Luciana Marcelino de Oliveira

# Uma análise do procedimento de controle on-line de qualidade de Taguchi baseado nos resultados de uma seqüência de inspeções

Dissertação de Mestrado apresentada ao Departamento de Estatística do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais, como requisito parcial para obtenção do título de Mestre em Estatística.

Orientador: Prof. Roberto da Costa Quinino

Universidade Federal de Minas Gerais
Belo Horizonte, Fevereiro de 2012

Dedico este trabalho a minha família e amigos.

# Agradecimentos

Agradeço primeiramente a Deus pela presença constante em minha vida, por me dar força e sabedoria para vencer esse grande desafio.

Aos meus pais pela educação que me deram, pelo amor incondicional.

Ao meu querido amor Lipe pelo amor, compreensão, incentivo e companheirismo nos momentos tristes e alegres durante a jornada do curso.

Ao professor Roberto da Costa Quinino, pela confiança, paciência e dedicação dispensados.

Aos meus amigos e em especial ao Ronaldo e Katia, por me ajudarem nos momentos difíceis e compartilhar alegrias.

# Resumo

Neste trabalho, um procedimento para o controle on-line de processos de variáveis é proposto. Este procedimento consiste em inspecionar o m-ésimo item a cada m itens produzidos e decidir, a cada inspeção, se o processo está fora de controle. Dois conjuntos de limites, alerta ( $\mu_0 \pm W$ ) e controle ( $\mu_0 \pm C$ ), são utilizados. Se o valor da estatística monitorada está além dos limites de controle ou se uma sequência de *h* observações consecutivas está entre os limites de alerta e os limites de controle, a produção é parada para ajuste, caso contrário, a produção continua. As propriedades de uma cadeia de Markov ergódica são usadas para obter uma expressão para o custo médio por item. Os parâmetros (o intervalo de amostragem *m*, a largura do alarme W, os limites de controle *C, com* $W \leq C$, e o tamanho da sequência (*h*)) são otimizados minimizando a função custo. Um exemplo numérico ilustra o procedimento proposto.

**Palavras chaves:** Controle on-line de processos; Cadeia de Markov; Seqüência de inspeções; Intervalo de amostragem variável

# Abstract

In this paper, a procedure for the on-line process control of variables is proposed. This procedure consists of inspecting the *m*-th item from every *m* produced items and deciding, at each inspection, whether the process is out-of-control. Two sets of limits, warning ($\mu_0 \pm W$) and control ($\mu_0 \pm C$), are used. If the value of the monitored statistic falls beyond the control limits or if a sequence of *h* observations falls between the warning limits and the control limits, the production is stopped for adjustment; otherwise, production goes on. The properties of an ergodic Markov chain are used to obtain an expression for the average cost per item. The parameters (the sampling interval *m*, the widths of the warning W, the control limits C ($W \leq C$), and the sequence length (*h*) are optimized by minimizing the cost function. A numerical example illustrates the proposed procedure.

**Keywords:** On-line process control; Markov chain, Sequence of inspections, Variable interval sampling

# Índice

# 1. Introdução

Taguchi et al. (1989) apresenta um sistema de controle on-line onde considera um processo que inicia sua operação produzindo itens com uma fração conforme $p_1 = 1$ (estado I). Após a ocorrência de uma causa especial essa fração passa a um valor $p_2$ ($0 \leq p_2 \leq p_1$) e o processo permanece produzindo nesta condição (estado II) até que esta mudança seja detectada e a causa especial removida. O sistema de controle consiste em inspecionar um item a cada m produzidos e, uma vez que este item seja julgado como não conforme, admite-se que ocorreu uma mudança da fração de itens conformes e o processo é parado para ajuste. Após esse ajuste, a fração de itens conformes retorna ao valor inicial $p_1=1$.

Nayebpour & Woodall (1993) apresentam um modelo de controle por atributos incluindo um mecanismo de falha para o processo (através de uma distribuição geométrica), o que representa uma maneira formal do processo mudar do estado I para o estado II. Com isso, afirmam que não pode ser obtida uma expressão analítica para intervalo ótimo entre inspeções e a busca deste parâmetro exige o uso de métodos numéricos.

No modelo de controle para atributos apresentado por Borges *et al.* (2001) admite-se que o sistema de inspeção está sujeito a erros de classificação. Pode-se julgar um item conforme dado que ele não o é (erro tipo II) ou julgar um item conforme como não conforme (erro tipo I). Suas conclusões indicam que o custo do sistema de controle é sensível à presença dos erros de classificação. Porém, nos três artigos mencionados não é considerada a hipótese de classificações repetidas do item inspecionado, o que pode reduzir o impacto dos erros de classificação e, conseqüentemente, o custo médio do sistema de controle por item produzido.

Alguns autores já propuseram o uso de classificações repetidas em um sistema de inspeção com erros de classificação, empregando critérios diferenciados na classificação final do item inspecionado (o item examinado é classificado repetidamente e independentemente *r* vezes e em cada classificação, ele é avaliado conforme ou não conforme). Em Greenberg & Stokes (1995) este procedimento é proposto, mas admite-se apenas a existência do erro do tipo I na determinação do número ótimo de classificações repetidas. Em Quinino & Suyama (2002), *r* classificações repetidas são realizadas no item inspecionado, que é declarado conforme se pelo menos forem observadas *a* classificações conformes. Os parâmetros *a* e *r* são determinados segundo uma abordagem econômica. Em Quinino & Ho (2004), os itens são examinados repetidamente até observar *a* classificações conformes ou *b* classificações não conformes (no primeiro caso, o item é declarado conforme e no segundo , o item é declarado não conforme). O objetivo foi determinar os parâmetros *a* e *b* ótimos também segundo uma abordagem econômica. Porém, tais trabalhos não consideram a mudança da fração de itens conformes durante a produção.

