}^{(t)}$ = realized numbers transition matrix from stage *j*-1 into stage

j at sample t, for j=1,2,…,M. Matrix $\underline{\mathbf{S}}_{\mathbf{j}}^{(t)}$ has dimension (K_{j-1} x K_j). There are M matrices $\underline{\mathbf{S}}_{\mathbf{j}}^{(t)}$ to represent the MSMC at sample t.

The elements of $\underline{\mathbf{S}}_{\mathbf{j}}^{(t)}$ are $\underline{\mathbf{S}}_{\mathbf{j}}^{(t)}[l,m]$ = realized number of customers that split from category l in stage j-1 into category m in stage j at sample t, for l=1,2,…,K_{j-1} and m=1,2,…,K_j. These elements link the root category l in stage j-1 with the splitting category m in stage j. The matrices are populated using:

$$\underline{\mathbf{S}}_{\mathbf{j}}^{(t)}[l,m] = \begin{cases} 0 & \text{if } \underline{\mathbf{L}}_{\mathbf{j}}[l,m] = 0 \\ n^{(t)}{}_{(m,\,j)} & \text{if } \underline{\mathbf{L}}_{\mathbf{j}}[l,m] = 1 \end{cases} \tag{5-3}$$

Because of the properties of MSMC processes enunciated at the beginning of Section 5.1, the matrices $\underline{\mathbf{L}}_{\mathbf{j}}$ and $\underline{\mathbf{S}}_{\mathbf{j}}^{(t)}$ have a special structure. For example, the columns of $\underline{\mathbf{L}}_{\mathbf{j}}$ and $\underline{\mathbf{S}}_{\mathbf{j}}^{(t)}$ always are full of zeroes but in one row that identifies the root category l in stage j-1 ("unique path" property).

The rows of $\underline{\mathbf{S}}_{\mathbf{j}}^{(t)}$ to be denoted $\underline{\mathbf{S}}_{\mathbf{j}}^{(t)}[l,\cdot]$ are an array of transition realized numbers at sample t from category l of stage j-1 into realized numbers across categories of stage j. If the elements of a row vector $\underline{\mathbf{S}}_{\mathbf{j}}^{(t)}[l,\cdot]$ are summed, the result equals the realized number at sample t in the root category l within the previous stage j-1. Using the Manhattan distance, this is expressed as:

$$\mathrm{MD}(\underline{\mathbf{S}}_{\mathbf{j}}^{(t)}[l,\cdot]) = \sum_{i=1}^{Kj} n^{(t)}{}_{(i,\,j)} \;=\; n^{(t)}{}_{(l,\,j-1)} \tag{5-4}$$

Some elements of the matrices $\underline{\mathbf{S}}_{\mathbf{j}}^{(t)}$ change and others do not change. The elements $\underline{\mathbf{S}}_{\mathbf{j}}^{(t)}[l,m]$ that change or are active have a corresponding element $\underline{\mathbf{L}}_{\mathbf{j}}[l,m]$=1. Also if $\mathrm{MD}(\underline{\mathbf{L}}_{\mathbf{j}}[l,\cdot])$=1, i.e., if category l in stage j-1 is a final category, then its virtual splitting

into one category in the next stage j implies that the row vector $\underline{\mathbf{S_j}}^{(t)}[l,\cdot]$ has all elements equals zero except in column m, such that $n^{(t)}{}_{(l,\,j-1)} = n^{(t)}{}_{(m,\,j)}$.

For the monitoring procedure:

The first step is determining the tree fractions in their matrix form:

Define:

$\underline{\hat{\mathbf{F}}_\mathbf{j}}^{(t)}$ = tree fraction matrix from stage j-1 into stage j at sample t, for j=1,2,…,M. The dimension of each matrix is (K_{j-1} x K_j-1). There are M matrices $\underline{\hat{\mathbf{F}}_\mathbf{j}}^{(t)}$ to be monitored at sample t. The elements of these matrices are obtained using the probability tree method of Chapter 3 for every splitting within a stage across the MSMC process. Thus, these elements are given by:

$$\underline{\hat{\mathbf{F}}_\mathbf{j}}^{(t)}[l,\,m] = \begin{cases} \dfrac{\underline{\mathbf{S_j}}^{(t)}[l,m]}{\mathrm{MD}(\underline{\mathbf{S_j}}^{(t)}[l,\cdot])} & \text{if } m = 1 + \mathrm{UC}_j(l-1) \\ \dfrac{\underline{\mathbf{S_j}}[l,m]}{\mathrm{MD}(\underline{\mathbf{S_j}}^{(t)}[l,\cdot]) - \sum_{i=1}^{m-1} \underline{\mathbf{S_j}}^{(t)}[l,i]} & \text{if } 2 + \mathrm{UC}_j(l-1) \le m \le \mathrm{UC}_j(l) \\ 0 & \text{if } \underline{\mathbf{L_j}}[l,m] = 0 \end{cases} \qquad (5\text{-}5)$$

where $\mathrm{UC}_\mathbf{j}(i)$ = last category of stage j in which category i of stage j-1 splits into. The following relation holds:

$$\mathrm{UC}_j(i) = \sum_{k=1}^{i-1} \mathrm{MD}(\underline{\mathbf{L_j}}[k,\cdot])$$

The second step is determining which elements of these matrices can change or are active elements. Notice that any inactive element in $\underline{\mathbf{S_j}}^{(t)}$ has a corresponding inactive element in $\underline{\hat{\mathbf{F}}_\mathbf{j}}^{(t)}$ and take the value zero at every sample t. Additionally, some elements in $\underline{\hat{\mathbf{F}}_\mathbf{j}}^{(t)}$ always take the value one because they monitor the last category of a splitting

process – as explained in Section 3.1. Thus, define $\underline{\mathbf{A_j}}$=Active tree fractions matrix in stage *j*. The elements of $\underline{\mathbf{A_j}}$ identify active and inactive tree fractions. Row vector $\underline{\mathbf{A_j}}[l,\cdot]$ points out to MD($\underline{\mathbf{L_j}}[l,\cdot]$)-1 active tree fractions in $\underline{\hat{\mathbf{F}}_{\mathbf{j}}}^{(t)}$ (value of eqn. (5-2) minus one). The elements of $\underline{\mathbf{A_j}}$ are given by:

$$\underline{\mathbf{A_j}}[l,m] = \begin{cases} 1 \text{ or active} & \text{if } \underline{\mathbf{L_j}}[l,m] = 1 \text{ and } m \neq \sum_{i=1}^{l} \mathrm{MD}(\underline{\mathbf{L_j}}[i,\cdot]) \\ 0 \text{ or inactive} & \text{if otherwise} \end{cases} \tag{5-6}$$

Here we show that there are K_F-1 active tree fractions. According to eqn. (5-2), category *l* in stage *j*-1 splits into MD($\underline{\mathbf{L_j}}[l,\cdot]$) categories in stage *j*. Thus, using the probability tree method, this splitting process can be monitored with MD($\underline{\mathbf{L_j}}[l,\cdot]$) -1 tree fractions. The K_{j-1} splitting processes in stage *j* can be monitored with $\sum_{i=1}^{K_{j-1}} (\mathrm{MD}(\mathrm{Lj}[\mathrm{l},\cdot]) - 1)$ tree fractions, which gives:

$$\text{Number of active tree fractions in } \underline{\mathbf{A_j}} = K_j - K_{j-1} \tag{5-7}$$

Thus,

$$\text{Total number of tree fractions} = \sum_{j=1}^{M} (K_j - K_{j-1}) = K_M - 1 = K_F - 1 \tag{5-8}$$

Notice that $K_M = K_F$ because stage *M* includes those (virtual) categories that at intermediate stage split (virtually) into one category. Every tree fraction monitors an independent binary stage within a splitting process. In other words, according to the results in Johnson *et al.* (1996, p. 68) as well as Kemp and Kemp (1987) all binary

substages of a binary probability tree represents a sequence of independent binomial distributions.

It can be shown that the number of active tree fractions K_F-1 is usually significantly lower than the total number of elements in the matrices $\underline{\hat{\mathbf{F}}_j}^{(t)}$, which is given by $\sum_{j=1}^{M} K_{j-1} \cdot (K_j - 1)$. This fact opens the opportunity to investigate other structures of data in order to minimize the number or proportion of inactive elements in the monitoring procedure.

The third step is checking the multinomial assumption. If each splitting process can be modeled with a multinomial distribution, then the tree binary substages and the tree fractions are independent. Thus, the multinomial assumption is checked via testing whether the volume $N^{(t)}$ and the active tree fractions at sample t determined in the previous step for an in-control process are independent among themselves. We recommend to use the Kendall nonparametric test of independence in Kendall & Gibbons (1990, p. 66) and/or the test of null pairwise correlation between fractions as shown in Montgomery and Runger (2002, p. 402). In both cases, not rejecting the null hypothesis is a sign of independence.

If $N^{(t)}$ and the active tree fractions are not independent among themselves, then the MSMC process can not be modeled with the multinomial distribution, and the process can not be monitored using independent control charts as proposed here. A further multivariate method is needed.

The fourth step is monitoring the process. If the independence property is confirmed in the previous step, then univariate control charts can be used to monitor and interpret

the active tree fraction cells as well as the initial volume $N^{(t)}$ if needed. Similarly to the *p*-tree method, if the control chart for an active $\underline{\hat{\mathbf{F}}_{\mathbf{j}}}^{(t)}$ [*l*, *m*] signals, then category *m* at the stage *j* is causing a disturbance at sample *t*.

The method can be used either in Phase I or Phase II. Phase I provides an exploratory and retrospective analysis to answer whether the process is stable and in-control, and to find parameters in order to build control charts for monitoring in Phase II. Once the process is stable, we recommend monitoring each fraction using the CUSUM Arcsine proposed in Chapter 4. The CUSUM Arcsine method has the advantage that is easily designed and achieves a desired false alarm rate when monitoring a fraction, especially those with small sample sizes. The CUSUM Arcsine also has a better sensitivity than the *p*-chart and other easily designed existing methods when monitoring small shift sizes in binomial distributed data. Additionally, it can be shown that the CUSUM Arcsine has an acceptable sensitivity compared with the *p*-chart when monitoring large shift sizes, i.e., shifts with size over 2-sigma.

Suppose a total desired false alarm rate α and the user sorts the active tree fraction from 1,2,…,K_F-1. Because of the independence property, 1-α (the probability that the monitoring method does not signal given in-control) is:

$$1-\alpha = \prod_{i=1}^{K_F-1}(1-\alpha_i) \qquad (5\text{-}9)$$

where α_i is the individual false alarm for the control chart that monitors the active tree fraction *i*. If all individual control charts has the same false alarm rate $\alpha_i=\alpha^*$, this is given by:

$$\alpha^* = 1-(1-\alpha)^{(1/K_F-1)}. \qquad (5\text{-}10)$$

Eqn. (5-10) is based on the independent tree fractions and on Montgomery (2005, eqn. (10-2), p. 489). The CUSUM Arcsine method for each active tree fraction is implemented as follows:

i. Use the numerator of eqn. (5-5) for the realized number in the binary substage of interest at sample *t*.

ii. Use the denominator of eqn. (5-5) for the sample size at sample *t* in the binary substage of interest at sample *t*.

iii. Use the in-control value of the tree fraction of interest, obtained though eqn. (F-1) of Appendix F.

iv. Use the individual false alarm rate α* obtained through eqn. (5-10).

v. Construct the two-sided CUSUM control charts as shown in Table 4.1, and monitor each active tree fraction.

vi. If a CUSUM chart for tree fraction $\underline{\hat{\mathbf{F}}_{\mathbf{j}}}^{(t)}[l, m]$ signals, then category *m* at stage *j* is causing a disturbance at sample *t*.

5.2 Case study: a call center process

Consider a call center of a commercial bank described in Mandelbaum *et al.* (2001). This example is used here to show the application of the algorithm to monitor a MSMC process in Phase II. In the next sections, the algorithm to monitor the call center is presented, including a matrix representation; and then a simulated call center is monitored.

Figure 5.2 shows a business process diagram for the call center (decimals represent transition fractions in 1999). Three stages are proposed for this process: In stage one, the process starts with callers that seek to speak to a bank representative. Among the callers, 5% abandon (hang up) immediately, 35% speak to a representative without waiting at all - meaning that a representative is available - and the other 60% of customers are put in a waiting queue until a representative is available. In stage two, among those customers waiting in queue, 25% abandon and 75% finally do speak to a representative. In stage three, among those customers that abandon while waiting in queue, 20% are called back as ordered by a bank's supervisor, who makes that decision according to the customer's business priority.

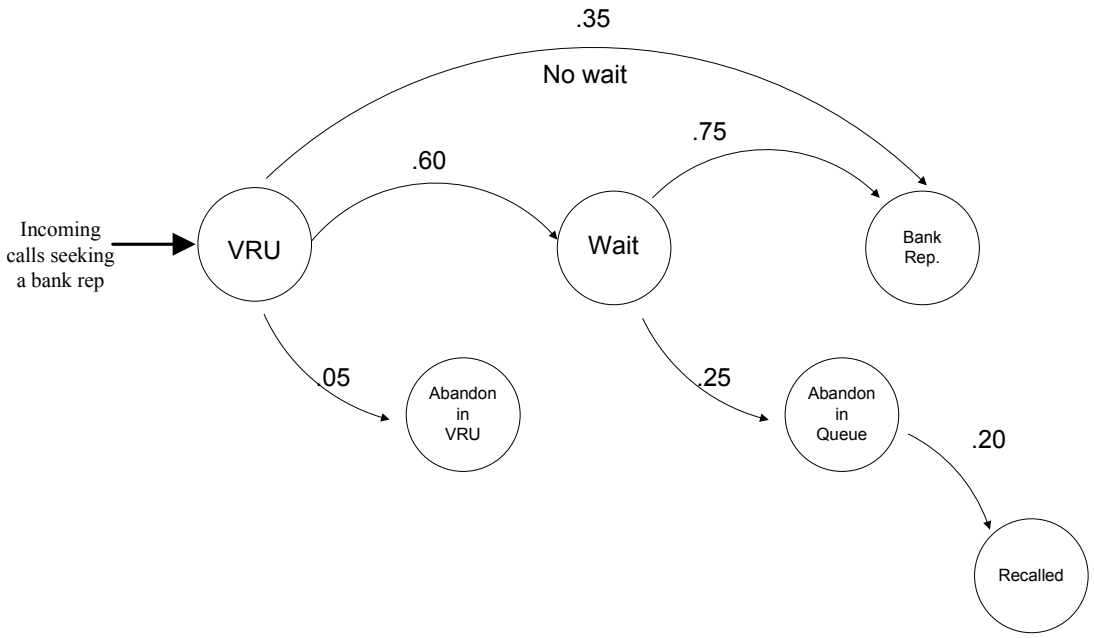

Figure 5.2. Call center business process diagram

A monitoring system would monitor any changes in the transition probabilities that characterize the process. For example, Gans *et al.* (2003, p. 89) mentions the need to measure the number of calls that abandon while waiting for attention as well as obtaining other quality measures. Mandelbaum *et al.* (2001, p. 70) actually envisions a systematic analysis of inter-relations between blocks or components of the system, and their effects on performances measures.

5.2.1 Algorithm applied to call center

The first step for the input procedure is obtaining a multinomial probability tree that is helpful for visualization. Figure 5.3 and Table 5.1 show the multinomial probability tree for the call center with M=3 stages and its notation respectively. In Figure 5.3, the decimal numbers indicate average yearly fractions, and the integers on the bottom indicate stage number.

Notice that any final category at an intermediate stage is represented with (virtual) successive splitting into one category. For example, category 1 at stage 1 (virtually) splits into one category at stage 2 and one category at stage 3. This is a requirement of the matrix representation. Stage 1 has K_1 =3 categories; stage 2 has K_2 =4 categories, and stage 3 has K_3 =5 categories. Thus, K_F=K_3=5 final categories.

