# CHAPTER IV

## Use of Curtailed Single Sampling and Double Sampling Plans in the Economic Design of np-Control Chart

**4.1** In Chapter II the economic model of np-control chart is constructed using more realistic assumptions as compared to the assumptions of Duncan (1956, 1971) as well as knappenberger and Grandage (1969) models. The aim of the construction of our model is to make it more realistic and hence more applicable. The improvement of the model under study over the earlier models is in terms of increasing the applicability of the model.

Another way of improving the economic model is to make it cheaper from cost point of view. In this chapter the economic model is improved from the cost point of view by using curtailed single sampling and double sampling policies in place of the traditional complete single sampling policy. The cost models used for comparing the performance of various sampling policies are (i) the Knappenberger and Grandage's (1969) model (ii) the model developed in chapter II.

In section 4.2 fully-curtailed single sampling is used in place of complete single sampling in the Knappenberger and Grandage's (1969) model. Theoretical as well as numerical comparisons are made between the performance of the complete, the semi-curtailed and the fully-curtailed single sampling policies. The results indicate that the semi-curtailed sampling is no more expensive than the uncurtailed sampling and the fully-curtailed sampling is no more expensive than the semi-curtailed sampling.

In section 4.3 the double sampling policy is used in place of the single sampling policy in the Knappenberger and Grandage's model. It is observed that the double sampling policy is more appropriate than the complete as well as curtailed single sampling policies from the cost point of view.

In section 4.4 the curtailed single sampling policies are used in place of the complete single sampling policy in the single assignable cause model developed by us in chapter II. The conclusions derived are the same as those found for the Knappenberger and Grandage's model.

## 4.2. Use of Fully-Curtailed Sampling Plan in the Economic Design of np-Control Charts.

**4.2.1** Montgomery Heikes and Mance (1975) developed the economic design of np-control chart using the Knappenberger and Grandage's (1969) model for $\bar{x}$-chart. Montgomery et al. developed the expected cost model using the complete sampling plan as a sampling policy. Williams, Looney and Peters (1985) developed the expected cost model using the semi-curtailed sampling plan as a sampling policy. The expected cost model developed by Williams et al. (1985) is analogous in all other respects to the complete sampling model developed by Montgomery et al. (1975). The type of sampling used by Williams et al. corresponds to the sampling plan-2 of Phatak and Bhatt (1967) in which sampling is stopped if either m nonconforming units are observed or n items are inspected. The process is declared to be in control if n items are inspected. The process is declared to be out of control if m nonconforming units are observed.

We develop an expected cost model using the fully-curtailed sampling plan as a sampling policy. The expected cost model developed by us is analogous in all other respects to the complete sampling model developed by Montgomery et al.(1975). The type of curtailed sampling used by us corresponds to the sampling plan-3 of Phatak and Bhatt (1967) in which sampling is stopped if either m nonconforming units are observed or g conforming units are observed. The process is declared to be in the in-control state if g conforming units are observed. The process is declared to be out of control if m nonconforming units are observed.

### 4.2.2 The Production Process and the Sampling Policy

The production process to be controlled starts in the in-control state in which it produces a known acceptable proportion, $p_o$, of nonconforming units. The process as time passes may deteriorate and start producing one of the nonacceptable proportions $p_i$ ($i = 1, 2, \ldots ,s$). Thus there are s out of control states $p_i$($i = 1, 2, \ldots ,s$) where $p_i > p_{i-1}$ ($i = 1, 2, \ldots ,s$). Transition to the out-of-control states is governed by a Poisson Process with assignable causes occurring at a rate $\lambda$ per unit time. This means that the time until the process remains in the in-control state before shifting to an out-of-control state is an exponential random variable. Transition from an out-of-control state $p_i$ to another out-of-control state $p_j$ is possible provided the direction of movement is towards further quality deterioration i.e. transition from $p_i$ to $p_j$ is possible if $p_j > p_i$. The process is not self correcting. Once a shift to out-of-control state has occurred, external intervention is required to restore the process to the in-control state.

The sampling and inspection procedure of fully-curtailed sampling plan is as follows.

After the production of every k units, units are inspected one by one until g conforming units are observed or m noncomforming units are observed. If g conforming units are observed, the process is declared to be in control and the production continues. If m nonconforming units are observed, the process is declared to be out of control, the production is halted and a search for the assignable cause is undertaken.

We want to find the optimal values of the design variables g, m and k which minimize the total expected cost per unit of controlling the process between two successive samples.

### 4.2.3 The Expected Cost Model

The total expected cost per unit of the product associated with controlling the process between two successive samples can be expressed as

$$E(C) = E(C_1) + E(C_2) + E(C_3) \qquad \text{...(4.2.1)}$$

where $E(C_1)$ is the expected cost per unit associated with sampling and inspection, $E(C_2)$ is the expected cost per unit associated with investigating and correcting the process, and

$E(C_3)$ is the expected cost per unit of producing a nonconforming unit.

**Computation of $E(C_1)$**

$$E(C_1) = (a_1 + a_2 \sum_{i=0}^{s} \alpha_i \bar{n}_i)/k \qquad \ldots(4.2.2)$$

where

$a_1$ = the fixed cost of sampling,

$a_2$ = the variable cost per unit of sampling,

$\alpha_i$ = the steady state probability that the process is in state $p_i$ at the time the sample is taken ($i = 0, 1, \ldots, s$),

$\bar{n}_i$ = the average sample number when the process is in state $p_i$ ($i = 0, 1, \ldots, s$).

**Computation of $E(C_2)$**

Let the cost of investigating and correcting the process plus the cost of lost production while searching for and possibly correcting the assignable cause be a random variable V with $E(V) = a_3$. It is assumed that the distribution of V does not depend upon the true process fraction of nonconforming units. Hence the cost of investigating real and false alarms is the same.

$$E(C_2) = a_3 \sum_{i=0}^{s} \alpha_i q_i /k \qquad \ldots(4.2.3)$$

where

$a_3$ = the average cost of investigating the process including the cost of lost production,

$q_i$ = the probability of concluding that the process is out of control when it is in state $p_i$ ($i = 0, 1, \ldots, s$).

**Computation of $E(C_3)$**

$$E(C_3) = a_4 \sum_{i=0}^{s} r_i p_i \qquad \ldots(4.2.4)$$

where

$a_4$ = the penalty cost of producing a nonconforming unit,

$r_i$ = the steady state probability that the process is in state $p_i$ at any point of time ($i$ = 0, 1, ..., s).

Thus the total expected cost per unit is given by the expression

$$E(C) = \frac{a_1 + a_2 \sum_{i=0}^{s} \alpha_i \bar{n}_i}{k} + \frac{a_3 \sum_{i=0}^{s} \alpha_i q_i}{k} + a_4 \sum_{i=0}^{s} r_i p_i \qquad \ldots(4.2.5)$$

$$= \frac{a_1 + a_2 \underline{\alpha}' \underline{\bar{n}}}{k} + \frac{a_3\ \underline{\alpha}' \underline{q}}{k} + a_4\ \underline{r}' \underline{p} \qquad \ldots(4.2.6)$$

where $\underline{q}$, $\underline{\alpha}$, $\underline{r}$, $\underline{p}$ are column vectors.

The cost coefficients $a_i$($i$=1,2,3,4) and the vector of fraction of nonconforming units $\underline{p}$ are known and are independent of design variables (g, m, k). The probability vectors $\underline{q}$, $\underline{\alpha}$, $\underline{r}$and the vector $\underline{\bar{n}}$ are functionally related to the design variables (g, m, k). In the next two sections we develop the expressions for the vectors $\underline{\alpha}$, $\underline{r}$, $\underline{q}$ and $\underline{\bar{n}}$.

### 4.2.4 The Probability Vectors $\underline{\alpha}$ and $\underline{r}$

$\alpha_i$ ($i$ = 0, 1, ..., s) is the steady state probability that the process is in state $p_i$ at the time the sample is taken. To obtain these steady state probabilities, the transition probability matrix B is required. The elements of B, say $b_{ij}$, represent the transition probabilities from $p_i$ to $p_j$ during the production of k units between two successive samples. Define $P_{ij}$ as the probability that the process shifts directly from the state $p_i$ to state $p_j$ during the production of k units between two successive samples. Suppose R units are produced per unit time. Then the probability that the process remains in the in-control state during the production of k units is

$$P_{oo} = 1 - \int_0^{k/R} \lambda\, e^{-\lambda t}\, dt \doteq \exp\,(-\lambda k/R) \qquad \ldots(4.2.7)$$

The production of k units takes k/R hours.

The probability that the process shifts from an in-control state $p_o$ to any of the s out of control states $p_1$, $p_2$, ... ,$p_s$ during the production of k units is

$$1 - p_{oo} = 1 - \exp(-\lambda k/R) \qquad \ldots(4.2.8)$$

We shall employ the method of Knappenberger and Grandage (1969) for distributing this probability. Using Knappenberger and Grandage's method we define

$$P_{oj} = \frac{\binom{s}{j}\, \pi^j (1-\pi)^{s-j}}{[1 - (1-\pi)^s]}\ [1 - \exp\,(-\lambda k/R)]$$

$$j = 1,\ 2,\ \ldots,\ s \qquad \ldots(4.2.9)$$

The probability distribution $\{P_{o1}, P_{o2}, \ldots, P_{os}\}$ represents the probability of shift from state $p_o$ to any other state $p_j$ directly during the production of k units.