Trindade et al (2007) desenvolveram um modelo probabilístico que considera classificações repetidas e independentes do item inspecionado. Buscando uma minimização do custo médio do sistema utilizando as propriedades de uma cadeia de Markov de estados discretos, é possível determinar uma estratégia ótima de controle on-line por atributos em um processo cujo sistema de inspeção está sujeito a erros de classificação. Essa estratégia consiste na determinação do intervalo entre inspeções (m), o número de classificações repetidas (r) e o número mínimo de classificações conformes (w), dentre as classificações repetidas, para julgar um item como conforme. Uma extensão dessa abordagem para o caso de variáveis pode ser vista em Ho et al (2007).

O objetivo deste trabalho é apresentar um procedimento para controle on-line de processo, adicionando critérios alternativos que levam a uma interrupção do processo de produção não incluídos na lista dos trabalhos anteriores. Como na abordagem de Taguchi, o m-ésimo item é inspecionado a cada m itens produzidos. Se o valor da estatística monitorada está fora dos limites de controle (zona vermelha - RZ) ou uma seqüência de h observações está entre os limites de alarme e os limites de controle (zona amarela - YZ), então a produção é interrompida para ajuste, caso contrário, (zona verde - GZ) a produção continua.

Nesta nova proposta consideramos dois tipos de modelo. Para isso definimos agora a probabilidade de eficiência do ajuste. Caso o sistema necessite de ajuste, este pode ser eficiente ou não, ou seja, se o ajuste for eficiente então o processo volta a produzir no Estado I, caso contrário continuará a produzir no Estado II.

No primeiro modelo, o ajuste torna o sistema de produção tão bom quanto um novo, ou seja, a probabilidade de eficiência do ajuste pe = 1, e no segundo modelo isso não acontece, a probabilidade de eficiência pode variar no intervalo $0.5 < pe < 1.0$ , tornando pe em um parâmetro de otimização.

Os parâmetros *h*, *C*, *W*, *m e pe* são determinados tal que minimize o custo médio por item do sistema controlado.

A Figura 1 representa esquematicamente a nova proposta.

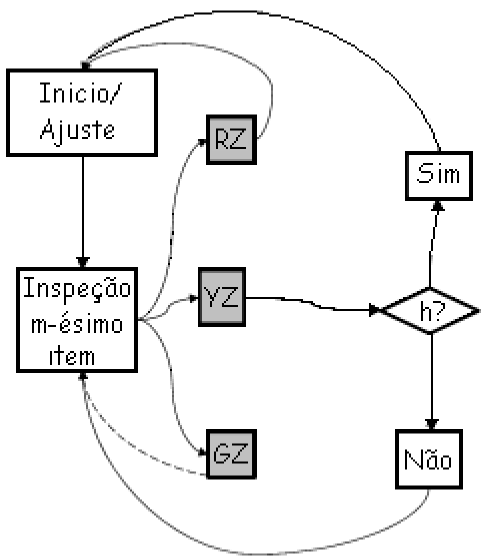


Figura 1 – Representação esquemática do sistema de inspeção

Este trabalho amplia o procedimento proposto por Ho et al. (2007). Sua contribuição pode ser vista como um caso específico da nova proposta, quando h = 1 e W = C. A generalização proposta no trabalho atual, $h > 1$e $W \leq C$, não constitui uma simples expansão metodológica da proposta de Ho et al (2007). Um novo desenvolvimento, com base em uma cadeia de Markov, é necessário para facilitar os avanços e para proporcionar uma melhor compreensão para o leitor.

# 2. Modelos Probabilísticos

Considere um processo em que itens são produzidos e a característica de qualidade de interesse (X) pode ser descrita por uma distribuição normal com média μ e desvio padrão σ. O processo inicia sob controle, em outras palavras, a média do processo é $\mu = \mu_0$ (Estado I), e o processo pode mudar para $\mu = \mu_1$, $\mu_1 \neq \mu_0$ (Estado II). A duração do processo sob controle é normalmente modelada por uma distribuição exponencial para um caso contínuo. A distribuição geométrica se comporta de forma semelhante à distribuição exponencial, mas é geralmente usada para o caso discreto no qual a duração é medida pelo número de unidades que são produzidos antes da mudança. Assim, como em trabalhos anteriores [Nayebpour e Woodall (1993), Nandi e Sreehari (1999), Jiang e Tsui (2000), Borges et al. (2001), Ho et al. (2007), Trindade et al. (2007), Dasgupta e Mandal (2008) e Ding e Gong (2008)], este estudo se baseia em uma distribuição geométrica com parâmetro π, $0 < \pi < 1$, para descrever a mudança aleatória de Estado I para o Estado II. Supõe-se que a probabilidade de mudança do Estado II para Estado I é zero, sem qualquer intervenção no processo.

Ambas as distribuições, geométrica e exponencial, não possuem memória, o que facilita a análise matemática. No entanto, essas distribuições são úteis por outros motivos além de suas facilidades matemáticas. Muitos pesquisadores têm utilizado recentemente as distribuições exponencial ou geométrica, para descrever as mudanças de estado sob-controle para fora de controle. Por exemplo, Wang e Sheu (2003) utilizam a distribuição exponencial, enquanto Ho et al. (2007), Trindade et al. (2007), Dasgupta (2008) e Ding & Gong (2008) utilizam a distribuição geométrica. Tais distribuições não facilitam somente o desenvolvimento de modelos matemáticos, mas também permitem a aplicação a problemas reais, como mencionado nos trabalhos citados.