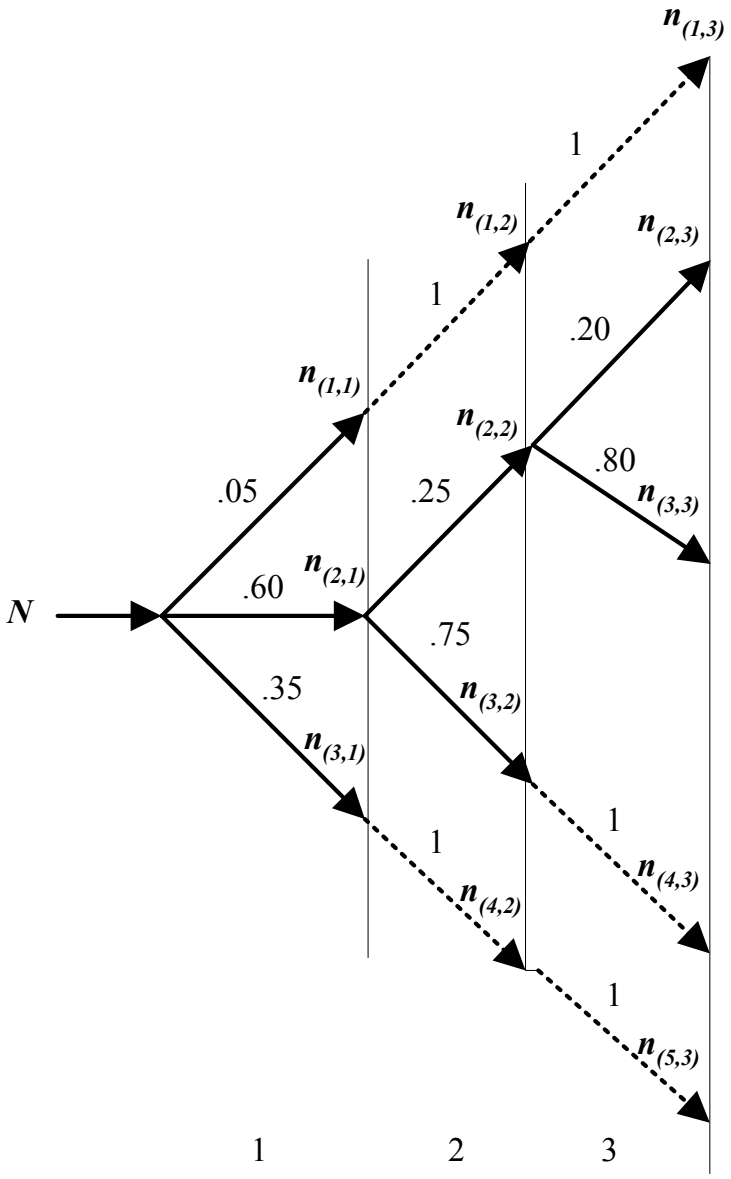

Figure 5.3. Multinomial probability tree for call center

Stage	Symbol	Number of customers at sample *t* that,
0	$N^{(t)}$	seek to speak to a bank representative
1	$n^{(t)}_{(1,1)}$	abandon the call center when entering the system
1	$n^{(t}_{(2,1)}$	have to wait to speak with a bank representative
1	$n^{(t)}_{(3,1)}$	do not wait at all to speak to a bank representative
2	$n^{(t)}_{(1,2)}$	(virtual) abandon the call center when entering the system
2	$n^{(t)}_{(2,2)}$	abandon the queue while waiting
2	$n^{(t)}_{(3,2)}$	speak to a bank representative after waiting in queue
2	$n^{(t)}_{(4,2)}$	(virtual) do not wait at all to speak to a bank representative
3	$n^{(t)}_{(1,3)}$	(virtual) abandon the call center when entering the system
3	$n^{(t)}_{(2,3)}$	are called back after abandoning the waiting queue
3	$n^{(t)}_{(3,3)}$	are not called back after abandoning the waiting queue
3	$n^{(t)}_{(4,3)}$	(virtual) speak to a bank representative after waiting in queue
3	$n^{(t)}_{(5,3)}$	(virtual) do not wait at all to speak to a bank representative

Table 5.1. Descriptions of categories for call center

Notice that two categories are considered for a bank representative. one is through category 3 at stage 1, in which customers do not wait at all to speak to a bank representative; the other is through category 3 at stage 2, in which customers speak to a bank representative after waiting in queue. There is actually another potential path, which is when a customer that abandons in queue in category 2 at stage 2 is called back as ordered by a bank supervisor based on the customer's importance.

The second step in the input procedure is identifying all splitting processes. The unique category at stage 0 splits into three categories at stage 1. The number of customers that have to wait to speak with a bank representative $n_{(2,1)}$ at stage 1 splits into two categories at stage 2, i.e., categories 2 and 3 at that stage. Finally, the number of

customers that abandon the queue while waiting $n_{(2,2)}$ at stage 2 splits into two categories at stage 3, i.e. categories 2 and 3 at that stage.

The following relations among the realized numbers through stages hold:

In stage 1: $N^{(t)} = n^{(t)}_{(1,1)} + n^{(t)}_{(2,1)} + n^{(t)}_{(3,1)}$

Between stage 1 and stage 2: $n^{(t)}_{(1,1)}=n^{(t)}_{(1,2)}$, $n^{(t)}_{(2,1)}= n^{(t)}_{(2,2)} + n^{(t)}_{(3,2)}$, $n^{(t)}_{(3,1)}=n^{(t)}_{(4,2)}$

Between stage 2 and stage 3: $n^{(t)}_{(1,2)}=n^{(t)}_{(1,3)}$, $n^{(t)}_{(2,2)}= n^{(t)}_{(2,3)} + n^{(t)}_{(3,3)}$, $n^{(t)}_{(4,2)}=n^{(t)}_{(4,3)}$

For the monitoring procedure, the first step is transforming the multinomial probability tree into a binary probability tree in order to identify the tree fractions to be monitored. Thus, stage 1 in Figure 5.3 with three categories is transformed into two binary substages (1a and 1b). The binary probability tree to be monitored is shown in Figure 5.4 (arrows in bold font suggest tree fractions)**.**

The *p-tree* method of Chapter 3 is applied to every binary stage in the process. The unique category at stage 0 splits into two categories at substage 1a, with realized numbers $n^{(t)}_{(1,1)}$ and $N^{(t)}$-$n^{(t)}_{(1,1)}$ at sample *t*. The realized number $N^{(t)}$-$n^{(t)}_{(1,1)}$ represents those customers that do not abandon the call center when entering the system at sample *t*. Those customers that do not to abandon at substage 1a have a splitting into two categories at substage 1b, with realized numbers $n^{(t)}_{(2,2)}$ and $n^{(t)}_{(3,2)}$. Stages 2 and 3 are not transformed because they already have two categories as shown on Figure 5.3**.**

Notice that the splitting of final stages (e.g.: category 1 at stage 1) do not appear in Figure 5.4. As mentioned before, the further splitting of final stages is really required for the matrix representation.

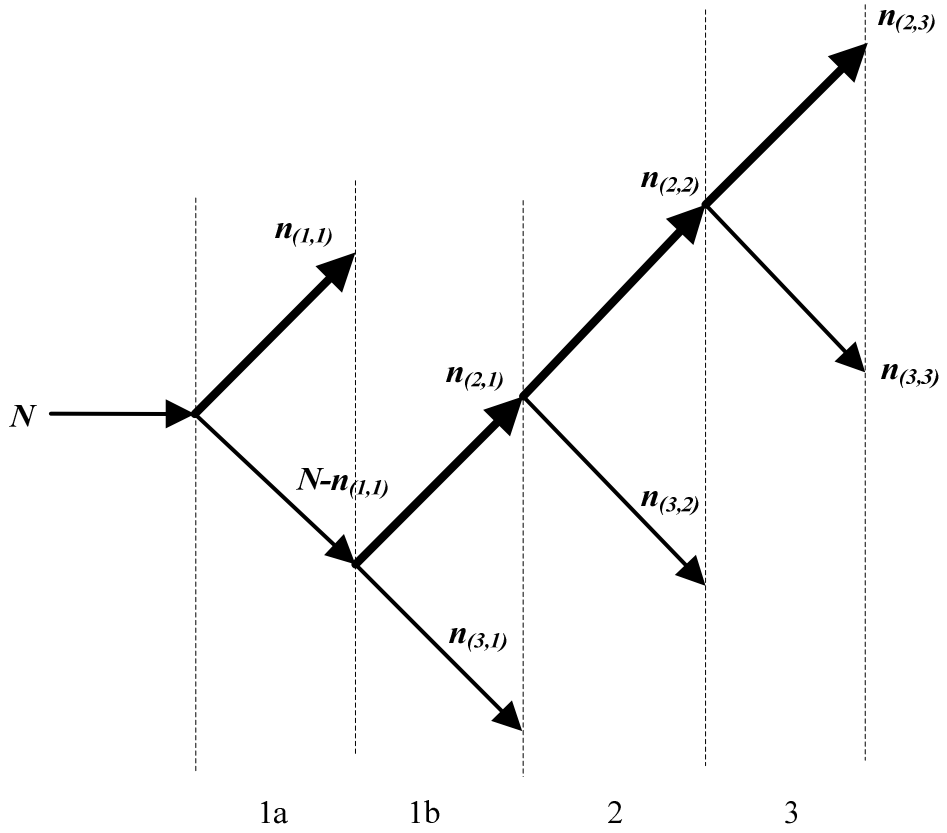

Figure 5.4. Binary probability tree across stages of call center

The second step in the monitoring procedure is determining the tree fractions through every binary substage in the binary probability tree of Figure 5.4. Table 5.2 shows the equations and descriptions of the tree fractions to be monitored. Every fraction has a numerator that equals the realized number at the first category of the binary substage and a denominator that equals the realized number in the root category at the previous substage.

Stage or substage	**Sample tree fraction and equation**	**Description of fraction**
1a	$\hat{f}^{(t)}{}_{(1,1)} = \frac{n^{(t)}{}_{(1,1)}}{N^{(t)}}$	number of customers that abandon the call center when entering the system over the total number of customers that seek to speak to a bank representative
1b	$\hat{f}^{(t)}{}_{(2,1)} = \frac{n^{(t)}{}_{(2,1)}}{N^{(t)} - n^{(t)}{}_{(1,1)}}$	number of customers that wait to speak to a bank representative over the number of customers that do not abandon the call center when entering the system
2	$\hat{f}^{(t)}{}_{(2,2)} = \frac{n^{(t)}{}_{(2,2)}}{n^{(t)}{}_{(2,1)}}$	number of customers that abandon the queue while waiting over number of customers that decided initially to wait to speak with a bank representative
3	$\hat{f}^{(t)}{}_{(2,3)} = \frac{n^{(t)}{}_{(2,3)}}{n^{(t)}{}_{(1,2)}}$	number of customers that are called back over number of customers that abandon the queue while waiting

Table 5.2. Tree fractions for call center (also as bold arrows in Figure 5.4)

The third step in the monitoring procedure is checking the multinomial distribution assumption through tests of independence among actual tree fractions for the in-control process. The fourth step in the monitoring procedure is monitoring the process. In Section

5.2.3 a simulated call center that follows the multinomial assumption is monitored using univariate control charts for the four independent tree fractions of Table 5.2**.**

5.2.2 Matrix representation of call center

The algorithm to monitor the call center is described here using matrix representation, which would be useful for software development of the algorithm. Recall that Figure 5.3 and Table 5.1 show the multinomial probability tree and its notation respectively.

For the input procedure, the first step is representing the tree structure using the following linking indicator matrices $\underline{\mathbf{L}_j}$, for j=1,2, 3. These matrices are obtained using eqn. (5-1) and are:

$$\underline{\mathbf{L}_1} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}, \quad \underline{\mathbf{L}_2} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \text{ and } \underline{\mathbf{L}_3} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The second step in the input procedure is recording the realized numbers transition matrices $\underline{\mathbf{S}_j}^{(t)}$ at sample t, for j=1,2,3. These matrices are obtained using eqn. (5-3) and are:

$$\underline{\mathbf{S}_1}^{(t)} = \begin{bmatrix} n^{(t)}_{(1,1)} & n^{(t)}_{(2,1)} & n^{(t)}_{(3,1)} \end{bmatrix},$$

$$\underline{\mathbf{S}_2}^{(t)} = \begin{bmatrix} n^{(t)}_{(1,2)} & 0 & 0 & 0 \\ 0 & n^{(t)}_{(2,2)} & n^{(t)}_{(3,2)} & 0 \\ 0 & 0 & 0 & n^{(t)}_{(4,2)} \end{bmatrix},$$

$$\text{and } \underline{\mathbf{S}_3}^{(t)} = \begin{bmatrix} n^{(t)}_{(1,3)} & 0 & 0 & 0 & 0 \\ 0 & n^{(t)}_{(2,3)} & n^{(t)}_{(3,3)} & 0 & 0 \\ 0 & 0 & 0 & n^{(t)}_{(4,3)} & 0 \\ 0 & 0 & 0 & 0 & n^{(t)}_{(5,3)} \end{bmatrix}$$

The matrices $\underline{\mathbf{S_j}}^{(t)}$ are time series in which some elements are active and can change over the samples and others are inactive. As shown on eqn. (5-3), the active elements are such that have a corresponding element $\underline{\mathbf{L_j}}[l,m]=1$.

For the monitoring procedure, the first step is determining the tree fraction matrices $\underline{\hat{\mathbf{F}}_\mathbf{j}}^{(t)}$, for j=1,23. The matrices $\underline{\hat{\mathbf{F}}_\mathbf{j}}^{(t)}$ are time series and their elements are obtained using eqn. (5-5) and are:

$$\underline{\hat{\mathbf{F}}_1}^{(t)}=\begin{bmatrix} \dfrac{n^{(t)}{}_{(1,1)}}{N^{(t)}} & \dfrac{n^{(t)}{}_{(2,1)}}{N^{(t)}-n^{(t)}{}_{(1,1)}} \end{bmatrix}, \qquad \underline{\hat{\mathbf{F}}_2}^{(t)}=\begin{bmatrix} 1 & 0 & 0 \\ 0 & \dfrac{n^{(t)}{}_{(2,2)}}{n^{(t)}{}_{(2,1)}} & 1 \\ 0 & 0 & 0 \end{bmatrix}, \quad \text{and}$$

$$\underline{\hat{\mathbf{F}}_3}^{(t)}=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \dfrac{n^{(t)}{}_{(2,3)}}{n^{(t)}{}_{(2,2)}} & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

The second step is determining which elements of these matrices $\underline{\hat{\mathbf{F}}_\mathbf{j}}^{(t)}$ can change or are active elements. The active tree fractions matrices $\underline{\mathbf{A_j}}$ show which elements of $\underline{\hat{\mathbf{F}}_\mathbf{j}}^{(t)}$ do change, and are determined using eqn. (5-6) as follows:

$$\underline{\mathbf{A_1}}=\begin{bmatrix} 1 & 1 \end{bmatrix}, \ \underline{\mathbf{A_2}}=\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \text{ and } \underline{\mathbf{A_3}}=\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Thus, the active tree fractions are the elements $\underline{\hat{\mathbf{F}}_1}^{(t)}[1,1]$, $\underline{\hat{\mathbf{F}}_1}^{(t)}[1,2]$, $\underline{\hat{\mathbf{F}}_2}^{(t)}[2,2]$, and $\underline{\hat{\mathbf{F}}_3}^{(t)}[2,2]$. Notice that these active elements have the same equations of the tree fractions

in Table 5.2. These tree fractions have in-control values that are calculated using eqn. (F-2) of Appendix F, and are 0.050, 0.632, 0250, and 0.200 respectively.

There is a total of K_F=5 final categories in the call center, so there is a total of K_F-1=4 active tree fractions out of 27 total elements in the matrices $\underline{\hat{\mathbf{F}}_j}^{(t)}$. As suggested before, other computational structures of data might be further investigated in order to minimize the proportion of inactive elements in the monitoring procedure.

5.2.3 Monitoring a simulated call center

Here a simulated call center is monitored to illustrate the application of the methodology proposed in Section 5.2.1. Two scenarios of call centers are simulated as shown in Figure 5.5: an in-control scenario with the same probability parameters shown in Figure 5.3**;** and an out-of-control scenario with probability parameters as shown in Figure 5.5b. Both scenarios have constant volume $N^{(t)}$=1,000 customers per period of time (about a daily actual number of customers). Stage 1 is simulated as a multinomial process with three categories, and stages 2 and 3 are simulated as binomial processes each one with samples size $n^{(t)}_{(2,1)}$ and $n^{(t)}_{(2,2)}$ respectively. The desired ARL_0 is 84 samples (equivalent to 84 days or 7 weeks). The number of runs in the simulation is such that each standard error is less or equal than 0.02ARL.