We further define,

$$p_{ij} = \begin{cases} p_{oj} / (1 - p_{oo}) & \text{if } j > i > 0 \\ \sum_{x=1}^{j} p_{ox} / (1 - p_{oo}) & \text{if } j = i > 0 \\ 0 & \text{if } j < i > 0 \end{cases} \qquad \ldots(4.2.10)$$

We now define the elements of matrix B. When $0 \leq j < i$, $b_{ij}$ is the probability that the process is in state $p_i$ at the time of mth sample and has shifted to a better state $p_j$ at the time of (m+1)st sample. This is the probability of detecting an out-of-control state on mth sample times the probability that the process shifts from $p_o$ to state $p_j$ during the production of k units. That is

$$b_{ij} = q_i p_{oj} \qquad \text{if } 0 \leq j < i \qquad \ldots(4.2.11)$$

By similar argument we write

$$b_{ij} = q_i p_{oj} + (1 - q_i)\, p_{ij} \qquad \text{if } j > i \qquad \ldots(4.2.12)$$

This is the probability that the state $p_i$ is detected at the time of mth sample and the process has shifted from state $p_o$ to state $p_j$ at (m+1)st sample plus the probability that the state

$p_i$ is not detected at mth sample and the process further deteriorates from $p_i$ to $p_j$ at (m+1)st sample.

$$b_{ii} = q_i p_{oi} + (1 - q_i)p_{ii} \qquad \text{if } i > 0 \qquad \ldots(4.2.13)$$

Finally

$$b_{oj} = p_{oj} \qquad \text{if } j = 1, 2, \ldots, s \qquad \ldots(4.2.14)$$

The matrix B is the transition probability matrix of an irreducible, aperiodic, positive recurrent Markov Chain. Hence there exists a vector $\underline{\alpha}$ such that

$$\underline{\alpha}' B = \underline{\alpha}' \qquad \ldots(4.2.15)$$

where ith element $\alpha_i$ of vector $\underline{\alpha}$ is the steady state unconditional probability that the process is in state $p_i$ at the time the sample is taken, regardless of the initial state.

We can find $\underline{\alpha}$ by solving any s of the (s+1) equations (4.2.15) along with the restriction $\sum_{i=0}^{s} \alpha_i = 1$.

We now define the elements of vector $\underline{r}$ where $r_i$ (i = 0, 1, ..., s) is the steady state probability that the process is in state $p_i$ at any point of time. First consider $r_o$. The probability $r_o$ depends on the probability that the process starts in state $p_o$ at the time of mth sample and remains in the same state until (m+1)st sample, and the probability that the process starts in state $p_o$ at the time of mth sample and shifts to an out-of-control state until (m+1)st sample. Thus,

$$r_o = \alpha_o p_{oo} + \Delta\alpha_o(1 - p_{oo}) \qquad \ldots(4.2.16)$$

where $\Delta$ is the average fraction of the time that elapses before the shift occurs, given that the shift occurs between mth and (m+1)st sample.

Duncan (1956) has shown that

$$\Delta = \frac{1 - (1 + \lambda k/R) \exp(-\lambda k/R)}{(\lambda k/R)[1 - \exp(-\lambda k/R)]} \qquad \ldots(4.2.17)$$

We assume that the average fraction of time the process spends in the lower state when a transition to a higher state is made is the same as $\Delta$.

The probabilities $r_i$ $(i = 1, 2, \ldots, s)$ depends on the probability that the process is in state $p_i$ at the time of mth sample and stays there until (m+1)st sample, the process is in some better state $p_k$ $(k = 0, 1, 2, \ldots, i-1)$ at the time of mth sample and shifts to state $p_i$ before (m+1)st sample, and the probability that the process is in state $p_i$ at the time of mth sample and shifts to some higher state $p_j$ $(j = i+1, \ldots, s)$ before (m+1)st sample. Thus,

$$r_i = \alpha_i p_{ii} + (1 - \Delta) \sum_{k=0}^{i-1} \alpha_k p_{ki} + \Delta \sum_{j=i+1}^{s} \alpha_i p_{ij}$$

$$(i = 1, 2, \ldots, s) \qquad \ldots(4.2.18)$$

### 4.2.5 The Vectors $\underline{q}$ and $\underline{\bar{n}}$

Let Y denote the number of units sampled when mth nonconforming unit is observed.

Let Z denote the number of units sampled when gth conforming unit is observed.

Then Y has a negative binomial distribution with probability function

$$q_i(y) = \binom{y-1}{m-1} p_i^m (1-p_i)^{y-m} \qquad \ldots(4.2.19)$$

$$y = m,\ m+1,\ \ldots,\ n$$

$$\therefore\ q_i = \sum_{y=m}^{n} q_i(y) \qquad \ldots(4.2.20)$$

$$i = 0, 1, \ldots, s$$

where $q_i$ represents the probability of concluding that the process is out of control when it is in state $p_i$ $(i = 0, 1, \ldots, s)$.

Z has a negative binomial distribution with the probability function

$$s_i(z) = \binom{z-1}{g-1} (1 - p_i)^g\ p_i^{z-g} \qquad \ldots(4.2.21)$$

$$z = g,\ g+1, \ldots, n$$

Taking $m+g = n+1$ the average sample number, $\bar{n}_i$, when the process is in state $p_i$ is

$$\bar{n}_i = \sum_{y=m}^{n} y\binom{y-1}{m-1} p_i^m (1-p_i)^{y-m} + \sum_{z=g}^{n} z\binom{z-1}{g-1}(1-p_i)^g\ p_i^{z-g}$$

$$i = 0,\ 1,\ \ldots, s \qquad \ldots(4.2.22)$$

The simplified form of $\bar{n}_i$ as given by Shah and Phatak (1972) is

$$\bar{n}_i = (1-q_i)\ \frac{p_i(n+1)-m}{p_i(1-p_i)} + \frac{m}{p_i}\left[\ 1 - \binom{n}{m}\ p_i^m\ (1-p_i)^{n-m}\ \right]$$

$$i = 0,\ 1,\ \ldots, s \qquad \ldots(4.2.23)$$

Patil (1960) showed that the expression for $q_i$ remains the same under uncurtailed, semi-curtailed and fully-curtailed sampling policies and is

$$q_i = \sum_{d=m}^{n} \binom{n}{d} p_i^d (1 - p_i)^{n-d} \qquad \ldots(4.2.24)$$

$$i = 0, 1, \ldots, s$$

This equivalence is due to the relationship between the distribution functions of the binomial distribution and the negative binomial distribution. This relationship was also established by Morris (1963) independently.

Thus for all the sampling policies ith element $q_i$ of vector $\underline{q}$ is given by a common expression (4.2.24).

### 4.2.6 Solution Method and Sample Example

We have used Hooke-Jeeves' search procedure explained in Section 2.3.4 to find the optimal values of the design variables (g, m, k) which minimize the expected cost per unit of the product given by expression (4.2.6). The search technique gives the optimal values of n, m and k. One gets the optimal value of g using the relation $g = n - m + 1$. The details of this procedure are well explained in chapter II.

We consider an example presented in Table 2 of Williams et al. (1985).

Let $a_1$ = \$1, $a_2$ = \$0.1, $a_3$ = \$100, $a_4$ = \$10

Let $\lambda = 1$, $R = 1000$, $\pi = 0.597$, $s = 6$

Let

$(p_0, p_1, p_2, p_3, p_4, p_5, p_6) = (.01, .02, .04, .08, .16, .32, .64)$

For this combination of cost coefficients and systems parameters, the search technique yielded the following optimal procedure :

$g = 7$, $m = 2$, $k = 19$ with minimum $E(C) = \$0.4069$.

i.e. after every production of 19 units, one inspects the units one by one until 7 conforming units are observed or 2 nonconforming units are observed. In the first case the process is declared to be in control and in the second case the process is declared to be out of control.

For this example the optimal control procedure yielded under semi-curtailed sampling is $n = 8$, $m = 2$, $k = 20$ with minimum $E(C) = \$0.4115$. The optimal control procedure yielded under complete sampling is $n = 8$, $m = 2$, $k = 20$ with minimum $E(C) = \$0.4118$.

This indicates that the use of semi-curtailed sampling in place of complete sampling provides an improvement of 0.0003 whereas the use of fully-curtailed sampling in place of complete sampling provides an improvement of 0.0049.