Três linhas horizontais são utilizadas para construir um gráfico de controle padrão: uma em $\mu = \mu_0$, e outras duas em $\mu_0 \pm C$ (limites de controle superior e inferior (LC)). No entanto, vamos utilizar dois conjuntos de limites; os limites externos são os limites de ação usual (LC), e os limites internos são traçados em $\mu_0 \pm W$, $W \leq C$. Para ilustrar esses limites (ver figura 2), utilizaremos cores diferentes para identificar as diferentes zonas. Pontos plotados fora dos limites de controle (zona vermelha) levam a uma parada por uma causa atribuível; pontos que se situam entre os limites de alarme e os limites de controle (zona amarela) podem revelar que o processo pode não estar operando adequadamente, caso contrário, o processo deve operar de forma adequada se os pontos estiverem entre os limites de alarme (zona verde).

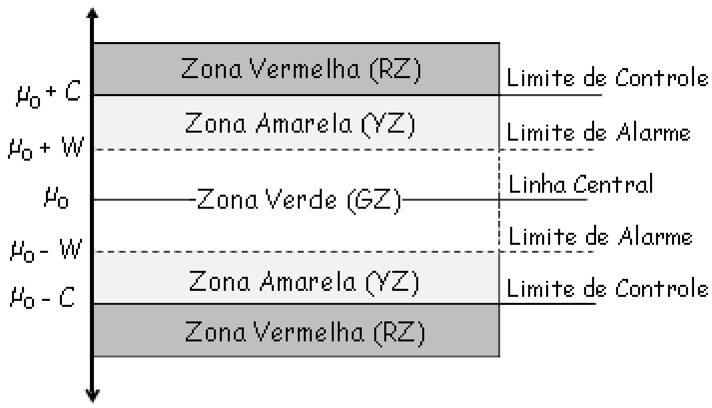


Figura 2 – Limites de Alarme e Controle

Seja $X_i$, $i = 1,2,3,...$ , o valor observado de alguma característica de interesse na i-ésima inspeção. O sistema de inspeção pode ser descrito por uma cadeia de Markov com um conjunto de estados (*s*,*k*). O primeiro índice indica em qual estado [Estado I: sob-controle (s = 0) ou Estado II: fora-de-controle (s > 0)] o processo está no momento da inspeção e assume três valores: s = 0, o item inspecionado é produzido com $\mu = \mu_0$; s = 1, o item inspecionado é produzido em $\mu = \mu_1$, mas a média do processo mudou de

$\mu = \mu_0$ para $\mu = \mu_1$, no ciclo da inspeção atual (assim, uma parte dos itens é produzida com $\mu = \mu_0$, e pelo menos o item inspecionado é produzido com $\mu = \mu_1$); s = 2, a média mudou para $\mu = \mu_1$, e todos os itens do ciclo atual são produzidos com $\mu = \mu_1$.

O segundo índice k está relacionado com o resultado da inspeção:

- $k = -1$, o valor observado da inspeção atual está na zona vermelha ($X_i > \mu_0 + C$ ou $X_i < \mu_0 - C$); e o processo é parado para ajuste;
- $k = 0$, o valor observado da inspeção atual está na zona verde ($\mu_0 - W < X_i < \mu_0 + W$); e o processo não é ajustado;
- $k = 1$, o valor observado da inspeção atual está na zona amarela ($\mu_0 + W < X_i < \mu_0 + C$ ou $\mu_0 - C < X_i < \mu_0 - W$), mas o valor observado da inspeção anterior não estava na zona amarela; e o processo não é ajustado;
- $k = 2$, o valor observado da inspeção atual está na zona amarela ($\mu_0 + W < X_i < \mu_0 + C$ ou $\mu_0 - C < X_i < \mu_0 - W$), e o valor observado da inspeção anterior também estava na zona amarela; mas o processo não é ajustado;
- $k = 3$, o valor observado da inspeção atual está na zona amarela ($\mu_0 + W < X_i < \mu_0 + C$ ou $\mu_0 - C < X_i < \mu_0 - W$), e os valores observados das duas inspeções anteriores também estavam na zona amarela; o processo não é ajustado;
- ....
- $k = h$, o valor observado da inspeção atual está na zona amarela ($\mu_0 + W < X_i < \mu_0 + C$ ou $\mu_0 - C < X_i < \mu_0 - W$) e os valores observados das (h-1) inspeções anteriores também estavam na zona amarela; o processo então é ajustado.

Note que o ajuste ocorre somente se $k = -1$ ou $k = h$. Um total de *3 x (h+2)* estados da cadeia (*s, k*) é usado para descrever o processo de inspeção e a inspeção ocorre sempre após a produção de m itens. O segundo índice assume valores inteiros no intervalo $-1 \le k \le h$ e $k = t \ge 0$ ,o que indica uma seqüência de t itens inspecionados para o qual os valores observados estão na zona amarela. Antes do modelo probabilístico, vamos apresentar as seguintes notações:

$G_0 = p(\mu_0 - W \le X_i \le \mu_0 + W \mid \mu = \mu_0)$ , é a probabilidade do valor observado da inspeção atual estar na zona verde com média $\mu_0$.

$G_1 = p(\mu_0 - W \le X_i \le \mu_0 + W \mid \mu = \mu_1)$ , é a probabilidade do valor observado da inspeção atual estar na zona verde com média $\mu_1$.