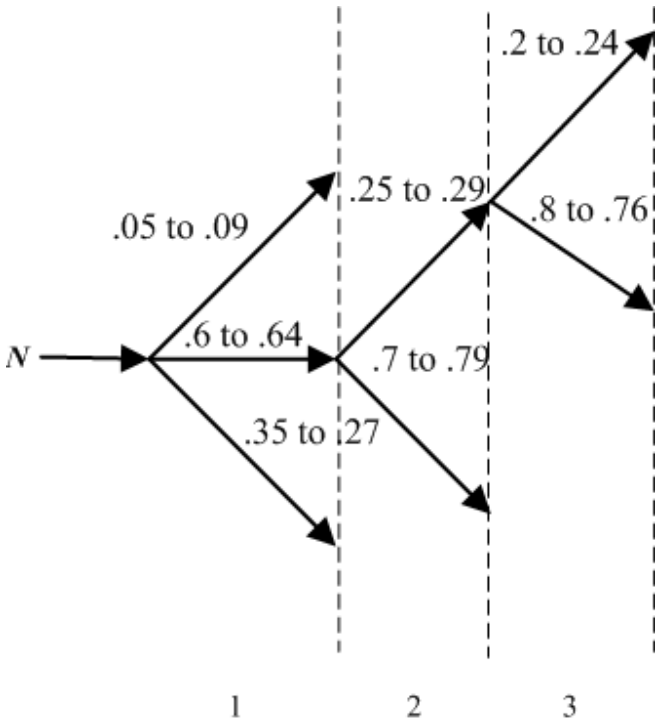

Figure 5.5 Shifts in call center process

Section 5.2.1 above described with enough detail the input procedure and the first two steps of the monitoring procedure. The first step of the monitoring procedure was building the binary probability tree as shown in Figure 5.4**.** The second step of the monitoring procedure was getting the tree fractions across stages as shown in Table 5.2.

The third step of the monitoring procedure is checking the multinomial distribution assumption. Kendall nonparametric tests independence (Kendall & Gibbons, 1990, p. 66) among tree fractions were performed for the in-control scenario with 200 samples. The p-values are between 0.12 and 0.88, so the null hypotheses of independence are not rejected.

The fourth step of the monitoring procedure is monitoring the tree fractions. Because of the theoretical and empirical independence among tree fractions, univariate control charts are constructed for each of the fractions. Each tree fraction is monitored either by a

p-chart or by a CUSUM Arcsine method with the same individual false alarm rate (using eqn. (5-10)). The process is monitored in Phase II, and the performance measures are the process ARL and the individual tree fraction ARLs.

Table 5.3 shows the ARL results. The first column shows what is being monitored – the total process or each tree fraction. The second, third, and fourth columns show the stage (or substage), the expected sample size for each fraction, and the shift size in multiples of the standard deviation of each fraction. The last four columns show the ARL results by scenario and by method used.

				ARLs results			
				In-control		Out-of-control	
Monitor	Stage or Substage	Expected sample size	Shift size	*p*-chart(s)	CUSUM Arcsine chart(s)	*p*-chart(s)	CUSUM Arcsine chart(s)
Process	All			81	83	1.0	1.1
$\hat{f}^{(t)}_{(1,1)}$	1a	1000	5.8	329	325	1.0	1.6
$\hat{f}^{(t)}_{(2,1)}$	1b	950	4.6	305	329	1.1	1.7
$\hat{f}^{(t)}_{(2,2)}$	2	600	2.3	322	341	3.7	3.2
$\hat{f}^{(t)}_{(2,3)}$	3	150	1.2	340	325	14.6	6.4

Table 5.3. ARL results for simulated call center. Total desired ARL_0=84

Table 5.3 shows that:

- Both methods achieve the desired total ARL_0 of 84.
- The higher the fraction on Table 5.3, the higher the sample size, the higher the shift size, the better the sensitivity (lower the ARL).

- As expected, the *p*-chart has better sensitivity than the CUSUM Arcsine for shifts size over 3σ ($\hat{f}^{(t)}_{(1,1)}$ and $\hat{f}^{(t)}_{(2,1)}$), and the CUSUM Arcsine has better sensitivity than the *p*-chart for shifts under 3σ ($\hat{f}^{(t)}_{(2,2)}$ and $\hat{f}^{(t)}_{(2,3)}$).

For example, if the user wishes to increase the sensitivity of the fractions $\hat{f}^{(t)}_{(2,2)}$ and $\hat{f}^{(t)}_{(2,3)}$, then their false alarm rates should be increased and the false alarm rate of the fractions $\hat{f}^{(t)}_{(1,1)}$ and $\hat{f}^{(t)}_{(2,1)}$ should be decreased in order to keep a desired total false alarm rate for the system. Another possibility would be to redefine stage 3 as stage 1, although altering the natural order of the stages would complicate the interpretations.

5.3 Concluding remarks about monitoring multistage and multicategory processes

We propose a methodology to monitor and interpret multistage and multicategory processes. The proposed methodology decomposes the process into binary stages and substages, which can be monitored with a set of so called tree fractions. Thus, these tree fractions describe the relation within and among stages and provide full interpretations across the process. The sufficient number of tree fractions equals the number of final categories minus one, where final categories are those that do not have any further splits in the process.

We show that if a multinomial distribution fits every stage of the process, then these tree fractions are independent and can be monitored with individual control charts. The proposed methodology can be expressed as an algorithm using matrix representation, which can be useful for a further computational programming of the algorithm.

The proposed methodology is not limited by the number of stages and categories. We show that the CUSUM Arcsine proposed in Chapter 4 can help both achieving a false alarm rate at process level and at individual fraction level, as well as improving sensitivity. However, a large number of categories could imply that some tree fractions with very low sample sizes would not get appropriate sensitivities.

The order of the stages and categories has an impact on the sample sizes of tree fractions, and therefore has an impact on the sensitivity of every tree fraction being monitored. The user may order the categories according to their monitoring importance within each stage. The user could also reorder the stages and get a new multinomial probability tree as long this new tree might make sense. For example, the user may redefine stage 1 of the call center as a splitting of customers among two categories:

customers that wait in queue for a bank representative and other customers. Monitoring this new tree would emphasize detecting changes on customers that have to wait in queue.

6 Future research

I propose several research topics to extend the work of this dissertation:

- Monitoring non-multinomial MSMC processes.
- Monitoring routing matrices in queuing systems.
- Monitoring waiting and service times in multistage processes.
- Monitoring multifacility MSMC processes.
- Forecasting and monitoring in service.
- Testing and interpreting associations in contingency tables using trees.

Monitoring non-multinomial multistage and multicategory processes

Here we propose to investigate how to monitor MSMC processes in which the process' data is not multinomial distributed and therefore the method proposed in Chapter 5 may not be applied. The multinomial assumption implies that the binary substages are independent and binomial distributed. However, this assumption is not always fulfilled. At least two situations may occur:

- Tree fractions that are correlated with the volume and also correlated among themselves
- Tree fractions that present overdispersion with respect to the binomial distribution.

In case of processes whose tree fractions are correlated with the volume and also correlated among themselves, multivariate methods need to be developed. Existing methods to be explored are commented next.

Sulek *et al.* (2006) monitor a two-stage service process in a retail operation, using a linear regression method of Wade and Woodall (1993) that relates the two stages. Woodall *et al.* (2004) as well as Kang and Albin (2000) propose the profile method to monitor processes in which there is a relation between a response variable and one or more covariates. If this relation is linear, then not only the residuals are monitored, but also the estimated slope and intercept.

Another approach is proposed by Jearkpaporn *et al.* (2005) and Skinner *et al.* (2003, 2004). They propose monitoring methods based on generalized linear models (GLM), which allow modeling various distributions for dependent variables and include covariates that can represent customers' characteristics.

With respect to the tree fractions that present overdispersion with respect to the binomial distribution, two approaches may be attempted:

- Identifying groups of customers that share specific behaviors and hence split the data set into subsets that fulfill the binomial assumption of binary substages.
- Fitting other distributions. This may consider analyzing whether groups of customers arrive in clusters as responding to specific causes (commercial campaigns, deadlines, etc). However, Jackson (1972, p 91) points out that a process must be in-control or stable in order to fit a distribution well. Fitting other discrete distributions may be tried as in Friedman and Albin (1991) and Jackson (1972). Mixes of binomial may represent distinct groups of customers and generalized Poisson distributions (see Table 6.1) may address overdispersion and/or clusters. If none distribution can fit the data, the distribution free CUSUM

Box-Cox method proposed in Chapter 4 could be applied to monitor fractions (if they are independent yet).

Distribution	**Distribution of No. of Clusters**	**Distribution of Counts per Cluster**
Neyman type A	Poisson	Poisson
Thomas	Zero-truncated Poisson	Poisson
Poisson binomial	Poisson	Zero or some constant
Negative binomial, also knows as Polya-Eggenberger	Gamma	Poisson
Neyman type B	Poisson	Uniform
Neyman type C	Poisson	Triangular

Table 6.1. Generalized Poisson distributions

Here we consider monitoring the call center using actual data for a year of operation. We show that the volume and tree fractions are correlated, so they can not be monitored with individual control charts as proposed by the methodology developed in Chapter 5 for MSMC processes. We show that the tree fractions also present overdispersion with respect to the binomial distribution.

The data set with a year of operation was obtained thanks to Professor Avishai Mandelbaum of the Israel Institute of Technology. Recall that the first step of the monitoring methodology is building the binary probability tree as shown in Figure 5.4, and the second step of the monitoring procedure is getting the tree fractions as shown in Table 5.2.

We choose to monitor the process on a weekly basis, which provides 52 samples for the complete year, although the user may use a different level of aggregation (daily,

biweekly, monthly, etc). Table 6.2 shows the descriptive statistics of the volume and the tree fractions on a weekly basis.

Measure	Mean	SD	Minimum	Maximum
$N^{(t)}$	8547	1349	5757	11676
$\hat{f}^{(t)}_{(1,1)}$	0.05	0.02	0.03	0.12
$\hat{f}^{(t)}_{(2,1)}$	0.62	0.14	0.28	0.85
$\hat{f}^{(t)}_{(2,2)}$	0.24	0.05	0.16	0.36
$\hat{f}^{(t)}_{(2,3)}$	0.19	0.06	0.07	0.32

Table 6.2. Descriptive statistics of the volume and tree fractions (weekly basis)

In the third step of the monitoring procedure we find that the multinomial assumption is violated because there are dependencies among the tree fractions. Consider Figure 6.1 that shows scatter plots among the volume *N* and the four tree fractions. Figure 6.1 clearly shows empirical correlation between $\hat{f}^{(t)}_{(2,1)}$ and $\hat{f}^{(t)}_{(2,2)}$, which violates the independence assumption. The interpretation is that the fraction of customers that wait in queue ($\hat{f}^{(t)}_{(2,1)}$) is linearly and positively correlated with the fraction of those waiting customers that abandon the queue ($\hat{f}^{(t)}_{(2,2)}$).

Through tests of hypotheses, we observe that there are correlations among the volume *N* and stages 1 and 2, i.e., among $N^{(t)}$ and the tree fractions $\hat{f}^{(t)}_{(1,1)}$, $\hat{f}^{(t)}_{(2,1)}$ and $\hat{f}^{(t)}_{(2,2)}$. Additionally, depending on the test, $N^{(t)}$ may be correlated with $\hat{f}^{(t)}_{(2,3)}$. This is shown in Table 6.3, which contains a sample correlation matrix among the volume $N^{(t)}$ and the four tree fractions. The first number in each cell is the Pearson correlation coefficient, and the numbers in parenthesis are p-values for two null hypotheses of independence tested. The first null hypothesis is whether the Pearson correlation coefficient equals zero, and the

second null hypothesis relates to the Kendall nonparametric test of independence. All p-values under a type I error of 0.05 are marked in bold.

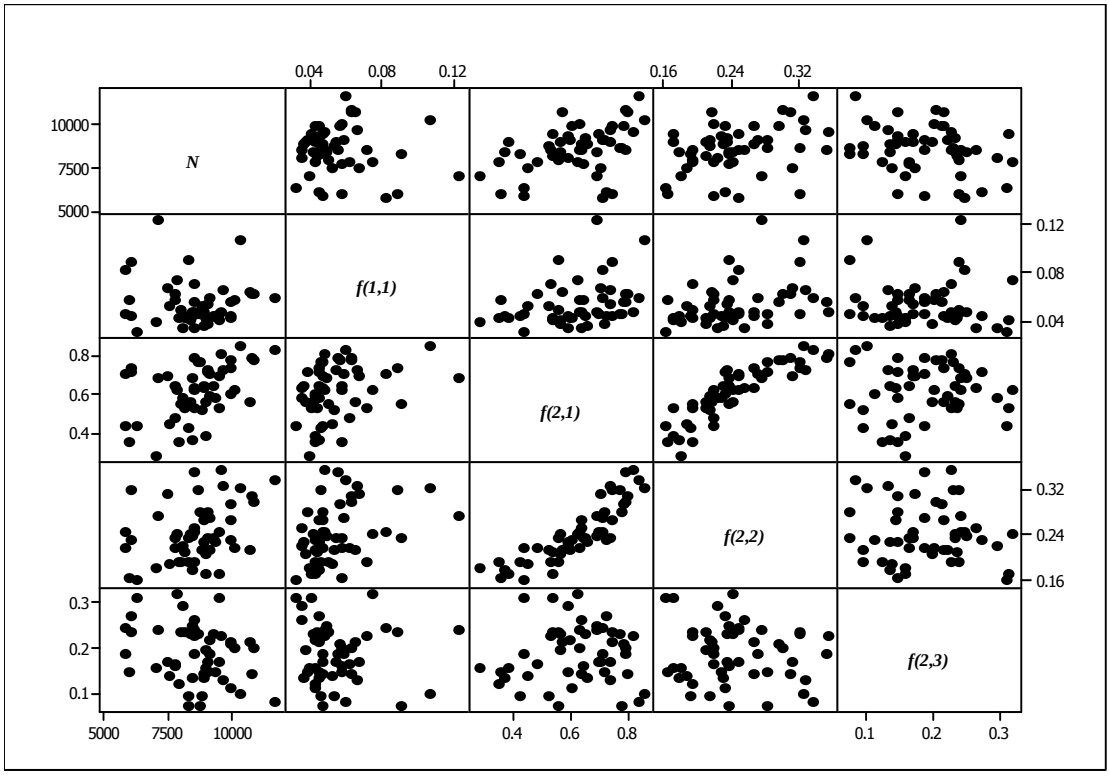

Figure 6.1. Multiple scatter plot among *N* and tree fractions

	$N^{(t)}$	$\hat{f}^{(t)}_{(1,1)}$	$\hat{f}^{(t)}_{(2,1)}$	$\hat{f}^{(t)}_{(2,2)}$
$\hat{f}^{(t)}_{(1,1)}$	-0.07 (0.6, 0.9)			
$\hat{f}^{(t)}_{(2,1)}$	0.44 **(0.001, 0.001)**	0.33 **(0.02, 0.01)**		
$\hat{f}^{(t)}_{(2,2)}$	0.36 **(0.01, 0.01)**	0.37 **(0.006, 0.003)**	0.87 **(0.000, 0.000)**	
$\hat{f}^{(t)}_{(2,3)}$	-0.28 **(0.04,** 0.06)	-0.09 (0.5, 0.5)	0.05 (0.7, 0.8)	-0.12 (0.4, 0.7)

Table 6.3. Sample correlation matrix and p-values for null hypotheses

The fourth step of the monitoring procedure is monitoring the tree fractions. The volume $N^{(t)}$ and the tree fractions cannot be subject to individual monitoring and interpretations because they are correlated as shown by Table 6.3.

As an illustration that the call center seems to be in a non-stationary mode, here we set a MEWMA control chart (multivariate EWMA as proposed by Lowry *et al.* (1992) and Prabhu and Runger (1997)) for the volume $N^{(t)}$ and the four tree fractions with an ARL_0=100 weeks (using an yearly estimate of a covariance matrix and without removing any sample for calculations). The MEWMA chart in Figure 6.2 suggests that the process is non-stationary during most of the weeks. A better knowledge of the process would allow identifying appropriate special causes. For example, changes in staffing allocations or changes in procedures can be related to special causes.