### 4.2.7 Comparision of Complete, Semi-Curtailed and Fully-Curtailed Sampling Scheme.

#### (A) Theoretical Comparision

We have seen earlier that the probability vector $\underline{q}$ is the same for all the three sampling policies. The vector $\underline{\alpha}$ depends upon the vector $\underline{q}$ and the transition probability matrix $(p_{ij})$ which is the same regardless of which sampling policy is used. Therefore the vector $\underline{\alpha}$ is the same for all the three sampling policies. Hence $E(C_2)$ is the same for all the three sampling policies. The probability vector $\underset{\sim}{r}$ depends upon the vector $\underset{\sim}{\alpha}$, $\Delta$ and the transition probability matrix $(p_{ij})$. The vector $\underset{\sim}{\alpha}$ is the same for all the three sampling policies. The matrix $(p_{ij})$ and $\Delta$ are external to the type of sampling policy. Hence vector $\underset{\sim}{r}$ remains the same for all the three sampling policies. The vector $\underline{p}$ is assumed to be known. Hence $E(C_3)$ is the same for all the three sampling policies.

It now remains only to compare $E(C_1)$ under the three sampling policies.

For complete single sampling

$$E(C_1) = (a_1 + a_2 n)/k \qquad \ldots(4.2.25)$$

For both the types of curtailed single sampling

$$E(C_1) = (a_1 + a_2 \sum_{i=0}^{s} \alpha_i \bar{n}_i)/k \qquad \ldots(4.2.26)$$

The expression for $\bar{n}_i$ under fully-curtailed sampling is given by (4.2.22) and (4.2.23). The expression for $\bar{n}_i$ under semi-curtailed sampling is given by

$$\bar{n}_i = n \sum_{d=0}^{m-1} \binom{n}{d} p_i^d (1-p_i)^{n-d} + \sum_{y=m}^{n} y \binom{y-1}{m-1} p_i^m (1-p_i)^{y-m}$$

which Phatak and Bhatt (1967) have shown to be equal to

$$n \sum_{d=0}^{m-1} \binom{n}{d} p_i^d (1-p_i)^{n-d} + (m/p_i)\left[1 - \sum_{d=0}^{m} \binom{n+1}{d} p_i^d (1-p_i)^{n+1-d}\right]$$

$$(i = 0, 1, \ldots, s) \qquad \ldots(4.2.27)$$

Phatak and Bhatt (1967) have also shown that

$$\bar{n}_{i(\text{fully-curt})} \leq \bar{n}_{i(\text{semi-curt})} \leq n \qquad \ldots(4.2.28)$$

$$(i = 0, 1, \ldots, s)$$

This result and the fact that $\sum_{i=0}^{s}\alpha_i = 1$ implies that

$$E(C_1)_{\text{fully-curt}} \leq E(C_1)_{\text{semi-curt}} \leq E(C_1)_{\text{uncurt}} \qquad \ldots(4.2.29)$$

if these expected costs are calculated for the same values of n, m, k.

This leads to a conclusion that

$$E(C)_{\text{fully-curt}} \leq E(C)_{\text{semi-curt}} \leq E(C)_{\text{uncurt}} \qquad \ldots(4.2.30)$$

if these total expected costs are calculated for the same values of n, m, k.

This shows that the fully-curtailed sampling is no more expensive than semi-curtailed sampling and the semi-curtailed sampling is no more expensive than uncurtailed sampling for those cases where the optimal procedures yielded under the three policies are the same.

We now show that the same result holds for those cases also where the optimal procedures yielded under the three sampling policies are different.

Let $(n_u, m_u, k_u)$, $(n_s, m_s, k_s)$ and $(n_f, m_f, k_f)$ represent the optimal procedures under the uncurtailed, the semi-curtailed and the fully-curtailed sampling policies respectively.

Let $E_u(n, m, k)$, $E_s(n, m, k)$ and $E_f(n, m, k)$ represent the expected cost per unit of the product under uncurtailed, semi-curtailed and fully-curtailed sampling policies respectively. It is clear from the concept of optimality that

$$E_f(n_f, m_f, k_f) \leq E_f(n, m, k) \quad \text{for all } n, m, k. \qquad \ldots(4.2.31)$$

$$E_s(n_s, m_s, k_s) \leq E_s(n, m, k) \quad \text{for all } n, m, k. \qquad \ldots(4.2.32)$$

$$E_u(n_u, m_u, k_u) \leq E_u(n, m, k) \quad \text{for all } n, m, k. \qquad \ldots(4.2.33)$$

We therefore have,

$$\begin{aligned} E_f(n_f, m_f, k_f) &\leq E_f(n_s, m_s, k_s) \quad \text{by } (4.2.31) \\ &\leq E_s(n_s, m_s, k_s) \quad \text{by } (4.2.30) \\ &\leq E_s(n_u, m_u, k_u) \quad \text{by } (4.2.32) \\ &\leq E_u(n_u, m_u, k_u) \quad \text{by } (4.2.30) \end{aligned}$$

Thus,

$$E_f(n_f, m_f, k_f) \leq E_s(n_s, m_s, k_s) \leq E_u(n_u, m_u, k_u) \qquad \ldots(4.2.34)$$

Williams et al. (1985) theoretically established the second part of the inequality (4.2.30) while comparing the semi-curtailed sampling scheme with the uncurtailed sampling scheme. Using this inequality they arrived at a conclusion that the semi-curtailed sampling scheme is no more expensive than the uncurtailed sampling scheme if the optimal design variables yielded under these two schemes are the same. But they could not establish theoretically that the same result holds even if the optimal design variables yielded under the two policies are different.

Furthermore, in their recent paper (1990) appeared in the Journal of Quality and Technology where they have introduced the use of fully-curtailed sampling scheme, they do not make any mention of the theoretical comparision of the fully-curtailed sampling scheme with the other two sampling schemes.

We would like to mention from the priority point of view that we have introduced the use of the fully-curtailed sampling scheme in the paper read at 1st Assian Congress on Quality and Reliablilty (1989) held at New Delhi. In that paper we have established the result (4.2.34) given in this section. The importance of this result is that the fully-curtailed sampling is no more expensive than semi-curtailed sampling and the semi-curtailed sampling is no more expensive than the uncurtailèd sampling even if the optimal design variables under the three sampling policies are different.

### (B) Numerical Comparision

The optimal design variables and the minimum expected costs are obtained for complete sampling policy and for semi-curtailed sampling policy by Williams et. al. (1985), for various cost coefficients and systems parameters. We have studied all 81 combinations considered by them and have obviously found that for each combination the minimum cost yielded by the optimal fully-curtailed sampling plan is less than or equal to the minimum cost yielded by the optimal semi-curtailed sampling plan. Equality holds if $m = 1$ since in this case the fully-curtailed sampling plan is equivalent to the semi-curtailed sampling plan. We have presented the optimal uncurtailed, semi-curtailed and fully-curtailed sampling plans with the minimum expected costs for a few combinations in Table 4.1.

Let $E(C)_u$, $E(C)_s$ and $E(C)_f$ denote the minimum costs of the optimal control procedures of the uncurtailed, semi-curtailed and fully-curtailed sampling policies respectively. Then the percentage improvement in the cost of the fully-curtailed and the semi-curtailed sampling plans over the uncurtailed sampling plan are given by

$$I_{fu} = 100 \left[ \frac{E(C)_u - E(C)_f}{E(C)_u} \right] \%$$

$$I_{su} = 100 \left[ \frac{E(C)_u - E(C)_s}{E(C)_u} \right] \% \qquad \ldots(4.2.35)$$

respectively. The percentage improvement in the cost of the fully-curtailed and the semi-curtailed sampling plans over the uncurtailed sampling plan are given in columns (15) and (16) of Table 4.1. It may be noted that all the numerical results given in Table 4.1 tally with the numerical results given by Williams et. al. (1990). The priority of these results is already established in part (A) of this section.

### 4.2.8 Conclusion

It is concluded that the improvement in cost due to fully-curtailed sampling over the traditional complete sampling is considerably large as compared to the improvement in cost due to semi-curtailed sampling over complete sampling. Hence if the goal of sampling is only to decide whether the process should be left

## Table 4.1

The optimal values of the design variables with minimum expected costs for the three sampling policies, the percent improvement in cost of semi-curtailed and fully-curtailed sampling over complete sampling.

| | | Uncurtailed | | | | Semi-curtailed | | | | Fully-curtailed | | | | Percent Improvement in cost | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $a_1$ (1) | $\pi$ (2) | n (3) | m (4) | k (5) | E(C) (6) | n (7) | m (8) | k (9) | E(C) (10) | g (11) | m (12) | k (13) | E(C) (14) | Isu (15) | Ifu (16) |
| 1 | .376 | 13 | 2 | 32 | .3807 | 14 | 2 | 87 | .3801 | 12 | 2 | 34 | .3777 | .1576 | .7880 |
| 5 | .376 | 23 | 2 | 80 | .4508 | 23 | 2 | 80 | .4497 | 22 | 2 | 79 | .4488 | .2440 | .4437 |
| 10 | .376 | 30 | 2 | 118 | .5019 | 30 | 2 | 117 | .5004 | 29 | 2 | 117 | .4999 | .2989 | .3985 |
| 1 | .597 | 8 | 2 | 20 | .4118 | 8 | 2 | 20 | .4115 | 7 | 2 | 19 | .4069 | .0729 | 1.1899 |
| 5 | .597 | 15 | 2 | 46 | .5323 | 15 | 2 | 46 | .5315 | 14 | 2 | 46 | .5297 | .1503 | .4884 |
| 10 | .597 | 19 | 2 | 66 | .6217 | 19 | 2 | 66 | .6206 | 18 | 2 | 66 | .6194 | .1769 | .3700 |
| 1 | .800 | 6 | 2 | 15 | .4352 | 6 | 2 | 15 | .4350 | 5 | 2 | 15 | .4288 | .0460 | 1.4706 |
| 5 | .800 | 10 | 2 | 32 | .6068 | 10 | 2 | 32 | .6062 | 9 | 2 | 32 | .6035 | .0989 | .5438 |
| 10 | .800 | 13 | 2 | 46 | .7358 | 13 | 2 | 46 | .7350 | 12 | 2 | 46 | .7332 | .1087 | .3534 |

Here $a_2$ = \$ .1, $a_3$ = \$ 100, $a_4$ = \$ 10, $\lambda = 1$, R = 1000, s = 6

p = (.01, .02, .04, .08, .16, .32, .64)

alone or whether the process should be halted to find the assignable cause then fully-curtailed sampling is the most appropriate among all the three sampling policies from cost point of view.