$Y_0 = p(\mu_0 + W \le X_i < \mu_0 + C \mid \mu = \mu_0) + P(\mu_0 - C < X_i \le \mu_0 - W \mid \mu = \mu_0)$ , é a probabilidade do valor observado da inspeção atual estar na zona amarela com média $\mu_0$.

$Y_1 = p(\mu_0 + W \le X_i < \mu_0 + C \mid \mu = \mu_1) + P(\mu_0 - C < X_i \le \mu_0 - W \mid \mu = \mu_1)$ , é a probabilidade do valor observado da inspeção atual estar na zona amarela com média $\mu_1$.

$R_0 = p(X_i > \mu_0 + C \mid \mu = \mu_0) + P(X_i < \mu_0 - C \mid \mu = \mu_0)$ , é a probabilidade do valor observado da inspeção atual estar na zona vermelha com média $\mu_0$.

$R_1 = p(X_i > \mu_0 + C \mid \mu = \mu_1) + P(X_i < \mu_0 - C \mid \mu = \mu_1)$ , é a probabilidade do valor observado da inspeção atual estar na zona vermelha com média $\mu_1$.

## 2.1. Probabilidade de eficiência do ajuste *pe = 1*

O sistema opera produzindo uma fração de itens conforme pe = 1 (sob-controle), após inspecionar um item como não conforme, esta fração diminui a um valor menor que 1 até que seja feito o ajuste do sistema. Após o ajuste a fração de itens conforme retorna ao valor inicial pe = 1, e segue produzindo tão bom quanto um novo.

As probabilidades de transição do estado (*s*,*k*) no momento da inspeção *i* para o estado (*s**,*k**) no momento da inspeção $i+1$, são os elementos da matriz P

$$P = \begin{pmatrix} A_{00} & A_{01} & A_{02} \\ A_{10} & A_{11} & A_{12} \\ A_{20} & A_{21} & A_{22} \end{pmatrix} \tag{1}$$

Onde $\mathbf{A}_{ss*}$ denota a matriz de probabilidade de transição $p_{(s,k)(s*,k*)}$; $s, s* = 0,1,2; k, k* = -1,0,1,...,h$. Como exemplo para ilustrar a notação utilizada, $p_{(0,0)(0,-1)}$ representa a probabilidade de que dado que a inspeção i ocorra sob o estado da cadeia *(0,0)*, a inspeção *(i+1)* ocorrerá sob o estado *(0,-1)*, ou seja, na inspeção *i* o ciclo foi produzido no estado I e o valor observado da peça inspecionada está na região verde; e na inspeção *(i+1)* o ciclo também será produzido no estado I e o valor da peça inspecionada estará na região vermelha.

Os elementos de $\mathbf{A}_{00}$ são as probabilidades de transição dos estados (*0,k*) para os estados (*0,k**). Ou seja, o processo está sob-controle na inspeção i, e também permanece sob controle na inspeção (i+1). Os elementos não-nulos da matriz $A_{00}$ são expressos por:

$$\begin{cases} p_{(0,0)(0,-1)} = p_{(0,-1)(0,-1)} = p_{(0,h)(0,-1)} = p_{(0,k)(0,-1)} = q_m R_0 \\ p_{(0,0)(0,0)} = p_{(0,-1)(0,0)} = p_{(0,h)(0,0)} = p_{(0,k)(0,0)} = q_m G_0 \\ p_{(0,0)(0,1)} = p_{(0,-1)(0,1)} = p_{(0,h)(0,1)} = p_{(0,k)(0,k+1)} = q_m Y_0 \end{cases} \tag{2}$$

Onde $0 < k < h$, com $q_m = (1-\pi)^m$, que representa a probabilidade do processo não mudar de estado.

Os elementos de $A_{01}$ são as probabilidades de transição dos estados (*0,k*) para os estados (*1,k**). Ou seja, o processo está sob controle na inspeção i, mas o parâmetro mudou de $\mu = \mu_0$ para $\mu = \mu_1$ durante algum instante do ciclo de inspeção. Assim, alguns itens são produzidos com $\mu = \mu_0$, e pelo menos o item inspecionado é produzido com $\mu = \mu_1$. Os elementos não nulos da matriz $A_{01}$ são expressos por:

$$\begin{cases} p_{(0,0)(1,-1)} = p_{(0,-1)(1,-1)} = p_{(0,h)(1,-1)} = p_{(0,k)(1,-1)} = (1-q_m) R_1 \\ p_{(0,0)(1,0)} = p_{(0,-1)(1,0)} = p_{(0,h)(1,0)} = p_{(0,k)(1,0)} = (1-q_m) G_1 \\ p_{(0,0)(1,1)} = p_{(0,-1)(1,1)} = p_{(0,h)(1,1)} = p_{(0,k)(1,k+1)} = (1-q_m) Y_1 \end{cases} \tag{3}$$

A matriz $A_{02}$ é uma matriz nula, pois representa as probabilidades de transição quando o processo está no estado I na inspeção *i* ou seja todas as peças do ciclo foram produzidas sob controle; e na inspeção *i+1* o processo estará fora de controle, ou seja todas as peças do ciclo serão produzidas no estado II. Os elementos não nulos de $A_{10}$ são as probabilidades de transição com k = - 1 ou h. Ou seja, o processo é ajustado na inspeção anterior, e ele reinicia sob controle no ciclo atual empregando um intervalo de amostragem m:

$$\begin{cases} p_{(1,-1)(0,-1)} = p_{(2,-1)(0,-1)} = p_{(1,h)(0,-1)} = p_{(2,h)(0,-1)} = q_m R_0 \\ p_{(1,-1)(0,0)} = p_{(2,-1)(0,0)} = p_{(1,h)(0,0)} = p_{(2,h)(0,0)} = q_m G_0 \\ p_{(1,-1)(0,1)} = p_{(2,-1)(0,1)} = p_{(1,h)(0,1)} = p_{(2,h)(0,1)} = q_m Y_0 \end{cases} \qquad (4)$$

Similarmente tem-se que $A_{10}=A_{20}$.