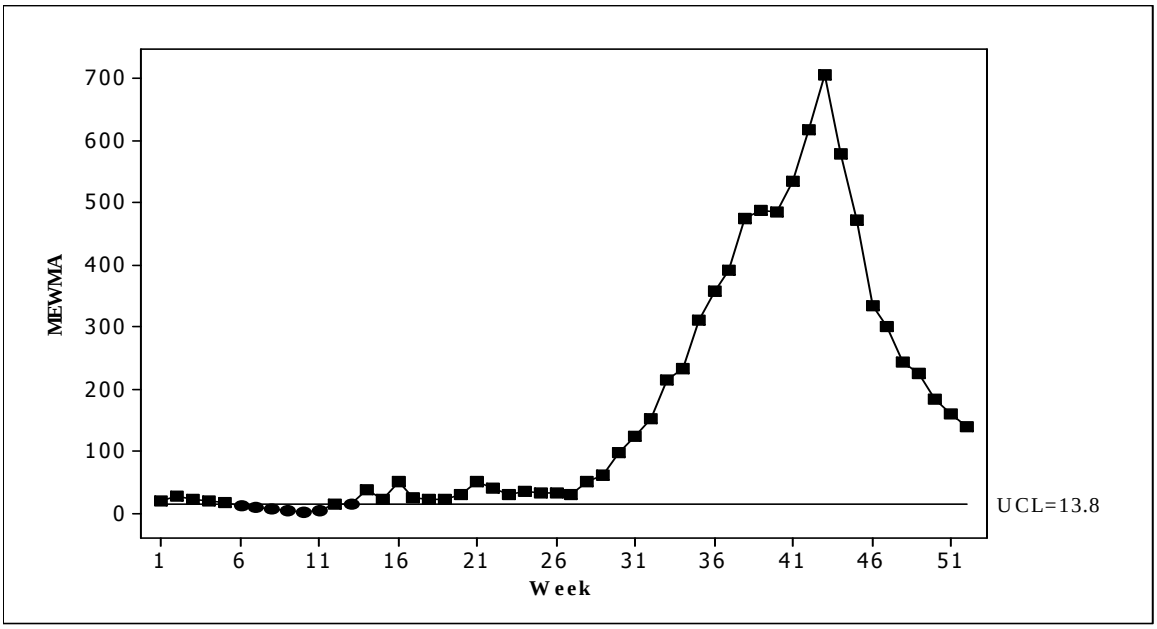

Figure 6.2. MEWMA control chart for $N^{(t)}$ and the four tree fractions. ARL_0=100 weeks

Here we explore the method of Sulek *et al.* (2006), which monitors a two-stage service process in a retail operation, using a linear regression method of Wade and

Woodall (1993) that relates the two stages. We show that linear regressions among call center's tree fractions may help but do not explain most of the process variation.

Consider the positive correlation between $\hat{f}^{(t)}_{(2,1)}$ and $N^{(t)}$, and between $\hat{f}^{(t)}_{(2,1)}$ and $\hat{f}^{(t)}_{(1,1)}$ as shown in Table 6.3. This means that the chance that the chance that a customer has to wait in queue increases with the volume of customers or with the fraction of customers that abandon the system. However, a linear regression of $\hat{f}^{(t)}_{(2,1)}$ on $N^{(t)}$ and $\hat{f}^{(t)}_{(1,1)}$ gets a R^2 statistic of 0.32, i.e., only 0.32 of the variation of $\hat{f}^{(t)}_{(2,1)}$ is explained by the regression.

Table 6.3 also shows that the fraction $\hat{f}^{(t)}_{(2,2)}$ is correlated with $N^{(t)}$, $\hat{f}^{(t)}_{(1,1)}$, and $\hat{f}^{(t)}_{(2,1)}$. In other words, stage 2 is correlated with the volume and with stage 1. A linear regression of $\hat{f}^{(t)}_{(2,2)}$ on $N^{(t)}$, $\hat{f}^{(t)}_{(1,1)}$, and $\hat{f}^{(t)}_{(2,1)}$ gets a satisfactory R^2 of 0.77, and only the coefficient associated to $\hat{f}^{(t)}_{(2,1)}$ is significant. The fitted regression equation is $\hat{f}^{(t)}_{(2,2)} = 0.043 + 0.32\, \hat{f}^{(t)}_{(2,1)}$, with residuals normally distributed. The interpretation here is that the higher the fraction of customers that wait in queue ($\hat{f}^{(t)}_{(2,1)}$.), the higher the fraction of those waiting customers that abandon the queue ($\hat{f}^{(t)}_{(2,2)}$).

As an illustration that the call center also presents overdispersion with respect to the binomial distribution, see Figure 6.3 that shows a binomial based 3-sigma *p*-chart on the actual fraction $\hat{f}^{(t)}_{(1,1)}$. Figure 6.3 shows that the control limits are too tight to take into account this violation of assumptions. This problem has been observed in processes with large sample sizes as in Heimann (1996). It can be shown also that N and $n_{(1,1)}$ have

overdispersion with regards to the Poisson distribution, which is consistent with the research of Borst *et al.* (2004, p. 32).

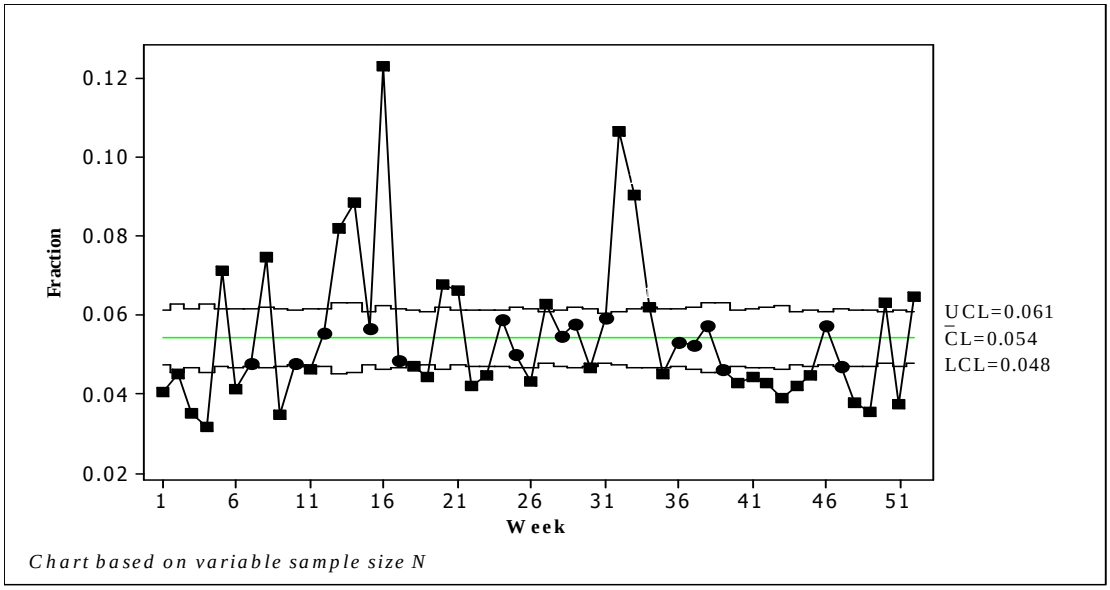

Figure 6.3. Binomial based 3-sigma *p*-chart for $\hat{f}^{(t)}_{(1,1)}$ shows overdispersion

Monitoring routing matrices in queuing systems

The multistage and multicategory processes considered in Chapter 5 impose several restrictions that can be relaxed for future research. Multistage processes in queuing systems may accept loops, i.e., customers that return to a category of a previous stage including a return to the same category. In this context, categories may be redefined as states, and the process can represent the routing of customers, transactions, or messages across a network. These processes can be represented as routing matrices in which each row is an array of probabilities from going from a certain state to another one.

Monitoring the variations on the estimates of these probabilities can be a relevant problem. Loops can introduce a source of significant correlation among stages, so multivariate methods should be investigated.

Monitoring waiting and service times in multistage processes

Multistage processes can involve queues across stages as shown in the call center example. It can be relevant for example to monitor waiting times instead of just a fraction of customers that need to wait for attention. The problem is that these fractions put together customers that wait a few seconds with customers that wait a lot more.

The call center used as an example has at least four instances of times: the time in VRU when entering the system in stage 1, the time in queue until abandoning in stage 2, the time in queue until speaking to a bank representative, and lastly the service time. Monitoring times in queue also could involve censored data, as shown by Mandelbaum *et al.* (2001). The time until abandoning is a censored observation with respect to the needed time to wait, and the time until speaking to a bank representative is censored with respect to the time that a customer is willing to wait.

In terms of existing methods that could be explored: Shore (2006) proposes to monitor the number of customers in a queuing system; Steiner and Mackay (2000) propose methods for monitoring censored data in manufacturing.

Monitoring multifacility MSMC processes

Consider that a multifacility, i.e., a multiple districts in which a MSMC process fits the operation of every district. The management would wish to monitor the multifacility, and then different monitoring interfaced should be built according with the user profile. Examples of users are: top level decision makers, regional managers, district managers,

SPC experts, and operating personnel. Benneyan *et al.* (2000) approaches a multifacility web-based monitoring system for health care.

Forecasting and monitoring in service applications

In many service applications the data are autocorrelated and it can be more relevant to forecast the current mean than to detect a change from a baseline or target. Methods that model actual service data as time series and then set monitoring methods could be developed.

In terms of existing methods, Montgomery (2005, p. 446) suggests a EWMA chart as a one-step-ahead forecast for correlated data. According to Montgomery and Mastrangelo (1991), if the process can be modeled with a first order integrated moving average model (ARIMA), a EWMA can be designed to be the best forecast of the process mean. Yashchin (1993) also suggests using a EWMA to forecast the process level. Montgomery and Mastrangelo (1991) propose a moving center-line EWMA method for autocorrelated data (MCEWMA). Boyles (2000) approaches the analysis of autocorrelated processes in Phase I for either stationary or non-stationary time series.

For the call center example, Figure 6.4 shows a time series of fraction $\hat{f}_{(2,2)}$ and its related EWMA (with smoothing parameter=0.2). Fraction $\hat{f}_{(2,2)}$ measures the number of customers that abandon the queue while waiting over the number of customers in queue waiting to speak with a bank representative. Fraction $\hat{f}_{(2,2)}$ has a mean of 0.24 and a range between 0.16 and 0.36. It can be shown that $\hat{f}_{(2,2)}$ has a significant autocorrelation of at least first order. In cases like this, it does not make sense to detect whether the

process is in-control or not with respect to a baseline. Many questions arise: What is the period of time to be considered in a Phase I study? All year? Some weeks and which weeks?

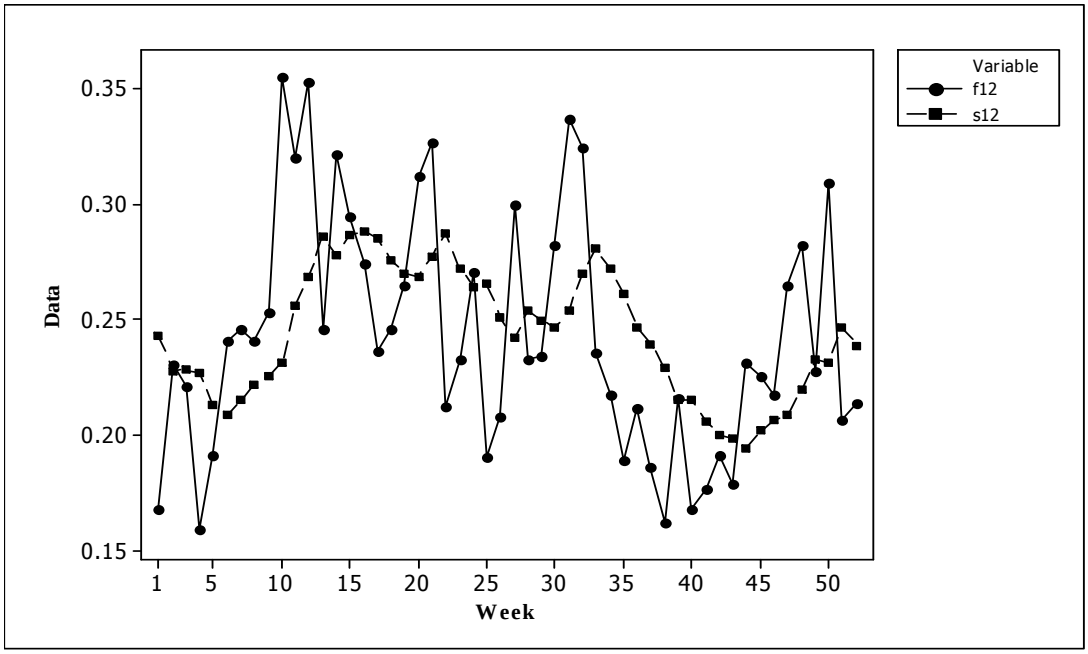

Figure 6.4. Time series of actual fraction $\hat{f}_{(2,2)}$ and its EWMA

Interpreting associations in contingency tables using trees

This future research proposes to use the probability tree methodology developed in Chapter 3 to test and interpret associations in contingency tables. The idea is based in that under the null hypothesis of independence; every row (and every column) is multinomial distributed. The proposed idea is explained using the example in Table 6.4, which shows a contingency table about gender and party identification (Ref.: Agresti (1996, p. 31)).

Gender	Democrats	Non-affiliated	Republicans	Total
Females	279 (261.4)	73 (70.7)	225 (244.9)	577
Males	165 (182.6)	47 (49.3)	191 (171.1)	403
Total	444	120	416	980

Table 6.4. Cross classification of party identification by gender (frequencies under independence in parenthesis)

The null hypothesis of independence between party identification and gender may be tested with a chi-square test. The chi-square statistic here is X^2=7.01. Under independence, X^2 is distributed chi-square with two degrees of freedom. This test has a p-value of 0.03, so the null hypothesis of independence is rejected for any type I error greater than this level.

According to Agresti (1996, p. 33), the chi-square test of independence simply indicates the degree of evidence for an association. Agresti recommends interpreting the nature of the association through techniques such as decomposition or partition of chi-square into components, analysis of residuals, and odds-ratios.

The method proposed here uses the probability tree technique to formulate a new test of independence, which also provides interpretations. For the example, the probability tree technique decomposes this multinomial with 3 categories shown in Table 6.4 into 2 binary substages as shown in Figures 6.5 and 6.6 as well as in Table 6.5 (integers under labels represent volumes and decimal numbers over arrows represent fractions). The tree categories are ordered as Non-affiliated, Democrat, and Republican, although other orders could also be proposed.

For the example, the proposed method would perform two independent tests in Figure 6.5: a first test that the probability of a non-affiliated female (sample tree fraction is 0.127) equals the probability a non-affiliated male (sample tree fraction is 0.117); and a second test that among those affiliated subjects, the probability of a female Democrat (sample tree fraction is 0.554) equals the probability of a male Democrat (sample tree fraction is 0.463). Additionally and for comparisons, Figure 6.6 shows that the sample fraction of non-affiliated subjects (both females and males) is 0.122 and the sample fraction of Democrats among affiliated subjects (both females and males) is 0.516.

This dissertation proposes to measure the power and significance of this tree association method and compare its interpretation with other techniques such as the partition of chi-square into components, analysis of residuals, and odds-ratios, as shown by Agresti (1996, p. 31-33). The method should consider that the order of categories in the tree affects the results, and also approach contingency tables with more rows and columns than the example just shown.

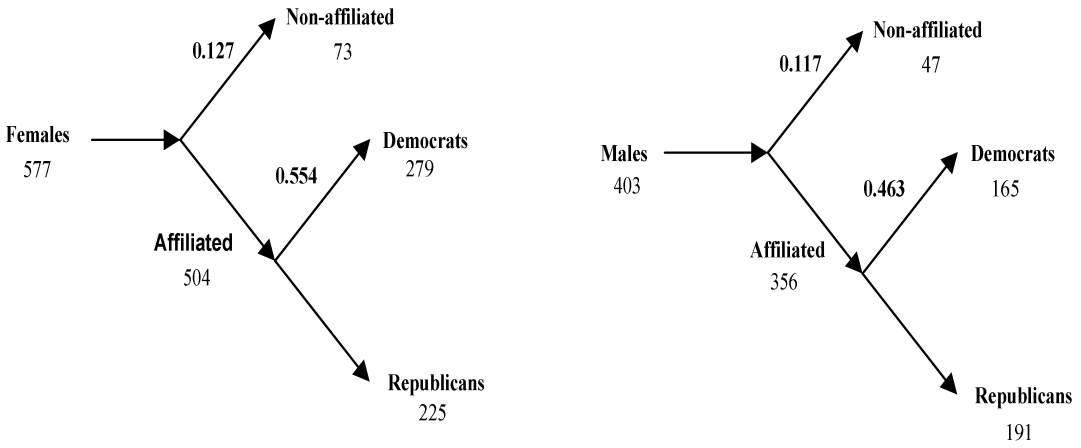

Figure 6.5. Binary probability trees for party identification for females and for males

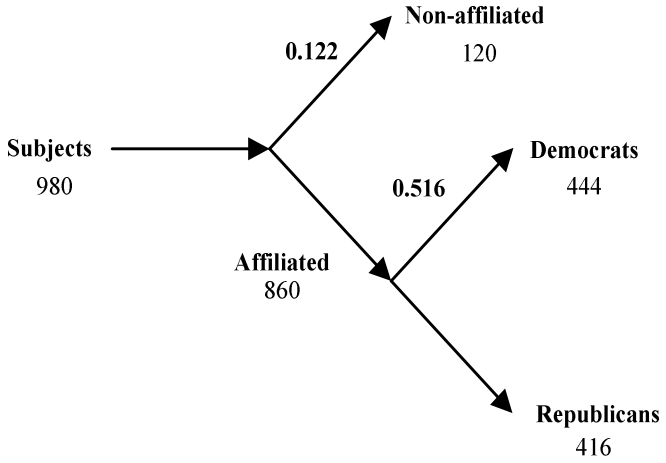

Figure 6.6. Binary probability tree for party identification for females and males subjects

Gender	Non-affiliated/Total	Democrats/(Republicans or Democrats)
Females	0.127	0.554
Males	0.117	0.463
Total Subjects	0.122	0.516

Table 6.5. Tree fractions for party identification by gender

7 Conclusions

Many processes function through routing transactions in a succession of stages and multiplicity of categories. This dissertation has developed a monitoring method to detect changes in the baseline fractions that characterize a multistage and multicategory process. Detecting these changes may allow users to understand the variation of the process, analyze its special and common causes, keep the process in statistical control as well as introduce improvements in the process.