## 4.3 Use of Double Sampling Plan in the Economic Design of np-Control Chart

**4.3.1** It is known that the double sampling plan involves less total inspection on average than the single sampling plan for any given quality protection (Grant (1964)). The plan also has certain psychological advantages based on the idea of giving a second chance to doubtful lots, since the plan may defer the decision of acceptance or rejection until a second sample has been inspected. Hence it is decided to give due consideration to the double sampling plan in the development of the economic design of np-control chart.

In this section we develop the expected cost model using double sampling plan as a sampling policy. The model is analogous in all other respects to the single sampling model developed by Montgomery et. al. (1975) which is well explained in the earlier part of this chapter. We compare the performance of double sampling plan with that of single sampling plan uncurtailed as well as curtailed for a variety of cost coefficients and systems parameters. The results indicate that the use of double sampling plan leads to smaller expected costs than the single sampling plan uncurtailed as well as curtailed.

### 4.3.2 The Production Process and the Sampling Policy

The production process is the same as the one described in Section 4.2.2.

The inspection procedure is as follows.

After every production of k units a sample of n units is examined. Let $d_1$ be the number of nonconforming units found in the sample.

If $d_1 \leq A_1$ the process is declared to be in control.

If $d_1 > A_2$ the process is declared to be out of control.

If $A_1 < d_1 \leq A_2$ a second sample of n units is examined.

Let $d_2$ be the number of nonconforming units found in the second sample.

If $d_1 + d_2 \leq A_2$ the process is declared to be in control.

If $d_1 + d_2 > A_2$ the process is declared to be out of control.

This is a procedure of usual double sampling plan with $n_1 = n_2 = n$ (say).

### 4.3.3 The Expected Cost Model

The cost model is the same as the one described in Section 4.2.3.

The expression for total expected cost per unit of the product under double sampling policy is the same as the one derived for the single sampling policy and is reproduced from (4.2.6) as follows.

$$E(C) = \frac{a_2 + a_2\,\underline{\alpha}'\underline{\bar{n}}}{k} + \frac{a_3\underline{\alpha}'\underline{\underline{q}}}{k} + a_4\underline{r}'\underline{p} \qquad \ldots(4.3.1)$$

The procedures for the calculation of $\underline{\alpha}$ and $\underline{r}$ remain the same as those described in Section 4.2.4. The expressions for $\underline{\alpha}$ and $\underline{r}$ are given by (4.2.15) and (4.2.18) respectively. However we would like to mention clearly that the vector $\underline{q}$ required in the calculation of $\underline{\alpha}$ and $\underline{r}$ is different from (4.2.24) when double sampling policy is used. The expression for $\bar{n}_i$ $(i=0,1,\ldots,s)$ is also different from (4.2.22) when double sampling policy is used. The expressions for $\underline{q}$ and $\underline{\bar{n}}$ required in computation of (4.3.1) are derived in the next section.

### 4.3.4 The Vectors $\underline{q}$ and $\underline{\bar{n}}$ under Double Sampling Plan

Let $P_{r.1}(i)$ denote the probability, based on the evidence of first sample, of concluding that the process is out of control when the process is in state $p_i(i=0,1,\ldots,s)$. Let the corresponding probability based on the evidence of the second sample be denoted by $P_{r.2}(i)$. Lastly, let $P_{a.1}(i)$ denote the probability, based on the evidence of the first sample, of concluding that the process is in control when it is in state $p_i$.

$$P_{a.1}(i) = \sum_{d_1=0}^{A_1} \binom{n}{d_1} p_i^{d_1} (1 - p_i)^{n-d_1} \qquad \ldots(4.3.2)$$

$$P_{r.1}(i) = \sum_{d_1=A_2+1}^{n} \binom{n}{d_1} p_i^{d_1} (1 - p_i)^{n-d_1} \qquad \ldots(4.3.3)$$

$$P_{r.2}(i) = \sum_{d_1=A_1+1}^{A_2} \sum_{d_2=A_2+1-d_1}^{n} \left[ \binom{n}{d_1} p_i^{d_1} (1 - p_i)^{n-d_1} * \binom{n}{d_2} p_i^{d_2} (1 - p_i)^{n-d_2} \right] \qquad ...(4.3.4)$$

Then,

$$q_i = P_{r.1}(i) + P_{r.2}(i) \qquad ...(4.3.5)$$

$$i = 0,\ 1,\ ....,\ s$$

where $q_i$ represents the probability of concluding that the process is out of control when it is in state $p_i$ $(i=0,1,....,s)$.

The expression for $\bar{n}_i$ $(i = 0,\ 1,\ ....,\ s)$ is

$$\bar{n}_i = n\ [\ P_{a.1}(i) + P_{r.1}(i)\ ] + 2n\ [\ 1 - P_{a.1}(i) - P_{r.1}(i)\ ]$$

$$i = 0,\ 1,\ ....,s \qquad ...(4.3.6)$$

### 4.3.5 Solution Method and Sample Example

We have used Hooke-Jeeves' search technique explained in Section 2.3.4 to obtain the optimal values of the design variables $n$, $A_1$, $A_2$, $k$ which minimize the expected cost per unit of the product given by the expression (4.3.1).

We consider the same example given in Section 4.2.6.

Let $a_1$ = \$1, $a_2$ = \$0.1, $a_3$ = \$100, $a_4$ = \$10.

Let $\lambda = 1$, $R = 1000$, $\pi = 0.597$, $s = 6$.

Let

$(p_0, p_1, p_2, p_3, p_4, p_5, p_6) = (.01, .02, .04, .08, .16, .32, .64)$

For this combination of cost coefficients and systems parameters the search technique yielded the following optimal procedure when double sampling scheme is used.

$n = 5$, $A_1 = 0$, $A_2 = 1$, $k = 19$ with minimum $E(C)$ = \$0.3961. i.e.

after every production of 19 units, one examines 5 units from the process. If no nonconforming unit is found in this sample the process is declared to be in control and the production continues. If the sample contains 2 or more nonconforming units the process is declared to be out of control. If the sample contains only one nonconforming unit, one examines next 5 units from the process and determines the number of nonconforming units in the combined sample of $5 + 5 = 10$ units. If the number of nonconforming units in the combined sample is greater than or equal to 2, the process is declared to be out of control, otherwise it is considered to be in control.

For this sample example the optimal control procedure under complete single sampling is $n = 8$, $m = 2$, $k = 20$ with minimum $E(C) = \$0.4118$ where $n$ is the sample size, $m$ is the rejection number, $k$ is the inter sample range. The optimal control procedure under semi-curtailed single sampling is $n = 8$, $m = 2$, $k = 20$ with minimum $E(C) = \$0.4115$. The optimal control procedure under fully-curtailed single sampling is $n = 8$, $m = 2$, $k = 19$ with minimum $E(C) = \$0.4069$.

This indicates that the use of semi-curtailed single sampling and fully-curtailed single sampling in place of complete single sampling provides an improvement of 0.0003 and 0.0049 respectively; whereas the use of double sampling in place of complete single sampling provides an improvement of 0.0157 per unit. This means that double sampling is better than the single sampling uncurtailed as well as curtailed from the cost point of view.

### 4.3.6 Numerical Comparision

The optimal control procedures and the minimum costs are obtained for complete single sampling policy by Williams et al. (1985) for various cost coefficients and operating conditions. We have studied all the combinations considered by them and have observed that whenever there is a scope for double sampling scheme (i.e. when the size of the sample for single sampling scheme is not too small and when the rejection number is greater than 1) the minimum cost yielded under double sampling is less then that yielded under single sampling uncurtailed as well as curtailed. We have observed that in 28 combinations among 81 combinations considered by Williams et al. (1985) the sample size $n > 3$ and the rejection number $m > 1$ for optimal single sampling plans. For each of these combinations the minimum cost yielded under double sampling is smaller than that yielded under single sampling.