Da mesma forma, os elementos não-nulos de $A_{11}$ são também as probabilidades de transição, com k = -1 ou h. No entanto, após o ajuste, o processo reinicia sob controle, mas o parâmetro $\mu$ muda na inspeção atual. Assim, os elementos não-nulos destas matrizes são

$$\begin{cases} p_{(1,-1)(1,-1)} = p_{(2,-1)(1,-1)} = p_{(1,h)(1,-1)} = p_{(2,h)(1,-1)} = (1-q_m)R_1 \\ p_{(1,-1)(1,0)} = p_{(2,-1)(1,0)} = p_{(1,h)(1,0)} = p_{(2,h)(1,0)} = (1-q_m)G_1 \\ p_{(1,-1)(1,1)} = p_{(2,-1)(1,1)} = p_{(1,h)(1,1)} = p_{(2,h)(1,1)} = (1-q_m)Y_1 \end{cases} \qquad (5)$$

Similarmente tem-se que $A_{11}=A_{21}$.

Em seguida, temos as matrizes $A_{12}$ e $A_{22}$, que são as probabilidades de transição dos estados (*1*, *k*) para os estados (*2*, *k* *) e dos estados (*2*, *k*) para os estados (*2*, *k**), respectivamente. Nestes casos, o parâmetro mudou nas inspeções anteriores, e o processo não foi ajustado. $A_{12} = A_{22}$ e as probabilidades não-nulas dessas matrizes são

$$\begin{cases} p_{(1,k)(2,-1)} = p_{(2,k)(2,-1)} = R_1 \\ p_{(1,k)(2,0)} = p_{(2,k)(2,0)} = G_1 \\ p_{(1,k)(2,k+1)} = p_{(2,k)(2,k+1)} = Y_1 \end{cases} \qquad (6)$$

para $0 < k < h$.

Para ilustrar as probabilidades descritas acima, consideramos o exemplo onde *h = 2*. A matriz de probabilidade de transição será:

$$P = \begin{array}{r} 0-1 \\ 0,0 \\ 0,1 \\ 0,2 \\ 1,-1 \\ 1,0 \\ 1,1 \\ 1,2 \\ 2,-1 \\ 2,0 \\ 2,1 \\ 2,2 \end{array} \begin{pmatrix} q_m R_0 & q_m G_0 & q_m Y_0 & 0 & (1-q_m)R_1 & (1-q_m)G_1 & (1-q_m)Y_1 & 0 & 0 & 0 & 0 & 0 \\ q_m R_0 & q_m G_0 & q_m Y_0 & 0 & (1-q_m)R_1 & (1-q_m)G_1 & (1-q_m)Y_1 & 0 & 0 & 0 & 0 & 0 \\ q_m R_0 & q_m G_0 & 0 & q_m Y_0 & (1-q_m)R_1 & (1-q_m)G_1 & 0 & (1-q_m)Y_1 & 0 & 0 & 0 & 0 \\ q_m R_0 & q_m G_0 & q_m Y_0 & 0 & (1-q_m)R_1 & (1-q_m)G_1 & (1-q_m)Y_1 & 0 & 0 & 0 & 0 & 0 \\ q_m R_0 & q_m G_0 & q_m Y_0 & 0 & (1-q_m)R_1 & (1-q_m)G_1 & (1-q_m)Y_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & R_1 & G_1 & Y_1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & R_1 & G_1 & 0 & Y_1 \\ q_m R_0 & q_m G_0 & q_m Y_0 & 0 & (1-q_m)R_1 & (1-q_m)G_1 & (1-q_m)Y_1 & 0 & 0 & 0 & 0 & 0 \\ q_m R_0 & q_m G_0 & q_m Y_0 & 0 & (1-q_m)R_1 & (1-q_m)G_1 & (1-q_m)Y_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & R_1 & G_1 & Y_1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & R_1 & G_1 & 0 & Y_1 \\ q_m R_0 & q_m G_0 & q_m Y_0 & 0 & (1-q_m)R_1 & (1-q_m)G_1 & (1-q_m)Y_1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

## 2.2. Probabilidade de eficiência do ajuste *0,5 < pe < 1*

Neste modelo consideramos que o ajuste não torna o sistema tão bom quanto novo, ou seja, a probabilidade de eficiência pode variar no intervalo 0,5 < PE < 1, de acordo com dados históricos. Caso o ajuste seja eficiente o processo volta a operar no estado 1 (sob controle), caso contrário o processo continua produzindo no Estado 2 (fora de controle).

As probabilidades de transição do estado (*s*,*k*) no momento da inspeção i para o estado (*s**,*k**) no momento da inspeção $i+1$, são os elementos da matriz M

$$M = \begin{pmatrix} B_{00} & B_{01} & B_{02} \\ B_{10} & B_{11} & B_{12} \\ B_{20} & B_{21} & B_{22} \end{pmatrix} \tag{7}$$

Onde $B_{ss*}$ denota a matriz de probabilidade de transição $M_{(s,k)(s*,k*)}$; $s, s* = 0,1,2; k, k* = -1,0,1,...,h$.

Os elementos da 1ª linha da matriz M são iguais aos elementos da 1ª linha da matriz P.