Monitoring multistage and multicategory processes requires a thorough knowledge of the process. Initially, the process has to be mapped using flowcharts and business process diagrams. Users with different profiles should contribute to this process' description. The monitoring method proposed requires that the process be represented as a tree in which the stages and the splitting of categories are visualized. These so called multinomial probability trees are decomposed into binary substages using a binary probability tree. If every stage and splitting of the process is multinomial distributed, then the binary substages are independent and binomial distributed, and the process can be monitored through independent tree fractions.

This dissertation first proposes methods for monitoring a single stage process with multiple categories (Chapter 3), and monitoring a fraction in single stage process with two categories (Chapter 4). These two proposed methods have their own merits and also contribute to the development of the proposed method for monitoring multistage and multicategory process (Chapter 5). The proposed methodology can be expressed as an

algorithm using matrix representation, which can be useful for a further development of software for the algorithm.

Both the *p-tree* method for a single stage with multiple categories and the method for monitoring multistage and multicategory process share a number of features:

- The tree fractions provide full interpretations across the process. If a chart for a tree fraction signals, it is straightforward to identify the stage and category causing the disturbance.
- The order of the stages and categories matters in terms of describing the process and in terms of monitoring properties such as sensitivity and achieving a desired false alarm rate. Thus, the user may order the categories according to their monitoring importance within each stage.
- The methods are not limited by the number of categories per stage. Although eventual very low samples sizes of some tree fractions can compromise the sensitivity and false alarm rates of monitoring methods as shown in Chapter 4.

The CUSUM Arcsine proposed in Chapter 4 can be a part of the method to monitor multistage and multicategory process. The CUSUM Arcsine method can monitor every tree fraction, contributing to achieving a false alarm rate at process level and at individual fraction level, as well as improving sensitivity of the method.

I propose several research topics to extend the work of this dissertation:

- Monitoring non-multinomial MSMC: data of actual process may not fit the multinomial assumption, and the binary substages may not be binomial distributed.

- Monitoring routing matrices: the methodology can be extended to processes where the transactions do not always move forward, e.g., loops. It may have applications in telecommunications, and financial markets.
- Monitoring waiting and service times in multistage processes: for example monitoring waiting and service times in call centers.
- Forecasting and monitoring in service: for processes that are autocorrelated and where forecasting is more relevant than deviations from a baseline.
- Testing and interpreting associations in contingency tables using trees: a novel method based on applying probability trees to represent contingency tables.

8 References

Acosta-Mejia C.A. (1999). “Improved *p* charts to monitor process quality”. *IIE Transactions*, Vol. 31, p. 509-516.

Agrawal R., Lawless J.F., and Mackay R.J. (1999). “Analysis of variation transmission in manufacturing processes—Part II”. *Journal of Quality Technology*, Vol. 31, n 2, p. 143–154.

Agresti A. (1996). *An Introduction to Categorical Data Analysis.* Wiley-Interscience publication. John Wiley & Sons Inc. New York.

Andersson E., Bock D., and Frisén M (2005). “Statistical surveillance of cyclical processes with application to turns in business cycles”. *Journal of Forecasting*, Vol. 24, p. 465-490.

Benneyan J.C. (2000). ”Development of a web-based multifacility healthcare surveillance information system”. *Journal of Health Care Information Management*, Vol. 14, n 3, p. 19-26.

Benneyan J.C. (2006). ”Discussion in: the use of control charts in health-care and public-health surveillance”. *Journal of Quality Technology*, Vol. 38, n 2, p. 113-123.

Beygelzimer A, Brodie M, Sheng Ma, and Rish I. (2005). “Test-based diagnosis: tree and matrix representations”. *2005 9th IFIP/IEEE International Symposium on Integrated Network Management* (*IEEE Cat No. 05EX1060*), 2005, p. 529-42.

Borst S., Mandelbaum A., and Reiman M.I. (2004). “Dimensioning large call centers”. *Operations Research*. Vol. 52, n 1, January–February, p. 17–34.

Bourke P.D. (2001). “Sample size and the binomial CUSUM control chart: the case of 100% inspection”. *Metrika*, Vol 53, p. 51-70.

Boyles R.A. (2000). “Phase I analysis for autocorrelated processes”. *Journal of Quality Technology*, Vol. 32, n 4, p. 395-409.

Brown L., Gans N., Mandelbaum A., Sakov A., Shen H., Zeltyn S., and Zhao L. (2002). “Statistical analysis of a telephone call center: a queuing-science perspective”. Technical report 03-12, Working Paper Series. The Wharton School, University of Pennsylvania, Philadelphia, PA.

Chakraborti S. and Human S.W. (2006). “Parameter estimation and performance of the *p*-chart for attributes data”. *IEEE Transactions on Reliability*, Vol. 55, n 3, September, p. 559-566.

Chan L.Y., Dennis K.J., Xie M., and Goh T.N. (2002). “Cumulative probability control charts for geometric and exponential process characteristics”. *International Journal of Production Research*, Vol. 40, n 1, p. 133-150.

Chan L.Y., Lai C.D., Xie M., and Goh T.N. (2003). “A two-stage decision procedure for monitoring processes with low fraction nonconforming”. *European Journal of Operational Research*, Vol. 150, p. 420–436.

Chen G. (1998). “An improved *p* chart trough simple adjustments”. *Journal of Quality Technology*, Vol. 30, n 2, p. 142-151.

Chesher D. and Burnett L. (1996). Using Shewhart *p* control charts of external quality-assurance program data to monitor analytical performance of a clinical chemistry laboratory”. *Clinical Chemistry*, Vol. 42, n 9, p. 1478-1482.

Chung S.H., Pearn W.L., and Yang Y.S. (2007). "A comparison of two methods for transforming non-normal manufacturing data". *International Journal of Advanced Manufacturing Technology,* Vol. 31, p. 957-968.

Coleman S.Y., Arunakumar G., Foldvary F., and Feltham R. (2001). "SPC as a tool for creating a successful business measurement framework". *Journal of Applied Statistics,* Vol. 28, n 3-4, p. 325-334.

Devor R.E., Chang T., and Sutherland J.W. (2007). *Statistical Quality Design and Control: Contemporary Concepts and Methods.* 2nd Edition. Pearson Prentice Hall. Upper Saddle River, New Jersey.

Ding Y., Shi J., and Ceglarek D. (2002a). "Fault diagnosis of multistage manufacturing processes by using state space approach". *Journal of Manufacturing Science and Engineering, Transactions of the ASME,* Vol. 124, p. 313–322.

Ding Y., Shi J., and Ceglarek D. (2002b). "Diagnosability analysis of multi-station manufacturing processes". *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME,* Vol. 124, n 1, March, p. 1-13.

Duda R.O., Hart P.E., and Stork D.G. (2001). *Pattern Classification*. 2nd Edition. John Wiley & Sons, Inc.

Duran R.I. and Albin S.L. (2009a). "Monitoring and accurately interpreting service processes with transactions that are classified in multiple categories". In print at *IIE Transactions*.

Duran R.I. and Albin S.L. (2009b). "Monitoring a fraction with simple and reliable settings of the false alarm rate". In print at *Quality and Reliability Engineering International*. Available online as an early view (since 03-20-2009).

Friedman D.J. and Albin S.L. (1991). "Clustered defects in IC fabrication: impact on process control charts". *IEEE Transactions on Semiconductor Manufacturing,* 4, p. 36-42.

Gadre M.P. and Rattihalli R.N. (2005). "Unit and group-runs control chart to identify increases in fraction nonconforming". *Journal of Quality Technology,* Vol. 37, n 3, p. 199-209.

Gan F.F. (1990). "Monitoring observations generated from binomial distribution using modified exponentially weighted moving average control chart". *Journal of Statistical Computation and Simulation,* Vol. 37, p. 45-60.

Gan F.F. (1993). "An optimal design of CUSUM control charts for binomial counts". *Journal of Quality Technology,* Vol. 20, n 4, p. 445-460.

Gans N., Koole G., and Mandelbaum A. (2003). "Telephone call centers: tutorial, review, and research prospects". *Manufacturing & Service Operations Management,* Vol 5, n 2, Spring, p. 79-141.

Gardiner S.C. and Mitra A. (1994). "Quality control procedures to determine staff allocation in a bank". *International Journal of Quality & Reliability Management,* Vol 11, p. 6-21.

Hauck D., Runger G., and Montgomery D. (1999). "Multivariate statistical process monitoring and diagnosis with grouped regression adjusted variables". *Communication in Statistics – Simulation,* Vol. 28, n 2, p. 309-328.

Hawkins D.M. (1993). "Regression adjustment for variables in multivariate quality control". *Journal of Quality Technology,* Vol. 25, p. 170-182.

Hawkins D.M. and Olwell D.H. (1998). *Cumulative Sum Charts and Charting for Quality Improvement.* Springer-Verlag New York, Inc.

Heimann P.A. (1996). “Attribute control charts with large sample sizes”. *Journal of Quality Technology*, Vol. 28, n 4, p. 451-459.

Heredia-Langner H.A., Montgomery D.C., and Carlyle W.M. (2002). “Solving a multistage partial inspection problem using genetic algorithms”. *International Journal of Production Research*, Vol. 40, n 8, p. 1923–1940.

Hutwagner L.C., Thompson W.W., Seeman G.M., and Treadwell T. (2005). “A simulation model for assessing aberration detection methods used in public health surveillance for systems with limited baselines”. *Statistics in Medicine*, Vol. 24, p. 543–550.

Jackson J.E. (1972). "All count distributions are not alike". *Journal of Quality Technology*, 4, p. 86-92.

Jearkpaporn D., Montgomery D.C., Runger G.C., and Borror C.M. (2005). “Model-based process monitoring using robust generalized linear models”. *International Journal of Production Research*, Vol. 43, n 7, April, p. 1337–1354.

Jensen J.B. and Markland, R.E. (1996). “Improving the application of quality conformance tools in service firms”. *Journal of Services Marketing*, Vol. 10, n 1, p. 35-55.

Johnson N.L., Kemp A.W., and Kotz S. (2005). “*Univariate Discrete Distributions*”. 3rd Edition. John Wiley & Sons, Inc.

Johnson N.L., Kotz S., and Balakrishnan, N. (1996) *Discrete Multivariate Distribution*, Wiley Series in Probability and Statistics, John Wiley & Sons, Inc.

Kang L. and Albin S. (2000). “On-line monitoring when the process yields a linear profile”. *Journal of Quality Technology*, Vol. 32, n 4, p. 418-426.

Kaplan S. (1982). “Matrix theory formalism for event tree analysis: application to nuclear-risk analysis”. *Risk Analysis*, Vol. 2, n 1, p. 9-18.

Kaya I. and Engin O. (2007). “A new approach to define sample size at attributes control chart in multistage processes: An application in engine piston manufacturing process”. *Journal of Materials Processing Technology*, Vol. 183, p. 38–48

Kemp C.D. and Kemp A.W. (1987). “Rapid generation of frequency tables”. *Applied Statistics*, Vol. 36, n 3, p. 277-282.

Kendall M.G. and Gibbons J.D. (1990). *Rank Correlation Methods*, 5th Edition, New York, Oxford University Press.

Laney D.B. (2002). “Improved control charts for attributes”. *Quality Engineering*, Vol. 14, n 4, p. 531–537.

Laviolette M. (1995). “Bayesian monitoring of multinomial processes”. *The Journal of the Industrial Mathematics Society*, Vol. 45, p41-49.

Lawless J.F., Mackay R.J., and Robinson J.A. (1999). “Analysis of variation transmission in manufacturing processes—Part I”. *Journal of Quality Technology*, Vol. 31, n 2, p. 131–142.

Lee J.H., Dorsey A.W., and Russell S. (2004). “Inferential product quality control of a multistage batch plant”. *Process Systems Engineering*, Vol. 50, n 1, p. 136-148.

Liu J.Y., Xie M., and Goh T. N. (2006). “CUSUM chart with transformed exponential data”. *Communications in Statistics Theory and Methods*, Vol. 35, n 10, p. 1829-1843.

Liu J. Y., Xie M., Goh T. N., and Lai C. D. (2007). "A study of EWMA chart with transformed exponential data". *International Journal of Production Research*, Vol. 45, n 3, p. 743-763.

Loredo E., Jearkpaporn D., and Borror C. (2002). "Model-based control chart for autoregressive and correlated data". *Quality and Reliability Engineering International,* Vol. 18, p. 489–496

Lowry C.A., Woodall W.H., Champ C.W., and Rigdon S.E. (1992). "A multivariate exponentially weighted moving average control chart". *Technometrics*, Vol. 34, n 1, p. 46-53.

Lucas J.M., Davis D.J. and Saniga E.M. (2006). "Detecting improvement using Shewhart attribute control charts when the lower control limit is zero". *IIE Transactions*, Vol. 38, n 8, p. 659-666.

MacCarthy B.L. and Wasusri T. (2002). "A review of non-standard applications of statistical process control (SPC) charts". *International Journal of Quality & Reliability Management,* Vol. 19, n 3, p. 295-320.

Mandelbaum A., Sakov A., Shen H., and Zeltyn S. (2001). "Empirical analysis of a call center". Technion (Israel Institute of Technology). www.ie.technion.ac.il/serveng/References.

Marcucci M. (1985). "Monitoring multinomial processes". *Journal of Quality Technology*. Vol. 17, April, n 2, p. 86-91.

Montgomery D. C. (2005). *Introduction to Statistical Quality Control*. 5th Edition, John Wiley & Sons, Inc., New York, NY.

Montgomery D.C. and Mastrangelo C.M. (1991). "Some statistical process control methods for autocorrelated data" (with discussion). *Journal of Quality Technology*, Vol. 23, n 3, p. 179-204.

Montgomery D.C. and Runger G.C. (2002). *Applied Statistics and Probability for Engineers.* 3rd Edition. John Wiley & Sons, Inc., New York, NY.

Niaki S. and Davoodi M. (2009). "Designing a multivariate-multistage quality control system using artificial neural networks". *International Journal of Production Research,* Vol. 47, n 1, p. 251-271.

Perry M.B. and Pignatiello J.J. (2005). "Estimation of the change point of the process fraction nonconforming in SPC applications". *International Journal of Reliability, Quality and Safety Engineering*, Vol. 12, no 2, p. 95-110.

Perry M.B., Pignatiello J.J., and Simpson J.R. (2007). "Estimating the change point of the process fraction nonconforming with a monotonic change disturbance in SPC". *Quality and Reliability Engineering International*, Vol. 23, n 3, p. 327-339.

Pettersson M. (2004). "SPC with applications to churn management". *Quality and Reliability Engineering International*, Vol. 20, n 5, p. 397-406.

Prabhu S. and Runger G. (1997). "Designing a multivariate EWMA control chart". *Journal of Quality Technology*, Vol. 29, p. 8-15.

Property Tax Assessment Study Commission (1986). *Report of the Property Tax Assessment Study Commission, State of New Jersey*. The Commission.

Quesenberry C.P. (1991). "SPC *Q* charts for a binomial parameter *p:* short or long runs". *Journal of Quality Technology,* Vol. 23, n 3, p. 239-246.