The percentage improvement in the cost of the double sampling plan over the complete single sampling plan is given by

$$I_{du} = 100 \left[ \frac{E(C)_d - E(C)_u}{E(C)_u} \right] \%$$

We have also observed that the percent improvement in the cost due to double sampling over complete single sampling is considerably large as compared to the percent improvement in cost due to fully-curtailed single sampling over complete single sampling.

We present a few optimal double sampling plans along with their minimum costs in Table 4.2. The percent improvement in cost of semi-curtailed, fully-curtailed and double sampling policies over complete sampling are given in columns (9), (10), (11) of Table 4.2.

### 4.3.7 Conclusion

The improvement in the cost due to double sampling over complete sampling is considerably large as compared to the improvement in cost due to both types of curtailed single sampling over complete single sampling. Hence double sampling is more appropriate than complete as well as curtailed single sampling from cost point of view.

## Table 4.2

The optimal values of the design variables of double sampling plan with minimum E(C) and percent improvement in cost of semi-curtailed sampling, fully-curtailed sampling and double sampling over complete single sampling.

| $a_1$ (1) | $\pi$ (2) | $n_1$ (3) | $n_2$ (4) | $A_1$ (5) | $A_2$ (6) | k (7) | E(C) (8) | Isu (9) | Ifu (10) | Idu (11) |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.376 | 8 | 8 | 0 | 1 | 33 | 0.3677 | 0.1576 | 0.7880 | 3.4410 |
| 5 | 0.376 | 14 | 14 | 0 | 1 | 73 | 0.4433 | 0.2440 | 0.4437 | 1.6637 |
| 10 | 0.376 | 18 | 18 | 0 | 1 | 115 | 0.4963 | 0.2989 | 0.3985 | 1.1158 |
| 1 | 0.597 | 5 | 5 | 0 | 1 | 19 | 0.3961 | 0.0729 | 1.1899 | 3.8125 |
| 5 | 0.597 | 9 | 9 | 0 | 1 | 45 | 0.5226 | 0.1503 | 0.4884 | 1.8223 |
| 10 | 0.597 | 12 | 12 | 0 | 1 | 66 | 0.6142 | 0.1769 | 0.3700 | 1.2064 |
| 1 | 0.800 | 4 | 4 | 0 | 1 | 15 | 0.4204 | 0.0460 | 1.4706 | 3.4007 |
| 5 | 0.800 | 6 | 6 | 0 | 1 | 31 | 0.5966 | 0.0989 | 0.5438 | 1.6809 |
| 10 | 0.800 | 8 | 8 | 0 | 1 | 45 | 0.7279 | 0.1087 | 0.3534 | 1.0737 |

Here $a_2$ = \$ .1, $a_3$ = \$ 100, $a_4$ = \$ 10, $\lambda = 1$, R = 1000; s = 6

p = (.01, .02, .04, .08, .16, .32, .64).

## 4.4 Comparision of Complete, Semi-Curtailed and Fully-Curtailed Sampling policies for the Economic Model developed in Chapter II

Under this economic model the expected cost per unit of the product is

$$ECPU = [\ E(C_1) + E(C_2) + E(C_3)\ ]\ /\ Nk \qquad ...(4.4.1)$$

Firstly we show that $E(C_2)$ and $E(C_3)$ remain the same under all the three sampling policies if they are calculated for the same design variables (n, m, k).

$$E(C_2) = a_{3,1}\ B_o + a_{3,2}\ 1 \qquad ...(4.4.2)$$

The expression for $B_o$ is

$$B_o = q_o\ \theta/(1 - \theta) \qquad ...(4.4.3)$$

where $\theta = \exp(-\lambda k/R)$.

As discussed earlier in Section 4.2.5 of this chapter, due to the relationship between the binomial distribution and the negative binomial distribution, the expresions for $q_i$ $(i = 0, 1)$ remain same for all the three sampling policies if they are calculated for the same values of n and m. The expression for $\theta$ is external to the sampling policy used. Hence $E(C_2)$ remains the same for all the three sampling policies.

Next,

$$E(C_3) = a_{4,1}\ S + a_{4,2}\ (D - S) \qquad ...(4.4.4)$$

where $$S = \frac{np_o\ \theta}{1-\theta} + \frac{np_1}{q_1}, \qquad ...(4.4.5)$$

$$D = \left[ \frac{k\theta}{1-\theta} + \Delta\ k \right] p_0 + \left[ \frac{k}{q_1} - \Delta\ k \right] p_1 \qquad ...(4.4.6)$$

We have just stated that $q_1$ remains the same for all the three sampling policies. We have also noted that the expression for $\Theta$ is external to the sampling policy used. Same is true for $\Delta$. The values $p_0$ and $p_1$ are known and are independent of the sampling policy used. Hence $E(C_3)$ remains the same for all the three sampling policies.

It now remains to compare $E(C_1)$ under the three sampling policies if $E(C_1)$ is calculated for the same triplet (n, m, k). Firstly under complete sampling the expression for $E(C_1)$ is given by

$$E(C_1) = (a_1 + a_2 n)\, N \qquad \text{...(4.4.7)}$$

where $N = \dfrac{\Theta}{1-\Theta} + \dfrac{1}{q_1}$.

Secondly under curtailed sampling the expression for $E(C_1)$ is given by

$$E(C_1) = \frac{(a_1 + a_2\bar{n}_0)\,\Theta}{1-\Theta} + \frac{(a_1 + a_2\bar{n}_1)}{q_1} \qquad \text{...(4.4.8)}$$

where $\bar{n}_i$ (i = 0, 1) is the average sample number when the process is in state $p_i$ (i = 0, 1).

Phatak and Bhatt (1967) have shown that

$$\bar{n}_{i(\text{fully-curt})} \leq \bar{n}_{i(\text{semi-curt})} \leq n_{(\text{uncurt})} \qquad (i = 0, 1)$$

This result implies that

$$E(C_1)_{\text{fully-curt}} \leq E(C_1)_{\text{semi-curt}} \leq E(C_1)_{\text{uncurt}} \qquad \ldots(4.4.9)$$

if these expected costs are calculated for the same values of (n, m, k).

We therefore have

$$E(C)_{\text{fully-curt}} \leq E(C)_{\text{semi-curt}} \leq E(C)_{\text{uncurt}} \qquad \ldots(4.4.10)$$

if these costs are calculated for the same triplet (n, m, k).

This shows that fully-curtailed sampling is no more expensive than semi-curtailed sampling and semi-curtailed sampling is no more expensive than complete sampling for those cases where the optimal procedures yielded under the three policies are same.

Using the same argument as given in section 4.2.7 it can be shown that the above result holds for those cases also where the optimal procedures yielded under the three sampling policies are different.

**Numerical Example**

We consider the same example given in Section 2.3.4 of chapter II.

Let $a_1$ = \$10.0, $a_2$ = \$1.0, $a_{3,1}$ = \$100, $a_{3,2}$ = \$100,
$a_{4,1}$ = \$10.0, $a_{4,2}$ = \$15.0
Let $\lambda$ = 1, R = 1000, $p_0$ = 0.01, $p_1$ = \$0.10

For this combination of cost coefficients and systems parameters the optimal procedure yielded under complete sampling is

n = 37, m = 2, k = 350 with minimum ECPU = $ 0.5457.

The optimal procedure yielded under semi-curtailed sampling is n = 46, k = 342, m = 2 with minimum E(C) = $0.5127. The optimal procedure yielded under fully-curtailed sampling is n = 42, k = 345, m = 2 with minimum ECPU = $ 0.5097.