Os elementos não nulos de $B_{10}$ são as probabilidades de transição com k = - 1 ou h. Ou seja, o processo é ajustado na inspeção anterior, e ele reinicia sob controle caso o ajuste seja eficiente, empregando um intervalo de amostragem m:

$$\begin{cases} p_{(1,-1)(0,-1)} = p_{(2,-1)(0,-1)} = p_{(1,h)(0,-1)} = p_{(2,h)(0,-1)} = peq_m R_0 \\ p_{(1,-1)(0,0)} = p_{(2,-1)(0,0)} = p_{(1,h)(0,0)} = p_{(2,h)(0,0)} = peq_m G_0 \\ p_{(1,-1)(0,1)} = p_{(2,-1)(0,1)} = p_{(1,h)(0,1)} = p_{(2,h)(0,1)} = peq_m Y_0 \end{cases} \quad (8)$$

Similarmente tem-se que $B_{20} = B_{10}$

Da mesma forma, os elementos não-nulos de $B_{11}$ são também as probabilidades de transição, com k = -1 ou h. No entanto, após o ajuste, o processo reinicia sob controle caso ajuste seja eficiente, mas o parâmetro μ muda na inspeção atual. Os elementos não-nulos destas matrizes são

$$\begin{cases} p_{(1,-1)(1,-1)} = p_{(2,-1)(1,-1)} = p_{(1,h)(1,-1)} = p_{(2,h)(1,-1)} = pe(1-q_m)R_1 \\ p_{(1,-1)(1,0)} = p_{(2,-1)(1,0)} = p_{(1,h)(1,0)} = p_{(2,h)(1,0)} = pe(1-q_m)G_1 \\ p_{(1,-1)(1,1)} = p_{(2,-1)(1,1)} = p_{(1,h)(1,1)} = p_{(2,h)(1,1)} = pe(1-q_m)Y_1 \end{cases} \quad (9)$$

Similarmente tem-se que $B_{21}$ e $B_{11}$.

Em seguida, temos as matrizes $B_{12}$ e $B_{22}$, que são as probabilidades de transição dos estados (*1, k*) para os estados (*2, k **) e dos estados (*2, k*) para os estados (*2, k**), respectivamente, e $B_{12} = B_{22}$. Nestes casos, o parâmetro μ mudou nas inspeções anteriores, e as probabilidades dessa matriz quando o processo não foi ajustado são:

$$\begin{cases} p_{(1,k)(2,-1)} = p_{(2,k)(2,-1)} = R_1 \\ p_{(1,k)(2,0)} = p_{(2,k)(2,0)} = G_1 \\ p_{(1,k)(2,k+1)} = p_{(2,k)(2,k+1)} = Y_1 \end{cases} \tag{10}$$

para $0 < k < h$.

E as probabilidades de transição dessa matriz quando k= -1 ou h, são

$$\begin{cases} p_{(1,-1)(2,-1)} = p_{(2,-1)(2,-1)} = p_{(1,h)(2,-1)} = p_{(2,h)(2,-1)} = (1-PE)R_1 \\ p_{(1,-1)(2,0)} = p_{(2,-1)(2,0)} = p_{(1,h)(2,0)} = p_{(2,h)(2,0)} = (1-PE)G_1 \\ p_{(1,-1)(2,1)} = p_{(2,-1)(2,1)} = p_{(1,h)(2,1)} = p_{(2,h)(2,1)} = (1-PE)Y_1 \end{cases} \tag{11}$$

considerando que o ajuste do processo não foi eficiente.

Um exemplo da matriz de probabilidade de transição considerando h = 2 será:

$$P = \begin{array}{r} 0-1 \\ 0,0 \\ 0,1 \\ 0,2 \\ 1,-1 \\ 1,0 \\ 1,1 \\ 1,2 \\ 2,-1 \\ 2,0 \\ 2,1 \\ 2,2 \end{array} \begin{pmatrix} q_mR_0 & q_mG_0 & q_mY_0 & 0 & (1-q_m)R_1 & (1-q_m)G_1 & (1-q_m)Y_1 & 0 & 0 & 0 & 0 & 0 \\ q_mR_0 & q_mG_0 & q_mY_0 & 0 & (1-q_m)R_1 & (1-q_m)G_1 & (1-q_m)Y_1 & 0 & 0 & 0 & 0 & 0 \\ q_mR_0 & q_mG_0 & 0 & q_mY_0 & (1-q_m)R_1 & (1-q_m)G_1 & 0 & (1-q_m)Y_1 & 0 & 0 & 0 & 0 \\ q_mR_0 & q_mG_0 & q_mY_0 & 0 & (1-q_m)R_1 & (1-q_m)G_1 & (1-q_m)Y_1 & 0 & 0 & 0 & 0 & 0 \\ q_mR_0 & q_mG_0 & q_mY_0 & 0 & (1-q_m)R_1 & (1-q_m)G_1 & (1-q_m)Y_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & R_1 & G_1 & Y_1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & R_1 & G_1 & 0 & Y_1 \\ q_mR_0 & q_mG_0 & q_mY_0 & 0 & (1-q_m)R_1 & (1-q_m)G_1 & (1-q_m)Y_1 & 0 & 0 & 0 & 0 & 0 \\ q_mR_0 & q_mG_0 & q_mY_0 & 0 & (1-q_m)R_1 & (1-q_m)G_1 & (1-q_m)Y_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & R_1 & G_1 & Y_1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & R_1 & G_1 & 0 & Y_1 \\ q_mR_0 & q_mG_0 & q_mY_0 & 0 & (1-q_m)R_1 & (1-q_m)G_1 & (1-q_m)Y_1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Em ambos os modelos, note que a cadeia de Markov construída desta forma incorpora informações sobre o processo (se ela está sob controle ou fora de controle), o tamanho da sequência de observações na zona amarela e também o tamanho do intervalo de amostragem *m* nas matrizes P e M. As matrizes são recorrentes ergódicas e consequentemente $\Delta = \lim_{u \to \infty} P^u$ existe e não depende da probabilidade dos estados iniciais do processo. Assim, todas as linhas de $\Delta$ são iguais. Denotando a primeira linha da matriz $\Delta$ por $\delta = (\delta_{(0,-1)},...,\delta_{(0,h)},\delta_{(1,-1)},...,\delta_{(1,h)},\delta_{(2,-1)},...,\delta_{(2,h)})$ e resolvendo o sistema de equações lineares $\delta = \delta P$ sujeito à restrição $\sum_{s=0}^{2}\sum_{k=-1}^{h}\delta_{(s,k)} = 1$, $\delta_{(s,k)}$ pode ser determinado. Esta solução pode ser interpretada como a proporção das inspeções dos estados (s,k), $s = 0,1,2$; $k = -1,0,...,h$ após um grande número de inspeções.