Quesenberry C.P. (1995). "On properties of binomial *Q* charts for attributes". *Journal of Quality Technology,* Vol. 27, n 3, p. 204-213.

Rafool M. (2002). *A Guide to Property Taxes: An Overview*. NCSL Fiscal Affairs Program. National Conference of State Legislatures (www.ncsl.org).

Reynolds M.R. and Stoumbos Z.G (1999). "A CUSUM chart for monitoring a proportion when inspecting continuously". *Journal of Quality Technology,* Vol. 31, n 1, p. 87-108.

Reynolds M.R. and Stoumbos Z.G (2000). "A general approach to modeling CUSUM charts for a proportion". *IIE Transactions,* Vol. 32, n 6, p. 515-535.

Rogerson P.A. (2006). "Formulas for the design of CUSUM quality control charts". *Communications in Statistics Theory and Methods*, Vol. 35, n 2, p. 373-383.

Ryan T.P. and Schwertman N.C. (1997). "Optimal limits for attributes control charts". *Journal of Quality Technology,* Vol. 29, n 1, p. 86-98.

Schader M. and Schmid F. (1989). "Two rules of thumb for the approximation of the binomial distribution by the normal distribution". *The American Statistician*, Vol. 43, n 1, February, p. 23-24.

Schwertman N.C. (2005). "Designing accurate control charts based on the geometric and negative binomial distributions". *Quality and Reliability Engineering International*, Vol. 21, n 8, p. 743-756.

Schwertman N.C. and Ryan T.P (1999). "Using dual *np*-charts to detect changes". *Quality and Reliability Engineering International*, Vol. 15, p. 317-320.

Shiau J.H., Chen C., and Feltz C.J. (2005). "An empirical Bayes process monitoring technique for polytomous data". *Quality and Reliability Engineering International,* Vol. 21, p13–28.

Shore H. (2000). "General control charts for attributes". *IIE Transactions,* Vol. 32, p. 1149-1160.

Shore H. (2006). "Control charts for the queue length in a G/G/S system". *IIE Transactions*, Vol. 38, n 12, Dec, p. 1117-1130.

Shu L., Tsung F., and Tsui K.L. (2005). "Effects of estimation errors on cause-selecting charts". *IIE Transactions,* Vol. 37, p. 559-567.

Shu L., Tsung F., and Tsui K.L. (2004). "Run-length performance of regression control charts with estimated parameters". *Journal of Quality Technology,* Vol. 36, n 3, p. 280-292.

Simonoff J. S. (2003). *Analyzing Categorical Data.* Springer texts in statistics. Springer-Verlag New York, Inc.

Skinner K.R., Montgomery D.C., and Runger G.C. (2004). "Generalized linear model-based control charts for discrete semiconductor process data". *Quality and Reliability Engineering International.* Vol. 20, p. 777–786.

Skinner K.R., Runger G.C., and Montgomery D.C. (2006). "Process monitoring for multiple count data using a deleted-Y statistic". *Quality Technology and Quantitative Management.* Vol. 3, p. 247–262.

Skinner K.R., Runger G.C., and Montgomery D.C. (2003). "Process-monitoring for multiple count data using generalized linear model-based control charts". *International Journal of Production Research*, Vol. 41, p. 1167-1180.

Spanos C.J. and Chen R.L. (1997). "Using qualitative observations for process tuning and control". *IEEE Transactions on Semiconductor Manufacturing*, Vol. 10, n 2, p307-316.

Steiner S.H., Cook R. J**.,** and Farewell V.T. (1999). "Monitoring paired binary surgical outcomes using cumulative sum charts". *Statistics* in Medicine, Vol. 18, p. 69-86.

Steiner S.H., Cook R. J**.,** Farewell V.T., and Treasure T. (2000). "Monitoring surgical performance using risk-adjusted cumulative sum charts". *Biostatistics,* Vol. 1, p. 441-452.

Steiner S.H. and Mackay, R.J. (2000). "Monitoring processes with highly censored data". *Journal of Quality Technology*, Vol. 32, n 3, p. 199-208, July.

Sulek J.M. (2004). "Statistical quality control in services". *International Journal of Services Technology and Management,* Vol. 5, n 5 & 6, p. 522-531.

Sulek J.M., Marucheck A., and Lind M.R. (2006). "Measuring performance in multi-stage service operations: an application of cause selecting control charts". *Journal of Operations Management*, Vol. 24, n 5, p. 711-727.

Tucker G. R., Woodall W.H., and Tsui K-L. (2002). "A control chart for ordinal data". *American Journal of Mathematical and Management Sciences,* Vol. 22, n 1 & 2, p31-48.

Wade M.R. and Woodall W.H. (1993). "A review and analysis of cause selecting control charts". *Journal of Quality Technology*, Vol. 25, p. 161–169.

Woodall W.H. (2006). "The use of control charts in health-care and public-health surveillance". *Journal of Quality Technology,* Vol. 38, n 2, p. 89-104.

Woodall W.H. (1997). "Control charts based on attribute data: bibliography and review". *Journal of Quality Technology*, Vol. 29, n 2, p. 172-183.

Woodall W.H. and Adams B.M. (1993). "The statistical design of CUSUM charts". *Quality Engineering*, Vol. 5, n 4, p. 559-570.

Woodall W., Spitzner D., Montgomery D., and Gupta S. (2004). "Using control charts to monitor process and product quality profiles". *Journal of Quality Technology*, Vol. 36, n 3, p. 309-320.

Wu Z. and Jiao J. (2007). "Evaluating and improving the unit and group-runs chart". *Journal of Quality Technology*, Vol. 39, n 4, p. 355-363

Wu Z., Luo H., and Zhang X. (2006). "Optimal np control chart with curtailment" *European Journal of Operational Research*, Vol. 174, p.1723–1741.

Wu Z., Zhang X., Yeo S.H., and Chen Z. (2001). "Fractional control limits for *np* control chart". *Process Control & Quality,* Vol. 11, n 6, p. 491-501.

Xie M., Goh T.N., and Tang X.Y. (2000). "Data transformation for geometrically distributed quality characteristics". *Quality and Reliability Engineering International*, n 16, p. 9-15.

Yao D.D. and Zheng S. (1999). "Coordinated quality control in a two-stage system". *IEEE Transactions on Automatic Control*, Vol. 44, n 6, p. 1166-1179.

Yarnold J.K. (1970). "The minimum expectation in χ^2 goodness of fit tests and the accuracy of approximations for the null distribution". *Journal of the American Statistical Association,* Vol. 65, p864-886.

Yashchin E. (1993). "Statistical control schemes: methods, applications, and generalizations". *International Statistical Review*, Vol. 61, n 1, p. 41-66.

Zantek P.F., Wright G.P, and Plante R.D. (2006). "A self-starting procedure for monitoring process quality in multistage manufacturing systems". *IIE Transactions*, Vol. 38, p. 293–308.

Zhou S., Ding Y., Chen Y., and Shi J. (2003). "Diagnosability study of multistage manufacturing processes based on linear mixed-effects models". *Technometrics*, Vol. 45, n 4, p. 312–325.

Zou C. and Tsung F. (2008). "Directional MEWMA schemes for multistage process monitoring and diagnosis". *Journal of Quality Technology,* Vol. 40, n 4, October, p. 407-427.

9 Appendices

Appendix A. Articles about monitoring single stage processes with multiple categories

(Summary in Table 2.1)

Laviolette (1995)

In this paper, a Bayesian monitoring system is proposed to monitor all the fractions but one of items falling in different quality categories. The categories are sorted from poorest to best quality. Only the fraction related to the best quality is not monitored.

The count variables are modeled by a multinomial distribution for known probability parameters. The Bayesian approach allows the probabilities parameters to vary according to a prior probability distribution, which is typically the Dirichlet distribution.

A probability control limit is obtained for the posterior cumulative distribution of the multinomial probability parameters, for all but the parameter related to the best quality. If this posterior probability is less than a type I error α, the process is considered out-of-control. The author refers to this chart as the Dirichlet *p*-chart.

An example is given for monitoring the multinomial process with 3 categories found in Marcucci (1985). This Dirichlet *p*-chart monitors the posterior cumulative distribution of the two nonconforming probability parameters, for an arbitrary prior distribution, showing better sensitivity than Marcucci's method (1985). Laviolette suggests monitoring individual probability parameters with marginal distributions of the posterior cumulative distribution in order to solve for the interpretation of out-of-control points.

Marcucci (1985)

Marcucci gives data with variable sample size, where bricks are classified into conforming, nonconforming type *A* and nonconforming type *B* categories with in-control probabilities 0.95, 0.03, and 0.02 respectively. He proposes a monitoring system based on a multinomial model, using a Pearson statistic (also known as chi-square statistic) that follows an approximate chi-square distribution. While Marcucci's method is simple to use, it is difficult to interpret an out-of-control signal. Marcucci's work is still the most accepted procedure to monitor nominal categorical data for uniattribute processes as recalled in Tucker *et al.* (2002).

In general, if the in-control probabilities that a variable is classified in a category out of a total of *K* categories are known, then the chi-square statistic that is monitored at time t is,

$$X_t^2 = \sum_{i=1}^{K} \frac{(n_{ti} - n_t p_i)^2}{n_t p_i} \sim \chi^2_{(K-1)} \text{ pdf} \qquad \text{(A-1)}$$

Where,

n_{ti} = number of occurrences of attribute *i* in a sample of *n* independent trials at time *t*

n_t = sample size at time *t*

p_i = probability of occurrences of attribute *i* in a trial; $i=1,2,\ldots, K$

The following relations hold: $\sum_{j=1}^{K} n_{tj} = n_t$ $\sum_{i=1}^{K} p_j = 1$

When the multinomial process is in-control, the Pearson statistic approximately distributes chi-square with *K*-1 degrees of freedom. Let us call the above method the Marcucci mathod. The Pearson approximation might be acceptable if:

i. No more than 20% of the expected frequencies (sample size multiplied by probability of a category) are less than five (Cochran 1954). A less restrictive rule by Yarnold (1970): let K be the number of categories, and let r be the number of expected frequencies less than five. For $K \geq 3$, the minimum expected frequency should be at least $5K/r$.

ii. A sample size n of at least 167 observations, based on Yarnold (1970).

Changes of both positive and negative signs in the quality proportions lead to increases in the Pearson. When the process is out-of-control, this statistic is asymptotically non-central chi-square distributed. The non-centrality parameter measures deviations from baseline known proportions.

The Marcucci method signals when any fraction departs from its in-control value. If only one or two fractions are of special interest, Marcucci proposes a bivariate control chart for monitoring the chosen two nonconforming fractions. The author suggests monitoring for the proportions associated with the major and the minor nonconformity. In fact, these bivariate p-charts serve partially to solve for the interpretation problem. He calls this procedure as “One-Sided Generalized p-Charts”, which is intended to address trinomial processes.

In case that the probabilities of the attributes are unknown, Marcucci presents the Pearson-Duncan statistic to be monitored (Z_t). This statistic is used to test homogeneity of proportions between the base period (time 0) and each monitoring period (time t). When in-control, the statistic distributes asymptotically chi-squared with (K-1) degrees of freedom.

$$Z_t^2 = n_t \cdot n_o \sum_{i=1}^{K} \frac{\left(\frac{n_{ti}}{n_t} - \frac{n_{0i}}{n_0}\right)^2}{n_{ti} + n_{0i}} \qquad \text{(A-2)}$$

Johnson *et al.* (1996) provides better approximations for the probability function, expectation, and variance of the Pearson statistic (p. 45-47). These approximations might be useful to improve the sensitivity of a (further modified) Marcucci control chart, particularly when the chi-square approximation assumptions are not well fulfilled.

Shiau *et al.* (2005)

In this paper, a Bayesian method is proposed to monitor the fractions of items falling in multiple categories fail modes plus one category of pass. Unlike Laviolette (1995), Shiau *et al.* method focuses on fractions of every category in the process (Laviolette focuses only on nonconforming categories).

The Bayesian approach allows the probabilities parameters of the multinomial process to vary according to a prior probability distribution, which is typically the Dirichlet distribution. The expectation of a prior probability parameter (p_i) is denoted as α_i.

In Bayesian terminology, the Dirichlet distribution is the conjugate prior distribution for the multinomial model. Being a conjugate prior implies that the posterior distribution for the probability parameters given a data set has the same distribution than the prior distribution. Thus, the posterior distribution is also a Dirichlet one. The posterior expected values of the probability parameters - probability vector - are a linear weighted average of the α_i's and of the observed fractions.

The paper shows that the distribution of the counts is a Dirichlet-compound multinomial, also known as the Polya-Eggenberger distribution as seeing in Johnson *et al.* (1996). Instead of proposing control regions for this multivariate discrete distribution like Laviolette (1995), the paper considers monitoring every individual fraction of items falling in different categories. The previous work of Laviolette (1995) is not mentioned by the authors. Marginal distributions are described for every fraction plotted and upper and lower randomized control limits are determined. The setting of the individual false alarm rates is not further discussed as explicitly mentioned in page 20. In an example, a process with 5 categories goes out of control in the prior probability of the *good* category, and in the prior probability of the fourth *bad* category, keeping constant the other three prior probabilities.

The control chart about the fourth fraction shows that the process is indeed out-of-control. The control chart about the third fraction does not show out of control points. This paper does not approach the issue of correlation among individual fractions, so for example it can not always be concluded that the process is in-control if all charts do not signal. As shown in Chapter 3, the *p-tree* method has better diagnosis accuracy and sensitivity than the Shiau *et al.* (2005) method. Thus, the Shiau *et al.* (2005) method should be used only to detect whether the process is in-control or not, which is the intention of the authors. It should not be used for interpretations, because its univariate charts are negatively correlated, so its diagnosis accuracy is low. The inadequacy of using univariate charts for correlated variables is also noted in Montgomery (2005, p. 487-488 and p. 499) and Lowry *et al.* (1992, p. 52).

Spanos and Chen (1997)

The authors present a plasma etching problem in semiconductor manufacturing. The objective of the etching process is to create silicon lines that match a target pattern as close as possible. The process settings are continuous covariates like temperature, etch time, power, pressure, and oxygen flow rate inside the reactor. The output is multivariate in nature. Some output variables are categorical ordinal data such as the sidewall roughness, and the presence of indentations in line profile, which are called mouse bites. Roughness can take the values: smooth, fair, rough, and roughest. Mouse bites can take the values: good, fair, poor, and worst.

Spanos and Chen set a model just for the mouse bites based on logistic regression, also called logit model. The model fit comes very significant, and only the covariates power and pressure are selected to stay in the final model. Thus, the logs of cumulative probabilities are fit as a linear function of the controllable process inputs. Estimating the probability of every category is straightforward after predicting the cumulative probabilities. The paper proposes two methods to monitor for deviations in the mouse bites fractions corresponding to each category (*good, fair, poor,* and *worst*): one method for short term monitoring, and another one for long term monitoring.

Short term monitoring: The objective of the short term monitoring method is to monitor for abrupt process changes during production. Spanos and Chen propose sequential run rules. A weakness of this method is that monitors only for the fraction of wafers being in one category. For example, monitoring for the fraction of the category *good* does not depend on the other individual three fractions (for categories *fair, poor,* and *worst*). The paper proposes that in case of an optimized process regarding mouse

bites, the fraction associated to the category *worst* should be monitored, because that category will be less likely to appear.

Long term monitoring: The objective of the long term monitoring method is to control for permanent process shifts that might happen due to natural process aging. For this case, the authors propose to monitor with a Pearson statistic similar with Marcucci (1985).

After adjusting, there is also the problem of what fraction is selected to monitor using the short term monitoring method. An automated scheme that integrates the short term and long term monitoring systems, the model estimation and update, and a feedback control (process settings) is proposed.

Tucker *et al.* (2002)

Tucker *et al.* propose a maximum likelihood based chart to monitor for ordinal categorical data when the underlying quality follows some unobservable and unknown distribution. A finished bricks example is provided where nonconforming bricks type *B* are worst that nonconforming bricks type *A*, and the probability of nonconforming is 0.95, the probability of nonconforming bricks type *A* is 0.03, and the probability of nonconforming bricks type *B* is 0.02.

Assume that the probability distribution contains a location parameter θ. A maximum likelihood estimate (MLE) procedure is used to find an estimate $\hat{\theta}$. The statistic that is monitored in a Shewhart control chart corresponds to:

$$\hat{\theta} \Big/ STD(\hat{\theta}) \qquad \text{(A-3)}$$

Where, STD is the standard deviation. The estimate $\hat{\theta}$ distributes standard normal if the process is in-control.