Hence the percent improvement in the cost of semi-curtailed sampling over complete sampling is 6.0473. Whereas the percent improvement in the cost due to fully-curtailed sampling over complete sampling is 6.5970. The optimal procedures under the three sampling policies are obtained by the direct search technique given by Hooke-Jeeves (1961).

```
C LISTING OF CHAPTER IV
        SUBROUTINE OBJ3(AKE,NSTAGE,SUMN,A1,A2,A3,A4,A5,RATE,ALEMDA,
     1  PNOT,PONE)
        DIMENSION AKE(10)
C FILE NAME IS NANDI2
C  PROGRAM FOR ECPU SEMI CURTAILED SAMPLING MY MODEL
         WRITE(*,1) A1,A2,A3,A4,A5
  1      FORMAT(1X,5F10.4)
        WRITE(*,3) ALEMDA,RATE
  3      FORMAT(1X,2F10.4)
        WRITE(*,3) PNOT,PONE
        SNOT=AKE(1)
        SRNOT=AKE(1)
        REJNOT=AKE(3)
        WRITE(*,5) SNOT,SRNOT,REJNOT
  5     FORMAT(1X,3F10.4)
        POWER=ALEMDA*SRNOT/RATE
        PPOWER=-POWER
        THEETA=EXP(PPOWER)
        WRITE(*,7) THEETA
  7     FORMAT(1X F10.6)
        MM=REJNOT
        NT=SNOT-REJNOT+1
        CALL BIN(PONE,MM,NT,CPR,CPL,PI)
        QONE=CPR
        WRITE(*,8)QONE
  8     FORMAT(1X,F10.6)
        MM=REJNOT
        NT=SNOT-REJNOT+1
        CALL BIN(PNOT,MM,NT,CPR,CPL,PI)
        QNOT=CPR
        WRITE(*,8)QNOT
        R=1/QONE+THEETA/(1-THEETA)
C   IR IS EXPECTED NO OF SAMPLES REQUIED TO DETECT SHIFT
        IR=R+0.5
        WRITE(*,9)IR
  9     FORMAT(1X,I3)
C    COMPUTATION  OF EXPECTED COSTS
        MM=REJNOT
        NT=SNOT-REJNOT+1
        CALL BIN(PNOT,MM,NT,CPR,CPA,PI)
        PAO=CPA
        NN=SNOT+1
        MM=REJNOT+1
        NT=NN-MM+1
        CALL BIN(PNOT,MM,NT,CPR ,CPA,PI)
        PRO=CPR
        ASNO=SNOT*PAO+REJNOT*PRO/PNOT
        MM=REJNOT
        NT=SNOT-REJNOT+1
        CALL BIN(PONE,MM,NT,CPR,CPA,PI)
        PA1=CPA
        NN=SNOT+1
        MM=REJNOT+1
        NT=NN-MM+1
        CALL BIN(PONE,MM,NT,CPR,CPA,PI)
        PR1=CPR
        ASN1=SNOT*PA1+REJNOT*PR1/PONE
```

```
      WRITE(*,40) ASNO,ASN1
 40   FORMAT(1X,'ASNO=',F10.4,'ASN1=',F10.4)
      ECT1=(A1+A2*ASNO)*THEETA/(1-THEETA)+(A1+A2*ASN1)/QONE
      EC1=ECT1/(IR*SRNOT)
      EC2=A3*(QNOT*THEETA/(1-THEETA)+1)/(IR*SRNOT)
      ADALTA=(1-(1+POWER)*THEETA)/(POWER*(1-THEETA)).
      WRITE(*,35)ADALTA
 35   FORMAT(1X,'ADALTA=', F10.6)
      D=THEETA*SRNOT*PNOT/(1-THEETA)+ADALTA*SRNOT*PNOT+SRNOT*PONE/
     1 QONE-ADALTA*SRNOT*PONE
      DS=THEETA*SNOT*PNOT/(1-THEETA)+SNOT*PONE/QONE
      WRITE(*,15)D,DS
15    FORMAT(1X,2F10.4)
C   DS GIVES  EXPECTED  NO OF  DEFECTIVES  DETECTED  IN  SAMPLING
C   D  GIVES   EXPECTED  NO  OF  DEFECTIVES PRODUCED
      EC3=(A4*DS+A5*(D-DS))/(IR*SRNOT)
      TC=EC1+EC2+EC3
      SUMN=TC
      WRITE (*,30)TC,EC1,EC2,EC3
 30   FORMAT(1X,'TC=',F10.6,'EC1=',F10.6,'EC2=',F10.6,'EC3=',F10.6)
      RETURN
      END


      SUBROUTINE OBJ4(AKE,NSTAGE,SUMN,A1,A2,A3,A4,A5,RATE,ALEMDA,
     1 PNOT,PONE)
      DIMENSION  AKE(10)
C FILE NAME IS NANDI3
C  PROGRAM FOR ECPU FULLY CURTAILED SAMPLING MY MODEL
  1   format(1x,5f10.4)
      write(*,1)a1,a2,a3,a4,a5
 3    FORMAT(1X,2F10.4)
      WRITE(*,3)ALEMDA,RATE
      WRITE(*,3)PNOT,PONE
      SNOT=AKE(1)
      SRNOT=AKE(2)
      REJNOT=AKE(3)
      WRITE(*,5)SNOT,SRNOT,REJNOT
 5    FORMAT(1X,3F10.4)
      POWER=ALEMDA*SRNOT/RATE
      PPOWER=-POWER
      THEETA=EXP(PPOWER)
      WRITE(*,7)THEETA
 7    FORMAT(1X,F10.6)
      MM=REJNOT
      NT=SNOT-REJNOT+1
      CALL BIN(PONE,MM,NT,CPR,CPL,PI)
      QONE=CPR
      WRITE(*,8)QONE
 8    FORMAT(1X,F10.6)
      MM=REJNOT
      NT=SNOT-REJNOT+1
      CALL BIN(PNOT,MM,NT,CPR,CPL,PI)
      QNOT=CPR
      WRITE(*,8)QNOT
      R=1/QONE+THEETA/(1-THEETA)
```

```
C   IR IS EXPECTED NO OF SAMPLES REQUIED TO DETECT SHIFT
        IR=R+0.5
        WRITE(*,9)IR
  9     FORMAT(1X,I3)
C    COMPUTATION  OF EXPECTED COSTS
        NN=SNOT
        MM=REJNOT+1
        NT=NN-MM+1
        CALL BIN(PNOT,MM,NT,CPR,CPA,PI)
        PI0=PI
        ASN0=(1-QNOT)*(PNOT*(SNOT+1)-REJNOT)/(PNOT*(1-PNOT))+
     1  REJNOT*(1-PI0)/PNOT
        NN=SNOT
        MM=REJNOT+1
        NT=NN-MM+1
        CALL BIN(PONE,MM,NT,CPR,CPA,PI)
        PI1=PI
        ASN1=(1-QONE)*(PONE*(SNOT+1)-REJNOT)/(PONE*(1-PONE))+
     1  REJNOT*(1-PI1)/PONE
        WRITE(*,40) ASN0,ASN1
   40   FORMAT(1X,'ASN0=',F10.4,'ASN1=',F10.4)
        ECT1=(A1+A2*ASN0)*THEETA/(1-THEETA)+(A1+A2*ASN1)/QONE
        EC1=ECT1/(IR*SRNOT)
        EC2=A3*(QNOT*THEETA/(1-THEETA)+1)/(IR*SRNOT)
        ADALTA=(1-(1+POWER)*THEETA)/(POWER*(1-THEETA))
        WRITE(*,35)ADALTA
  35    FORMAT(1X,'ADALTA=', F10.6)
        D=THEETA*SRNOT*PNOT/(1-THEETA)+ADALTA*SRNOT*PNOT+SRNOT*PONE/
     1  QONE-ADALTA*SRNOT*PONE
        DS=THEETA*SNOT*PNOT/(1-THEETA)+SNOT*PONE/QONE
        WRITE(*,15)D,DS
 15     FORMAT(1X,2F10.4)
C    DS GIVES   EXPECTED  NO OF   DEFECTIVES  DETECTED   IN   SAMPLING
C    D  GIVES   EXPECTED  NO   OF  DEFECTIVES PRODUCED
        EC3=(A4*DS+A5*(D-DS))/(IR*SRNOT)
        TC=EC1+EC2+EC3
        SUMN=TC
        WRITE (*,30)TC,EC1,EC2,EC3
  30    FORMAT(1X,'TC=',F10.6,'EC1=',F10.6,'EC2=',F10.6,'EC3=',F10.6)
        RETURN
        END


C    FILE NAME IS DSP.FOR
C    PROGRAM FOR E(C) OF MONTGOMERY MODEL FOR DOUBLE
C    SAMPLING POLICY
        SUBROUTINE OBJ7 (AKE,NSTAGE,SUMN,A1,A2,A3,A4,ALEMDA,RATE,
     1  PIE,NSTAT,PIN,NA1,NA2)
        DIMENSION PIN(10),PZ(10),P(10,10),QR(10),ZP(10),BZ(10),ZB(10)
     1  ,B(10,10),BST(10,10),CZ(10),ZC(10),C(10,10),D(10,10),
     1  DST(10,10),ALPHA(10),GAMMA(10),A(10,10),BSTZ(10),T(10,10),ZBST
     1  (10),S(10,10),U(10,10),V(10,10),BB(10,10),AKE(5),ASN(10)