# 3. Função Custo

Para obter a função de custo, são necessárias algumas suposições. Uma vez que se decide fazer um ajuste, a paralisação do processo é instantânea. Após o ajuste, caso este tenha sido eficiente, o processo reinicia no Estado I ($\mu = \mu_0$), caso contrário, ele continua operando no Estado II ($\mu = \mu_1$). Neste trabalho, os custos seguem uma estrutura semelhante aos projetos econômicos usuais, onde $\mathbf{c_i}$ é o custo para uma única inspeção e $\mathbf{c_a}$ é o custo para o ajuste. Um item é declarado como não-conforme se o valor observado da característica de qualidade está além dos limites de especificação ou se uma sequência de h observações estão na amarela. Assim, os próximos custos relacionados com itens não-conformes e descartados também estão incluídos no modelo: $\mathbf{c_n}$ é o custo para a entrega de um item não-conforme (este item é enviado para o cliente ou para a próxima etapa da produção), e $\mathbf{c_d}$ é o custo para descartar o item examinado. Para cada ciclo de inspeção, os (m-1) itens são enviados para o cliente ou para as próximas fases da produção. Supõe-se que todos os itens inspecionados são descartados ou retificados nas fases posteriores uma vez que a inspeção é parcialmente destrutiva.

Para um número suficientemente grande de inspeções, $\delta = (\delta_{(0,-1)},...,\delta_{(0,h)},\delta_{(1,-1)},...,\delta_{(1,h)},\delta_{(2,-1)},...,\delta_{(2,h)})$ é o vetor de probabilidades dos estados da Cadeia de Markov. Seja C a variável aleatória relacionada com o custo de cada ciclo de inspeção.

Assumimos valores discretos relacionados aos estados da cadeia de Markov. O custo dos estados $(s,k), s = 0,1,2; k = -1,0,...h$ pode ser escrito como

$$C(s,k) = c_i + a(s,k) + n(s,k) + c_d ; s = 0,1,2; k = -1,0,...,h \qquad (12)$$

- $c_i$, $c_d$, respectivamente, são os custos de inspeção e descarte de um único item, constante para todos os estados $(s,k), s=0,1,2; k=-1,0,...h$;
- $a(s,k)$ é o custo de ajuste do processo;
- $n(s,k)$ é o custo de envio de um item não conforme ao cliente ou para a próxima etapa da produção.

No próximo parágrafo, os diferentes custos estão detalhados.

- Custo de ajuste do processo:

  O custo para ajuste está incluído para os estados $(s,k), s=0,1,2; k=-1,0,\ldots,h$.

Assim,

$$a(s,k) = c_a; s=0,1,2; k=-1,0,\ldots,h \quad .$$

- Custo de envio de itens não-conforme ao cliente ou para a próxima etapa do processo:

  Para os estados $(0,k); k=-1,0,...h$, todos os itens são produzidos com $\mu=\mu_0$, assim,

$$n(0,k) = c_n p_1 (m-1); k=-1,0,...,h,$$

com $p_1 = 1 - P(\mu - LE \le X_i \le \mu + LE \mid \mu = \mu_0)$, sendo a probabilidade de itens conformes. De forma similar para os estados (*2*,*k*), quando todos os itens são produzidos com $\mu=\mu_1$, temos

$$n(2,k) = c_n p_2 (m-1); k=-1,0,...,h,$$

com $p_2 = 1 - P(\mu - LE \le X_i \le \mu + LE \mid \mu = \mu_1)$, sendo a probabilidade de itens conformes. Para os estados (*1*,*k*), $k=-1,0,...h$, $v<m$ itens são produzidos com $\mu=\mu_0$, e então, $m-v$ são produzidos com $\mu=\mu_1$. Considerando todas as possibilidades

$$n(1,k) = c_n \frac{\sum_{v=1}^{m} \pi(1-\pi)^{v-1}[(v-1)p_1 + (m-v)p_2]}{1-(1-\pi)^m}; k = -1,0,...,h.$$

Para um número suficientemente grande de inspeções, consideramos nossa configuração como renovação-recompensa. Um processo de contagem {N(t), t ≥ 0} é dito ser de renovação, se uma sequência de variáveis aleatórias não-negativas $\{X_1, X_2, ...\}$ é independente e identicamente distribuída. Assim, um processo de renovação é um processo de contagem tal que o tempo até o primeiro evento ocorrer tem uma distribuição F, o tempo entre o primeiro e o segundo evento tem, independente do tempo do primeiro, a mesma distribuição F, e assim sucessivamente. Quando um evento ocorre é dito que houve uma renovação.