The maximum likelihood based chart of Tucker *et al.* is relevant when the user may guess well about the underlying quality distribution. However the authors point out that no amount of historical ordinal data will reveal the shape of the underlying distribution, even if chopping the scale in more intervals. If ordinal data is simulated assuming an underlying distribution (like the normal or exponential distribution), then the sensitivity of the proposed ordinal method is better that the chi-square control chart for detecting quality improvement but not necessarily for detecting quality deterioration. No interpretation of signals is provided.

Appendix B. The fractions $\hat{f}_i$ are unbiased estimates of f_i

(Complimentary material)

Proof

By definition: $E[\hat{f}_1] = E[\frac{n_1}{N}] = f_1$

Consider,

$$E[\hat{f}_i] = E\left[\left(\frac{n_i}{N-\sum_{j=1}^{i-1} nj}\right)\right] = \sum_{m=0}^{N} E\left[\frac{n_i}{N-\sum_{j=1}^{i-1} n_j}\middle/\sum_{j=1}^{i-1} n_j = m\right]\cdot \Pr\{\sum_{j=1}^{i-1} n_j = m\} \quad , \; i=2,\ldots,K\text{-}1$$

$$= \sum_{m=0}^{N} E\left[\frac{n_i}{N-m}\middle/\sum_{j=1}^{i-1} n_j = m\right]\cdot \Pr\{\sum_{j=1}^{i-1} n_j = m\} = \sum_{m=0}^{N} \frac{1}{N-m} E\left[n_i\middle/\sum_{j=1}^{i-1} n_j = m\right]\cdot \Pr\{\sum_{j=1}^{i-1} n_j = m\}$$

Using Johnson *et al*. (1996, p. 35), $n_i \Big/ \sum_{j=1}^{i-1} n_j \sim$ Binomial $(N-\sum_{j=1}^{i-1} n_j, f_i)$

Therefore,

$$E[\hat{f}_i] = \sum_{m=0}^{N} \frac{1}{(N-m)}\cdot(N-m)\cdot f_i \cdot \Pr\{\sum_{j=1}^{i-1} n_j = m\} = f_i \cdot \sum_{m=0}^{N} \Pr\{\sum_{j=1}^{i-1} n_j = m\} = f_i \cdot 1 = f_i \qquad \text{(B-1)}$$

Proved.

Appendix C. Contiguous tree fractions are uncorrelated

(Complimentary material)

Let start with the proof that $\hat{f}_1$ and $\hat{f}_2$ are uncorrelated

$$\mathrm{Cov}[\hat{f}_1, \hat{f}_2] = \mathrm{Cov}[\frac{n_1}{N}, \frac{n_2}{N-n_1}] = \mathrm{Cov}[1-\frac{N-n_1}{N}, \frac{n_2}{N-n_1}] = -\mathrm{Cov}[\frac{N-n_1}{n}, \frac{n_2}{N-n_1}]$$

$$= -\mathrm{E}[\frac{(N-n_1)}{N} \cdot \frac{n_2}{(N-n_1)}] + \mathrm{E}[\frac{(N-n_1)}{N}] \cdot \mathrm{E}[\frac{n_2}{N-n_1}]$$

$$= -\mathrm{E}[\frac{n_2}{N}] + \mathrm{E}[1-\frac{n_1}{N}] \cdot \mathrm{E}[\frac{n_2}{N-n_1}] = -p_2 + (1-p_1) \cdot \mathrm{E}[\hat{f}_2]$$

$$= -p_2 + (1-p_1) \cdot \frac{p_2}{1-p_1} = 0. \quad \text{Proved.}$$

Let prove now that in general the tree fractions $\hat{f}_j$ and $\hat{f}_{j+1}$ are uncorrelated random variables, for any $K \geq 3$.

Proof

Consider,

$$\mathrm{Cov}[\hat{f}_j, \hat{f}_{j+1}] = \mathrm{Cov}[\frac{n_j}{N-\sum_{i=1}^{j-1} ni}, \hat{f}_{j+1}] = \mathrm{Cov}[\frac{n_j}{N-\sum_{i=1}^{j-1} ni} - 1 + 1, \hat{f}_{j+1}] \qquad j=2,\ldots,K\text{-}1$$

$$= \mathrm{Cov}[\frac{n_j}{N-\sum_{i=1}^{j-1} n_i} - \frac{N-\sum_{i=1}^{j-1} n_i}{N-\sum_{i=1}^{j-1} n_i} + 1, \hat{f}_{j+1}] = \mathrm{Cov}[\frac{n_j - N + \sum_{i=1}^{j-1} n_i}{N-\sum_{i=1}^{j-1} n_i} + 1, \hat{f}_{j+1}]$$

$$= \mathrm{Cov}[\frac{\sum_{i=1}^{j} n_i - N}{N-\sum_{i=1}^{j-1} n_i} + 1, \hat{f}_{j+1}] = \mathrm{Cov}[-\frac{N-\sum_{i=1}^{j} n_i}{N-\sum_{i=1}^{j-1} n_i}, \hat{f}_{j+1}]$$

Using the definition of $\hat{f}_{j+1}$ (see Section 3.1):

$$= \mathrm{Cov}\left[-\frac{N-\sum_{i=1}^{j} n_i}{N-\sum_{i=1}^{j-1} n_i} \;,\; \frac{n_{j+1}}{N-\sum_{i=1}^{j} n_i} \right] = -\,\mathrm{Cov}\left[\frac{N-\sum_{i=1}^{j} n_i}{N-\sum_{i=1}^{j-1} n_i} \;,\; \frac{n_{j+1}}{N-\sum_{i=1}^{j} n_i} \right]$$

$$= -\,\mathrm{E}\left[\left(\frac{N-\sum_{i=1}^{j} n_i}{N-\sum_{i=1}^{j-1} n_i} \right) \cdot \left(\frac{n_{j+1}}{N-\sum_{i=1}^{j} n_i} \right) \right] + \mathrm{E}\left[\left(\frac{N-\sum_{i=1}^{j} n_i}{N-\sum_{i=1}^{j-1} n_i} \right) \right] \cdot \mathrm{E}\left[\hat{f}_{j+1} \right]$$

Simplifying the first term,

$$= -\,\mathrm{E}\left[\frac{n_{j+1}}{N-\sum_{i=1}^{j-1} n_i} \right] + \mathrm{E}\left[\left(\frac{N-\sum_{i=1}^{j} n_i}{N-\sum_{i=1}^{j-1} n_i} \right) \right] \cdot \mathrm{E}\left[\hat{f}_{j+1} \right]$$

Let work now on the first expectation that appears above,

$$\mathrm{E}\left[\frac{n_{j+1}}{N-\sum_{i=1}^{j-1} n_i} \right] = \sum_{m=0}^{n} E\left[\frac{n_{j+1}}{N-\sum_{i=1}^{j-1} n_i} \middle/ \sum_{i=1}^{j-1} n_i = m \right] \cdot \Pr\left\{ \sum_{i=1}^{j-1} n_i = m \right\} =$$

$$= \sum_{m=0}^{n} E\left[\frac{n_{j+1}}{N-m} \middle/ \sum_{i=1}^{j-1} n_i = m \right] \cdot \Pr\left\{ \sum_{i=1}^{j-1} n_i = m \right\}$$

$$= \sum_{m=0}^{n} \frac{1}{N-m} E\left[n_{j+1} \middle/ \sum_{i=1}^{j-1} n_i = m \right] \cdot \Pr\left\{ \sum_{i=1}^{j-1} n_i = m \right\}$$

Using the conditional distribution of n_{j+1} (see Section 3.2):

$$= \sum_{m=0}^{n} \frac{1}{(N-m)} \cdot \frac{(N-m)\cdot(p_{j+1})}{1-\sum_{l=1}^{j-1} p_l} \cdot \Pr\left\{ \sum_{i=1}^{j-1} n_i = m \right\}$$

$$= \frac{p_{j+1}}{1-\sum_{l=1}^{j-1} p_l} \cdot \sum_{m=0}^{n} \Pr\left\{ \sum_{i=1}^{j-1} n_i = m \right\} = \frac{p_{j+1}}{1-\sum_{l=1}^{j-1} p_l} \cdot 1 = \frac{p_{j+1}}{1-\sum_{l=1}^{j-1} p_l}$$

With this result, Cov[$\hat{f}_j$, $\hat{f}_{j+1}$] becomes,

$$\text{Cov}[\hat{f}_j, \hat{f}_{j+1}] = -\frac{p_{j+1}}{1-\sum_{l=1}^{j-1} p_l} + \text{E}\left[\frac{N-\sum_{i=1}^{j} n_i}{N-\sum_{i=1}^{j-1} n_i}\right] \cdot \text{E}[\hat{f}_{j+1}] = -\frac{p_{j+1}}{1-\sum_{l=1}^{j-1} p_l} + \text{E}\left[\frac{N-\sum_{i=1}^{j} n_i}{N-\sum_{i=1}^{j-1} n_i}\right] \cdot \left(\frac{p_{j+1}}{1-\sum_{l=1}^{j} p_l}\right)$$

$$= -\frac{p_{j+1}}{1-\sum_{l=1}^{j-1} p_l} + \text{E}\left[\frac{N-\sum_{i=1}^{j} n_i}{N-\sum_{i=1}^{j-1} n_i}\right] \cdot \left(\frac{p_{j+1}}{1-\sum_{l=1}^{j} p_l}\right)$$

Let work on the remaining expectation that appears above,

$$\text{E}\left[\frac{N-\sum_{i=1}^{j} n_i}{N-\sum_{i=1}^{j-1} n_i}\right] = \sum_{l=0}^{n} E\left[\frac{N-\sum_{i=1}^{j} n_i}{N-\sum_{i=1}^{j-1} n_i} \middle/ \sum_{i=1}^{j-1} n_i = l\right] \cdot \Pr\{\sum_{i=1}^{j-1} n_i = l\}$$

$$= \sum_{l=0}^{n} E\left[\frac{N-\sum_{i=1}^{j} n_i}{N-l} \middle/ \sum_{i=1}^{j-1} n_i = l\right] \cdot \Pr\{\sum_{i=1}^{j-1} n_i = l\}$$

$$= \sum_{l=0}^{n} \frac{1}{N-l} E\left[n-\sum_{i=1}^{j} n_i \middle/ \sum_{i=1}^{j-1} n_i = l\right] \cdot \Pr\{\sum_{i=1}^{j-1} n_i = l\}$$

$$= \sum_{l=0}^{n} \frac{1}{N-l} E\left[n-\sum_{i=1}^{j-1} n_i - n_j \middle/ \sum_{i=1}^{j-1} n_i = l\right] \cdot \Pr\{\sum_{i=1}^{j-1} n_i = l\}$$

$$= \sum_{l=0}^{n} \frac{1}{N-l} E\left[N-l-n_j \middle/ \sum_{i=1}^{j-1} n_i = l\right] \cdot \Pr\{\sum_{i=1}^{j-1} n_i = l\}$$

$$= \sum_{l=0}^{n} \frac{1}{N-l} \left(N-l-E\left[n_j \middle/ \sum_{i=1}^{j-1} n_i = l\right]\right) \cdot \Pr\{\sum_{i=1}^{j-1} n_i = l\}$$

$$= \sum_{l=0}^{n} \frac{1}{N-l} \left(N-l-\frac{(N-l)\cdot p_j}{1-\sum_{l=1}^{j-1} p_l}\right]\right) \cdot \Pr\{\sum_{i=1}^{j-1} n_i = l\}$$

$$= (1-\frac{p_j}{1-\sum_{l=1}^{j-1} p_l}) \cdot \sum_{l=0}^{n} \Pr\{\sum_{i=1}^{j-1} n_i = l\} = (1-\frac{p_j}{1-\sum_{l=1}^{j-1} p_l}) \cdot 1 = 1-\frac{p_j}{1-\sum_{l=1}^{j-1} p_l}$$

With this result, Cov[$\hat{f}_j$, $\hat{f}_{j+1}$] becomes,

$$\text{Cov}[\hat{f}_j, \hat{f}_{j+1}] = -\frac{p_{j+1}}{1-\sum_{l=1}^{j-1} p_l} + (1-\frac{p_j}{1-\sum_{l=1}^{j-1} p_l}) \cdot (\frac{p_{j+1}}{1-\sum_{l=1}^{j} p_l})$$

$$= -\frac{p_{j+1}}{1-\sum_{l=1}^{j-1} p_l} + (\frac{1-p_j-\sum_{l=1}^{j-1} p_l}{1-\sum_{l=1}^{j-1} p_l}) \cdot (\frac{p_{j+1}}{1-\sum_{l=1}^{j} p_l})$$

$$= -\frac{p_{j+1}}{1-\sum_{l=1}^{j-1} p_l} + (\frac{1-\sum_{l=1}^{j} p_l}{1-\sum_{l=1}^{j-1} p_l}) \cdot (\frac{p_{j+1}}{1-\sum_{l=1}^{j} p_l})$$

$$= -\frac{p_{j+1}}{1-\sum_{l=1}^{j-1} p_l} + \frac{p_{j+1}}{1-\sum_{l=1}^{j-1} p_l} = 0$$

Cov[$\hat{f}_j$, $\hat{f}_{j+1}$] = 0 for j=2,…K-2 Proved.

Appendix D. Normalizing transformations and related Shewhart charts

The Box-Cox, and *Q* transformations for a fraction are described here. Let N_t=sample size or volume of units at sample *t*, p_0=in-control probability of falling in category of interest, *X*=random variable of the number of units in category of interest, and x_t = realized number of units in category of interest at sample *t*.

a) Box-Cox transformation

Let,

bc_t = Box-Cox transformation of fraction f_t

L = power parameter

bc_0 = average of Box-Cox transformation in in-control data

σ_{bc} = standard deviation of Box-Cox transformation in in-control data

The Box-Cox transformation is obtained as $bc_t = \dfrac{\left(\dfrac{x_t}{N_t}\right)^L - 1}{L}$

where L is found through minimizing the skewness of bc_t in the in-control data set (if L=0, the transformation corresponds to the natural logarithm). As an example, Chung *et al*. (2007) develop a monitoring method based on a Box-Cox transformation for a manufacturing application.

b) *Q* transformation (Quesenberry, 1991, 1995)

Let, Q_t = *Q* statistic conditioned on N_t

where, $Q_t = Z_{(1-u_t)}$, $u_t = \Pr\{X \le x_t\}$ that comes from the cumulative Binomial(N_t, p_0) distribution.

The Q_t statistic is approximately normally distributed N(0,1). Quesenberry (1991, p. 63) suggests that this approximation is accurate for Np_0>6.3. In manufacturing applications, this transformation seems to be more accurate regarding the upper tail area than the lower tail area.

c) Shewhart charts for a fraction

Table D.1 summarizes the *p*-chart and other Shewhart charts based on transformations, where z_γ comes from the standard normal distribution such that the upper tail area is γ.

Chart	**Statistic Monitored**	**LCL**	**CL**	**UCL**
p-chart	$\frac{x_t}{N_t}$	$p_0 - Z_{(1-\alpha/2)}\sqrt{\frac{p_0(1-p_0)}{N_t}}$	p_0	$p_0 + Z_{(1-\alpha/2)}\sqrt{\frac{p_0(1-p_0)}{N_t}}$
Arcsine	$2\sqrt{N_t}\left[\sin^{-1}\left(\sqrt{\frac{x_t+3/8}{N_t+3/4}}\right)-\sin^{-1}\left(\sqrt{p_0}\right)\right]$	$- Z_{(1-\alpha/2)}$	0	$Z_{(1-\alpha/2)}$
Box-Cox	bc_t	$bc_0 - Z_{(1-\alpha/2)} \cdot \sigma_{bc}$	bc_0	$bc_0 + Z_{(1-\alpha/2)} \cdot \sigma_{bc}$
Q	Q_t	$- Z_{(1-\alpha/2)}$	0	$Z_{(1-\alpha/2)}$

Table D.1. Shewhart charts for a fraction

Appendix E. Modified *p*-chart in Chen (1998) and modified *np*-chart in Shore (2000)

These two methods are an approximation to the exact binomial method based on probability limits.

a) Modified *p*-chart Chen

This method monitors a fraction with *p*-chart that has control limits based on expansion of quantiles for a fraction. These limits adapted from Chen (1998, eqn. (1)) are the following:

$$UCL = p_0 + Z_{(1-\alpha/2)}\sqrt{\frac{p_0(1-p_0)}{N}} + \frac{(Z_{(1-\alpha/2)}{}^2 - 1)(1-2p_0)}{6N} \qquad \text{(E-1)}$$

$$LCL = p_0 - Z_{(1-\alpha/2)}\sqrt{\frac{p_0(1-p_0)}{N}} + \frac{(Z_{(1-\alpha/2)}{}^2 - 1)(1-2p_0)}{6N}$$

b) Modified *np*-chart Shore

This method monitors x_t, i.e., the number of units in the category of interest with a *np*-chart with control limits corrected by the skewness of the binomial distribution. These limits adapted from Shore (2000, p. 1153) are the following:

$$UCL = Np_0 + Z_{(1-\alpha/2)}\sqrt{Np_0(1-p_0)} + 1.44(1-2p_0)(0.4177 \cdot Z_{(1-\alpha/2)} - \frac{1}{3}) - \frac{1}{2} \qquad \text{(E-2)}$$

$$LCL = Np_0 - Z_{(1-\alpha/2)}\sqrt{Np_0(1-p_0)} - 1.44(1-2p_0)(-0.4177 \cdot Z_{(1-\alpha/2)} + \frac{1}{3}) + \frac{1}{2}$$

Note: here Z_p is based on the upper tail of the standard normal distribution N(0,1) instead of Shore (2000) that uses Z_p as based on the lower tail of N(0,1).