        WRITE(*,5)
 5       FORMAT(4X,'COST COEFFICIENTS')
        WRITE(*,1)A1,A2,A3,A4
 1      FORMAT(1X,4F10.4)
```

```
       WRITE(*,3)ALEMBDA,RATE,PIE,NSTAT
3      FORMAT(1X,3F12.4,I3)
       WRITE(*,6)(PIN(I),I=1,NSTAT)
6       FORMAT(1X,7F8.4)
       N1 = AKE(1)
       N2 = AKE(2)
       SRNOT = AKE(3)
       WRITE(*,2) N1,N2,NA1,NA2,SRNOT
2       FORMAT(1X,4I3,F6.2)
       POWER = ALEMDA*SRNOT/RATE
       PPOWER =-POWER
       PZZ=EXP(PPOWER)
       WRITE(*,7) PZZ
7      FORMAT(1X,F10.4)
       NSTATE = NSTAT-1
       DENO=1.-(1.-PIE)**NSTATE
       MSNOT = SNOT
       DO 10 J=1,NSTATE
       M1 = J+1
       M2 = NSTATE-J
       CALL BIN(PIE,M1,M2,CPR,CPL,PI)
       WRITE(*,8) CPR,CPL,PI,J
8      FORMAT(1X,'CPR=',F10.6,'CPL=',F10.6,'PI=',F10.6,'J=',I2)
10     PZ(J) = PI*(1.-PZZ)/DENO
       DO 600 I = 1,NSTATE
 600    ZP(I)=0.
       DO 20 I=1,NSTATE
       DO 20 J=1,NSTATE
       IF(I-J)30,31,32
  30   P(I,J) = PZ(J)/(1.-PZZ)
       GO TO 20
  31   SPZ=0.
       DO 40 KK=1,I
  40   SPZ = SPZ+PZ(KK)
       P(I,J) = SPZ/(1.-PZZ)
       GO TO 20
  32   P(I,J)=0.
  20  CONTINUE
       T(1,1)=PZZ
       DO 12 I =2,NSTAT
       K=I-1
  12   T(1,I) = PZ(K)
       DO 13 J=2,NSTAT
       K=J-1
  13   T(J,1)=ZP(K)
       DO 14 I=2,NSTAT
       K=I-1
       DO 14 J=2,NSTAT
       K1 = J-1
 14    T(I,J) = P(K,K1)
       CALL PROBR(N1,N2,NA1,NA2,PIN,NSTAT,QR,ASN)
       WRITE(*,301)(QR(I),I = 1,NSTAT)
301    FORMAT(1X,7F10.6)
       DO 60 I=2,NSTAT
       DO 60 J=1,NSTAT
       IF(I-J)61,62,63
 61      S(I,J)=QR(I)*T(1,J)+(1-QR(I))*T(I,J)
```

```
      GO TO 60
 62   S(I,J)=QR(I)*T(1,I)+(1-QR(I))*T(I,I)
      GO TO 60
 63   S(I,J)=QR(I)*T(1,J)
 60   CONTINUE
      DO 326 J=1,NSTAT
326   S(1,J)=T(1,J)
      WRITE(*,302)
302   FORMAT(1X,'MATRIX S(I,J)')
      WRITE(*,303) ((S(I,J),J=1,NSTAT),I=1,NSTAT)
303      FORMAT(1X,7F10.6)
      DO 330 I=1,NSTAT
      DO 330 J=1,NSTAT
      IF(I-J)331,332,331
331      U(I,J)=S(I,J)
      GO TO 330
332      U(I,J)=S(I,J)-1
330      CONTINUE
      WRITE(*,311)
311      FORMAT(1X,'MATRIX U(I,J)')
      WRITE(*,312)((U(I,J),J=1,NSTAT),I=1,NSTAT)
312    FORMAT(1X,7F10.6)
      DO 321 I=1,NSTAT
      DO 321 J=1,NSTATE
321     V(I,J)=U(I,J+1)
      DO 322 I=1,NSTAT
322     V(I,7)=1
      WRITE(*,323)
323     FORMAT(1X,'MATRIX V(I,J)')
      WRITE(*,324) ((V(I,J),J=1,NSTAT),I=1,NSTAT)
324   FORMAT(1X,7F10.6)
      DO 325 I=1,NSTAT
      DO 325 J=1,NSTAT
325      A(I,J)=V(I,J)
      N=NSTAT
      CALL INVRS(A,BB,N)
      WRITE(*,97)
97    FORMAT(1X,'INVERSE MATRIX')
      WRITE(*,98) ((BB(I,J),J=1,NSTAT),I=1,NSTAT)
98    FORMAT(1X,7F10.6)
C   BB(I,J) IS INVERSE OF A(I,J)
      DO 81 J=1,NSTAT
81    ALPHA(J)=BB(NSTAT,J)
      WRITE(*,150)
150     FORMAT(1X,'VECTOR ALPHA')
      WRITE(*,82) (ALPHA(J),J=1,NSTAT)
82    FORMAT(1X,7F10.6)
C COMPUTATION OF GAMMA
      ADALTA=(1.-(1.+POWER)*PZZ)/(POWER*(1.-PZZ))
      WRITE(*,160)
160       FORMAT(1X,'ADALTA')
      WRITE(*,82) ADALTA
      GAMMAZ=ALPHA(1)*PZZ+ALPHA(1)*ADALTA*(1-PZZ)
      WRITE(*,170)
 170     FORMAT(1X,'GAMMAZ')
      WRITE(*,82) GAMMAZ
      DO 90 I=2,NSTAT
```

```
        I3=I-2
        TERM3=0
        TERM4=0
        I1=I-1
        I2=I+1
        IF(I1-1) 101,102,101
 101       DO 100 J=1,I3
        K=J+1
 100     TERM3=TERM3+ALPHA(K)*P(J,I1)
        IF(I1-6) 102,104,102
 102      DO 110 K=I,NSTATE
 110     TERM4=TERM4+P(I1,K)
 104      GAMMA(I1)=ALPHA(I)*P(I1,I1)+(1.-ADALTA)*ALPHA(1)*PZ(I1)+
     1  (1.-ADALTA)*TERM3+ALPHA(I)*TERM4*ADALTA
        WRITE(*,82)GAMMA(I1)
 90     CONTINUE
C   COMPUTATION OF EXPECTATIONS
        TERM7=0
        DO 429 I=1,NSTAT
 429    TERM7=TERM7+ALPHA(I)*ASN(I)
        EC1=(A1+A2*TERM7)/SRNOT
        TERM5=0
        DO 120 I=1,NSTAT
 120     TERM5=TERM5+QR(I)*ALPHA(I)
        EC2=A3*TERM5/SRNOT
        TERM6=0
        TERM6=TERM6+PIN(1)*GAMMAZ
        DO 130 I=2,NSTAT
        J=I-1
 130     TERM6=TERM6+PIN(I)*GAMMA(J)
        EC3=A4*TERM6
        TC=EC1+EC2+EC3
        SUMN=TC
        WRITE(*,140)TC,EC1,EC2,EC3
 140     FORMAT(1X,'TOTAL COST=',E18.8,'EC1=',E18.8,'EC2=',E18.8,
     1  'EC3=',E18.8)
        RETURN
        END


C  FILE NAME IS DSP1
C  PROCRAM FOR COMPUTATION OF POWER AND ASN FOR DOUBLE SAMPLING
        SUBROUTINE  PROBR(N1,N2,NA1,NA2,PIN,NSTAT,QR,ASN)
        DIMENSION PIN(10),QR(10),ASN(10)
        DO 50 I=1,NSTAT
        PROB=PIN(I)
        L1=NA1+1
        L2=NA2+1
        SUM=0
        DO 51 ND=L1,NA2
        M1=L2-ND1
        NT1=N2-M1+1
        CALL BIN(PROB,M1,NT1,PR1,PA1,P)
        M2=ND1+1
        NT2=N1-M2+1
        CALL BIN(PROB,M2,NT2,PR2,PA2,PI)
```

```
      SR=PR1*PI
      SUM=SUM+SR
 51   CONTINUE
      M5=L2
      NT5=N1-M5+1
      CALL BIN(PROB,M5,NT5,PR5,PA5,P3)
      QR(I)=PR5+SUM
 50   CONTINUE
      WRITE(*,70) (Q(I),I=1,NSTAT)
 70   FORMAT(1X,7F10.6)
C   COMPUTATION OF AVERAGE SAMPLE NUMBER .
      DO 80 I=1,NSTAT
      PROB=PIN(I)
      M3=L1
      NT3=N1-L1+1
      CALL BIN(PROB,M3,NT3,PR3,PA3,P1)
      M4=L2
      NT4=N1-L2+1
      CALL BIN(PROB,M4,NT4,PR4,PA4,P2)
      T=PR3-PR4
      ASN(I)=N1+(N2*T)
 80   CONTINUE
      WRITE(*,70) (ASN(I),I=1,NSTAT)
      RETURN
      END
```

```
C   FILE NAME IS FCSP
C   PROGRAM FOR E(C) OF MONTGOMERYS MODEL USING FULLY
C   CURTAILED SAMPLING POLICY
      SUBROUTINE OBJ7 (AKE,NSTAGE,SUMN,A1,A2,A3,A4,ALEMDA,RATE,
     1 PIE,NSTAT,PIN)
      DIMENSION PIN(10),PZ(10),P(10,10),QR(10),ZP(10),BZ(10),ZB(10)
     1 ,B(10,10),BST(10,10),CZ(10),ZC(10),C(10,10),D(10,10),
     1 DST(10,10),ALPHA(10),GAMMA(10),A(10,10),BSTZ(10),T(10,10),ZBST
     1 (10),S(10,10),U(10,10),V(10,10),BB(10,10),AKE(5),AASN(10)
      WRITE(*,5)
 5     FORMAT(4X,'COST COEFFICIENTS')
      WRITE(*,1)A1,A2,A3,A4
 1    FORMAT(1X,4F10.4)
      WRITE(*,3)ALEMBDA,RATE,PIE,NSTAT
 3    FORMAT(1X,3F12.4,I3)
      WRITE(*,6)(PIN(I),I=1,NSTAT)
 6     FORMAT(1X,7F8.4)
      SNOT = AKE(1)
      SRNOT = AKE(2)
      REJNOT = AKE(3)
      WRITE(*,2)SNOT,SRNOT,REJNOT
 2     FORMAT(1X,'SAMPLESIZE =',F10.2,'INT SAM RANGE =',F10.2,'REJ
     1 NUM=',F10.2)
      POWER = ALEMDA*SRNOT/RATE
      PPOWER =-POWER
      PZZ=EXP(PPOWER)
      WRITE(*,7) PZZ
 7    FORMAT(1X,F10.4)
```

```
       NSTATE = NSTAT-1
       DENO=1.-(1.-PIE)**NSTATE
       MSNOT = SNOT
       DO 10 J=1,NSTATE
       M1 = J+1
       M2 = NSTATE-J
       CALL BIN(PIE,M1,M2,CPR,CPL,PI)
       WRITE(*,8) CPR,CPL,PI,J
8      FORMAT(1X,'CPR=',F10.6,'CPL=',F10.6,'PI=',F10.6,'J=',I2)
10     PZ(J) = PI*(1.-PZZ)/DENO
       DO 600 I = 1,NSTATE
 600    ZP(I)=0.
       DO 20 I=1,NSTATE
       DO 20 J=1,NSTATE
       IF(I-J)30,31,32
  30   P(I,J) = PZ(J)/(1.-PZZ)
       GO TO 20
  31   SPZ=0.
       DO 40 KK=1,I
  40   SPZ = SPZ+PZ(KK)
       P(I,J) = SPZ/(1.-PZZ)
       GO TO 20
  32   P(I,J)=0.
  20  CONTINUE
       T(1,1)=PZZ
       DO 12 I =2,NSTAT
       K=I-1
  12   T(1,I) = PZ(K)
       DO 13 J=2,NSTAT
       K=J-1
  13   T(J,1)=ZP(K)
       DO 14 I=2,NSTAT
       K=I-1
       DO 14 J=2,NSTAT
       K1 = J-1
 14    T(I,J) = P(K,K1)
       WRITE(*,15)
 15    FORMAT(1X,'TRANSITION MATRIX')
       WRITE(*,11)((T(I,J),J = 1,NSTAT),I=1,NSTAT)
 11    FORMAT(1X,7F10.6)
       S1 = SNOT
       S2 = SRNOT
       S3 = REJNOT
       CALL PROBR(S1,S2,S3,PIN,NSTAT,QR)
       WRITE(*,301)(QR(I),I = 1,NSTAT)
301    FORMAT(1X,7F10.6)
       DO 60 I=2,NSTAT
       DO 60 J=1,NSTAT
       IF(I-J)61,62,63
 61      S(I,J)=QR(I)*T(1,J)+(1-QR(I))*T(I,J)
       GO TO 60
 62    S(I,J)=QR(I)*T(1,I)+(1-QR(I))*T(I,I)
       GO TO 60
 63    S(I,J)=QR(I)*T(1,J)
 60    CONTINUE
       DO 326 J=1,NSTAT
326    S(1,J)=T(1,J)
```

```
        WRITE(*,302)
 302    FORMAT(1X,'MATRIX S(I,J)')
        WRITE(*,303) ((S(I,J),J=1,NSTAT),I=1,NSTAT)
 303       FORMAT(1X,7F10.6)
        DO 330 I=1,NSTAT
        DO 330 J=1,NSTAT
        IF(I-J)331,332,331
 331       U(I,J)=S(I,J)
        GO TO 330
 332       U(I,J)=S(I,J)-1
 330       CONTINUE
        WRITE(*,311)
 311       FORMAT(1X,'MATRIX U(I,J)')
        WRITE(*,312)((U(I,J),J=1,NSTAT),I=1,NSTAT)
 312     FORMAT(1X,7F10.6)
        DO 321 I=1,NSTAT
        DO 321 J=1,NSTATE
 321      V(I,J)=U(I,J+1)
        DO 322 I=1,NSTAT
 322      V(I,7)=1
        WRITE(*,323)
 323      FORMAT(1X,'MATRIX V(I,J)')
        WRITE(*,324) ((V(I,J),J=1,NSTAT),I=1,NSTAT)
 324    FORMAT(1X,7F10.6)
        DO 325 I=1,NSTAT
        DO 325 J=1,NSTAT
 325       A(I,J)=V(I,J)
        N=NSTAT
        CALL INVRS(A,BB,N)
        WRITE(*,97)
 97     FORMAT(1X,'INVERSE MATRIX')
        WRITE(*,98) ((BB(I,J),J=1,NSTAT),I=1,NSTAT)
 98     FORMAT(1X,7F10.6)
C   BB(I,J) IS INVERSE OF A(I,J)
        DO 81 J=1,NSTAT
 81     ALPHA(J)=BB(NSTAT,J)
        WRITE(*,150)
 150      FORMAT(1X,'VECTOR ALPHA')
        WRITE(*,82) (ALPHA(J),J=1,NSTAT)
 82     FORMAT(1X,7F10.6)
C COMPUTATION OF GAMMA
        ADALTA=(1.-(1.+POWER)*PZZ)/(POWER*(1.-PZZ))
        WRITE(*,160)
 160        FORMAT(1X,'ADALTA')
        WRITE(*,82) ADALTA
        GAMMAZ=ALPHA(1)*PZZ+ALPHA(1)*ADALTA*(1-PZZ)
        WRITE(*,170)
  170      FORMAT(1X,'GAMMAZ')
        WRITE(*,82) GAMMAZ
        DO 90 I=2,NSTAT
        I3=I-2
        TERM3=0
        TERM4=0
        I1=I-1
        I2=I+1
        IF(I1-1) 101,102,101
 101       DO 100 J=1,I3
```

```
        K=J+1
 100     TERM3=TERM3+ALPHA(K)*P(J,I1)
        IF(I1-6) 102,104,102
 102      DO 110 K=I,NSTATE
 110     TERM4=TERM4+P(I1,K)
 104      GAMMA(I1)=ALPHA(I)*P(I1,I1)+(1.-ADALTA)*ALPHA(1)*PZ(I1)+
     1  (1.-ADALTA)*TERM3+ALPHA(I)*TERM4*ADALTA
        WRITE(*,82)GAMMA(I1)
 90     CONTINUE
        CALL NBAR(S1,S3,PIN,NSTAT,QR,AASN)
        WRITE(*,301) (AASN(I),I=1,NSTAT)
C   COMPUTATION OF EXPECTATION
        TERM7=0
        DO 429 I=1,NSTAT
 429    TERM7=TERM7+ALPHA(I)*AASN(I)
        EC1=(A1+A2*TERM7)/SRNOT
        TERM5=0
        DO 120 I=1,NSTAT
 120     TERM5=TERM5+QR(I)*ALPHA(I)
        EC2=A3*TERM5/SRNOT
        TERM6=0
        TERM6=TERM6+PIN(1)*GAMMAZ
        DO 130 I=2,NSTAT
        J=I-1
 130     TERM6=TERM6+PIN(I)*GAMMA(J)
        EC3=A4*TERM6
        TC=EC1+EC2+EC3
        SUMN=TC
        WRITE(*,140)TC,EC1,EC2,EC3
 140     FORMAT(1X,'TOTAL COST=',E18.8,'EC1=',E18.8,'EC2=',E18.8,
     1  'EC3=',E18.8)
        RETURN
        END


C    FILE NAME IS FCSP1
C    PROGRAM FOR ASN OF FULLY-CURTAILED  SAMPLING POLICY
        SUBROUTINE  NBAR(S1,S3,PIN,NSTAT,QR,AASN)
        DIMENSION  PIN(10) ,QR(10),AASN(10)
        PROB=PIN(I)
        MM=S3+1
        NN=S1
        NT=NN-MM+1
        CALL BIN(PROB,MM,NT,PR,PA,PI)
        WRITE(*,3) PI
  3     FORMAT(1X,F10.6)
        S7=S1+1
        AASN(I)=(PIN(I)*S7-S3)*(1-QR(I))/(PIN(I)*(1-PIN(I)))+(S3*(1-PI))
     1  /PIN(I)
        WRITE(*,7) (AASN(I),I=1,NSTAT)
  7     FORMAT(1X,7F10.6)
  50    CONTINUE
        RETURN
        END
```

```
C   FILE NAME IS FCSP2.FOR
C   PROGRAM FOR PROBABILITY OF REJECTION FOR np-CHART USING
C   FULLT-CURTAILED SAMPLING
        SUBROUTINE PROBR(T1,T2,T3,PIN,NST,SR)
        DIMENSION PIN(10),SR(10),QR(10)
        DO 50 I=1,NST
        PROB=PIN(I)
        MM=T3
        NT=T1-T3+1
        CALL BIN(PROB,MM,NT,PC1,PC2,PIND)
 50     SR(I)=PC1
        RETURN
        END
```