Appendix F. In-control values for tree fraction matrices $\underline{\hat{\mathbf{F}}_j}$

The in-control values for tree the fraction matrices $\underline{\hat{\mathbf{F}}_j}$ are needed to set the CUSUM Arcsine that monitors every independent tree fraction. The method for monitoring MSMC processes is described in Section 5.1.

Let,

$\underline{\mathbf{P}_j}$ = in-control probability matrix from stage j-1 into stage j, for j=1,2,…,M. Matrix $\underline{\mathbf{P}_j}$ has dimension (K_{j-1} x K_j). There are M matrices $\underline{\mathbf{P}_j}$ to represent the transition probabilities in a MSMC process.

$\underline{\mathbf{F}_j}$ = in-control binary probability tree matrix from stage j-1 into stage j, for j=1,2,…,M. Matrix $\underline{\mathbf{F}_j}$ has dimension (K_{j-1} x K_j-1).

The elements of $\underline{\mathbf{P}_j}$ are $\underline{\mathbf{P}_j}[l,m]$ = in-control probability that a customer goes from category l in stage j-1 to category m in stage j, for l=1,2,…,K_{j-1} and m=1,2,…,K_j. The row vector $\underline{\mathbf{P}_j}[l,\cdot]$ represents the probability parameters of a multinomial distribution about going from category l at stage j-1 to any category at stage j. The elements $\underline{\mathbf{P}_j}[l,m]$ are expected values of the realized numbers in each category over its sample size:

$$\underline{\mathbf{P}_j}[l,m] = \begin{cases} \mathrm{E}\left[\dfrac{\underline{\mathbf{S}_j}[l,m]}{\mathrm{MD}(\underline{\mathbf{S}_j}[i,\cdot])}\right] & \text{if } \underline{\mathbf{L}_j}[l,m]=1 \\ 0 & \text{if } \underline{\mathbf{L}_j}[l,m]=0 \end{cases} \qquad \text{(F-1)}$$

Of course, MD($\underline{\mathbf{P}_j}[l,\cdot]$)=1. Also, the probability matrix for going from stage j to stage l equals to $\prod_{i=j}^{l} \underline{\mathbf{P}_j}$, for $j<l$ and has dimension (K_j x K_l).

The elements of $\underline{\mathbf{F_j}}$ are $\underline{\mathbf{F_j}}[l,m]$ = in-control probability that a customer goes from category l at stage j-1 to category m at stage j, given that customer does not go to category 1,…, m-1 at stage j, for m=2,…,K_j.

Similarly to the single stage case developed in Section 3.1, and assuming that the in-control values of $\underline{\mathbf{P_j}}$ are known, the in-control tree probability matrices $\underline{\mathbf{F_j}}$ can be obtained using the total probability rule for multiple events as shown in Montgomery and Runger (2002, p. 44-45) for every row vector $\underline{\mathbf{P_j}}[l,\cdot]$. Thus,

$$\underline{\mathbf{F_j}}[l,m] = \begin{cases} \dfrac{\underline{\mathbf{P_j}}[l,m]}{\sum_{i=1}^{UC_j(l)} \underline{\mathbf{P_j}}[l,i]} & \text{if } 1+UC(l\text{-}1) \le m \le UC(l) \\ 0 & \text{otherwise} \end{cases} \qquad \text{(F-2)}$$

where $UC_{\underline{j}}(i)$ = last category at stage j in which category i of stage j-1 splits into.

$$UC_{\underline{j}}(i) = \sum_{k=1}^{i-1} MD(\underline{\mathbf{L_j}}[k,\cdot])$$

The standard deviation of the in-control tree fractions $\underline{\hat{\mathbf{F}}_{\mathbf{j}}}$ are given by the matrix $\underline{\mathbf{SD_j}}$, with elements that approximately are:

$$\underline{\mathbf{SD}}_{\mathbf{j}}[l,m]=\begin{cases}\sqrt{\dfrac{\underline{\mathbf{F}}_{\mathbf{j}}[l,m](1-\underline{\mathbf{F}}_{\mathbf{j}}[l,m])}{\mathrm{E}[\mathrm{MD}(\underline{\mathbf{S}}_{\mathbf{j}}[l,\cdot])]}} & \text{if } m=UC(l\text{-}1)\\ \sqrt{\dfrac{\underline{\mathbf{F}}_{\mathbf{j}}[l,m](1-\underline{\mathbf{F}}_{\mathbf{j}}[l,m])}{\mathrm{E}[\mathrm{MD}(\underline{\mathbf{S}}_{\mathbf{j}}[l,\cdot])-\sum_{i=1}^{m-1}\mathrm{MD}(\underline{\mathbf{S}}_{\mathbf{j}}[l,i])]}} & \text{if } 2+UC(l\text{-}1)\le m\le UC(l\text{-}1)-1\\ \text{n/a} & \text{otherwise or if } \underline{\mathbf{A}}_{\mathbf{j}}[l,m]=0\end{cases}$$

(F-3)

Note: Equation (F-3) is valid when N is large and the probability that any realized number equals zero is negligible.

Appendix G. Tree fractions in a simple 2-stage process

(Complimentary material)

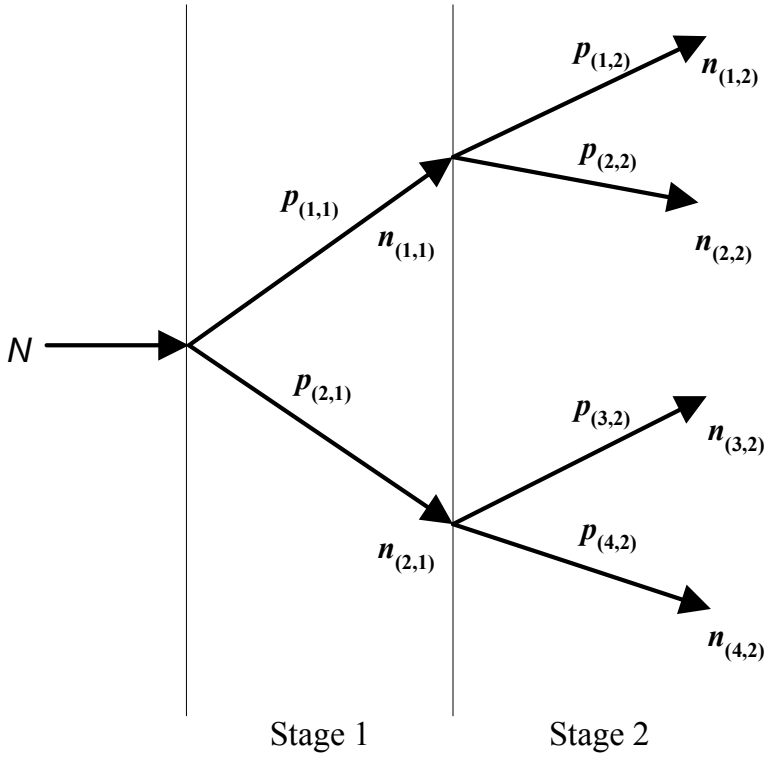

Figure G.1. Multistage process with two stages and four final categories

Consider the 2-stage process shown in Figure G.1 in which numbers under arrows are counts and numbers over arrows are conditional probabilities (elements of matrices $\underline{\mathbf{P}_i}$ as defined in Appendix F). The fraction $\hat{f}_{(1,1)} = \frac{n_{(1,1)}}{N}$ monitors stage 1. Stage 2 is monitored through the fractions $\hat{f}_{(1,2)} = \frac{n_{(1,2)}}{n_{(1,1)}}$ and $\hat{f}_{(3,2)} = \frac{n_{(3,2)}}{n_{(2,1)}}$. The following relations hold: $p_{(1,1)}+p_{(2,1)}=1$, $p_{(1,2)}+p_{(2,2)}=1$, $p_{(3,2)}+p_{(4,2)}=1$, $n_{(1,1)}+n_{(2,1)}==N$, $n_{(1,2)}+n_{(2,2)}= n_{(1,1)}$, and $n_{(3,2)}+n_{(4,2)}=n_{(2,1)}$. Here we show that these tree fractions are independent random variables as summarized in Table G.1.

Tree fractions	$\hat{f}_{(1,1)}$	$\hat{f}_{(1,2)}$	$\hat{f}_{(3,2)}$
$\hat{f}_{(1,1)}$	n/a	Independent	Independent
$\hat{f}_{(1,2)}$	Independent	n/a	Independent
$\hat{f}_{(3,2)}$	Independent	Independent	n/a

Table G.1. Independence of tree fractions for simple 2-stage process

First we show that $\hat{f}_{(1,1)}$ and $\hat{f}_{(1,2)}$ are independent. Consider a multinomial process with three categories formed by $n_{(2,1)}$, $n_{(1,2)}$, and $n_{(2,2)}$ with unconditional probabilities $p_{(2,1)}$, $p_{(1,1)}p_{(1,2)}$, and $p_{(1,1)}p_{(2,2)}$ respectively. According to the Johnson *et al.* (1996, p. 68) decomposition, this multinomial can be expressed as a sequence of independent binomials as follows:

$n_{(2,1)}$ ~ Binomial(N, $p_{(2,1)}$),

$n_{(1,2)}$ given $n_{(2,1)}$ ~ Binomial(N- $n_{(2,1)}$, $\frac{p_{(1,1)} \cdot p_{(1,2)}}{1 - p_{(2,1)}}$).

Thus, the counts divided by their samples sizes, $n_{(2,1)}/N$ and $n_{(1,2)}/(N - n_{(2,1)})$ are independent. The numerator of the first fraction equals N-$n_{(1,1)}$, and the denominator of the second fraction equals $n_{(1,1)}$. So, $(N - n_{(1,1)})/N = 1 - n_{(1,1)}/N$ and $n_{(1,2)}/n_{(1,1)}$ are independent. Necessarily, $\frac{n_{(1,1)}}{N}$ and $n_{(1,2)}/n_{(1,1)}$ are independent. The latter two expressions by definition are $\hat{f}_{(1,1)}$ and $\hat{f}_{(1,2)}$, which are then independent.

Second we show that $\hat{f}_{(1,1)}$ and $\hat{f}_{(3,2)}$ are independent. Consider a multinomial process with three categories formed by $n_{(1,1)}$, $n_{(3,2)}$, and $n_{(4,2)}$ with unconditional probabilities $p_{(1,1)}$, $p_{(2,1)}p_{(3,2)}$, and $p_{(2,1)}\,p_{(4,2)}$ respectively. According to the Johnson *et al.* (1996, p. 68) decomposition, this multinomial can be expressed as a sequence of independent binomials as follows:

$n_{(1,1)} \sim \text{Binomial}(N, p_{(1,1)})$,

$n_{(3,2)}$ given $n_{(1,1)} \sim \text{Binomial}(N\text{-}n_{(1,1)}, \dfrac{p_{(2,1)} \cdot p_{(3,2)}}{1 - p_{(1,1)}})$.

Similarly, $n_{(1,1)}/N$ and $n_{(3,2)}/(N - n_{(1,1)}) = n_{(3,2)}/n_{(2,1)}$ are independent, which using the definitions equal $\hat{f}_{(1,1)}$ and $\hat{f}_{(3,2)}$, and then they are independent.

Finally, we show that $\hat{f}_{(1,2)}$ and $\hat{f}_{(3,2)}$ are independent. Consider a multinomial process with four categories formed by $n_{(1,2)}$, $n_{(2,2)}$, $n_{(3,2)}$, and $n_{(4,2)}$ with unconditional probabilities $p_{(1,1)}p_{(1,2)}$, $p_{(1,1)}p_{(2,2)}$, $p_{(2,1)}p_{(3,2)}$, and $p_{(2,1)}\,p_{(4,2)}$ respectively. According to the Johnson *et al.* (1996, p. 68) decomposition, this multinomial can be expressed as a sequence of independent binomials as follows:

$n_{(1,2)} \sim \text{Binomial}(N, p_{(1,1)}p_{(1,2)})$,

$n_{(2,2)}$ given $n_{(1,2)} \sim \text{Binomial}(N\text{-}\,n_{(1,2)}, \dfrac{p_{(1,1)} \cdot p_{(2,2)}}{1 - p_{(1,1)}\,p_{(1,2)}})$,

$n_{(3,2)}$ given $n_{(1,2)}, n_{(2,2)} \sim \text{Binomial}(N\text{-}n_{(1,2)}\text{-}n_{(2,2)}, \dfrac{p_{(2,1)} \cdot p_{(3,2)}}{1 - p_{(1,1)}\,p_{(1,2)} - p_{(1,1)} \cdot p_{(2,2)}})$.

Similarly, $n_{(1,2)}/N$ and $n_{(3,2)}/(N - n_{(1,2)} - n_{(2,2)}) = n_{(3,2)}/n_{(2,1)}$ are independent. The first fraction is equivalent to $(n_{(1,2)}/n_{(1,1)}) \cdot (n_{(1,1)}/N) = \hat{f}_{(1,2)} \cdot \hat{f}_{(1,1)}$ and the second fraction equals $\hat{f}_{(3,2)}$. Thus,

the product $\hat{f}_{(1,2)} \cdot \hat{f}_{(1,1)}$ and the fraction $\hat{f}_{(3,2)}$ are independent. Recall from the above that $\hat{f}_{(1,1)}$ and $\hat{f}_{(3,2)}$ are independent. Thus, necessarily, $\hat{f}_{(1,2)}$ and $\hat{f}_{(3,2)}$ are independent.

10 Curriculum Vita

RODRIGO I. DURAN LOPEZ
rodrduran@yahoo.com

EDUCATION

PhD Rutgers University, Oct 2009
Industrial and Systems Engineering,
MS Rutgers University, Jan 2007
Statistics,
Option in Quality and Productivity Management
MS Rutgers University, Jan 2003
Industrial and Systems Engineering,
Option in Quality and Reliability Engineering
BS Pontificia Universidad Católica de Chile, 1987
Industrial Engineering and Computer Science,
Diploma in Decentralization and Financial Management, Harvard University, 1996
Harvard Institute for International Development – 4 week international program

WORK EXPERIENCE

Rutgers University, Dept of Industrial and Systems Engineering. Teaching Assistant 2006-2009

Rutgers University, Center for Urban Policy Research, Bloustein School of Public Policy. Research Assistant 2002-2009

Ministry of Finance in Chile, Internal Revenue Service, Assessment Director 1994-2001, Operations Manager 1992-1994

Inual-Tepual, Chile, Automated Control Systems Business Manager 1990-91

CyS Gestion, Chile, Management Consultant 1987-90

PUBLICATIONS

- Paper in print at *IIE Transactions*: "Monitoring and Accurately Interpreting Service Processes with Transactions that are Classified in Multiple Categories". March 2009. Rodrigo I. Duran and Susan L. Albin
- Paper in print at *Quality and Reliability Engineering International*: "Monitoring a Fraction with Simple and Reliable Settings of the False Alarm Rate". Rodrigo I. Duran and Susan L. Albin. Available online as an early view (since 03-20-2009)
- Paper submitted to *Journal of Regional Science*: "On the Effectiveness of Smart Growth Programs in Curbing Urban Sprawl ". February 2009. Rodrigo I. Duran and Michael L. Lahr
- Article "Scholars' Corner". *The Hispanic Outlook in Higher Education*. June 30, Vol. 18, n 19, p. 24