Dessa forma, o custo médio por item *CE(m)* é a razão do custo esperado por ciclo de inspeção *E(V)* pela quantidade de itens enviados ao cliente ou para a fase seguinte da produção, expressa por

$$CE(m) = \frac{E(V)}{m-1} = \frac{\sum_{s=0}^{2} \sum_{k=-1}^{h} C(s,k)\delta_{(s,k)}}{m-1}. \qquad (13)$$

O problema consiste em determinar os valores de

$$(m^o, W^o, C^o, h^o, pe^o) = \arg \min_{(m,W,C,h,pe)} [CE(m)] \qquad (14)$$

onde *m* é o intervalo de amostragem; as constantes *W* e *C* são usadas para representar, respectivamente, o limite de alarme e o limite de controle, e *h* é o tamanho da sequencia de valores que estão na zona amarela.

# 4. Exemplos Numéricos e Discussões

Para ilustrar o procedimento proposto, considere o exemplo adaptado de Taguchi et al. (1989), Taguchi et al. (2004) e Trindade et al. (2007).

Um fabricante de circuitos integrados de alto volume quer instalar um sistema para controlar a medição de alguma dimensão de interesse. Dados históricos permite uma estimativa dos componentes de custos como $c_i$ = \$0.25, $c_n$ = \$20, $c_d$ = \$2.0, e $c_a$ = \$900. Os limites de especificação (LE) foram fixados em ± 1,5, e a mudança do processo sob-controle [Estado I ($\mu_0 = 0$)] para fora-de-controle [Estado II ($\mu_1 = 1$)] pode ser descrito por uma distribuição geométrica com parâmetro $\pi = 0.001$. O desvio padrão do processo é conhecido e é igual a 0,5. Para calcular o valor ideal, um programa em MatLab foi desenvolvido para esta tarefa.

A Figura 3 mostra o gráfico do custo esperado versus o intervalo de amostragem m para alguns casos escolhidos para ilustrar o comportamento do conjunto ideal considerando a probabilidade de eficiência pe = 1.

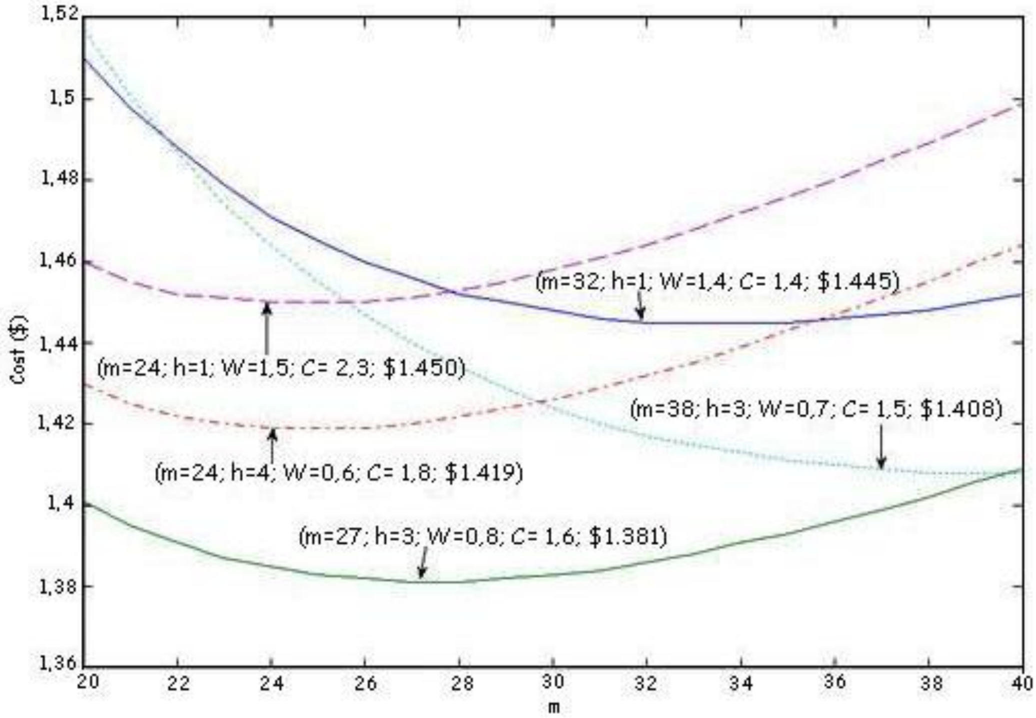


Figura 3 – Gráfico do custo médio versus m, com alternativa de valores para h, W e C.

O melhor projeto é o seguinte: O intervalo de amostragem é m° = 27; o comprimento da seqüência de observações na zona amarela é h° = 3; a largura do limite de alerta é W° = 0,8, e a largura do limite de controle é C° = 1,6; os resultados geram um custo médio de $ 1,381. A proposta atual é 4,4% mais barato quando comparado com a abordagem apresentada em Ho, Medeiros & Borges (2007). Nesse caso, o intervalo de amostragem aumenta para m° = 32 como o custo médio também aumenta para $1,445 (com h = 1 e W = C = 1.4).

Considerando o mesmo exemplo e utilizando agora o modelo onde o ajuste não torna o sistema 100% eficiente, o custo do ajuste assume uma função quadrática em função da probabilidade de eficiência pe. Este custo foi definido especificando um custo mínimo para uma probabilidade pe=0,5; e calculando então a função quadrática. Utilizando o programa desenvolvido no software Matlab, variando PE no intervalo 0.5 < PE <1.0, ou seja, transformando-o em um parâmetro de otimização, obtemos a seguinte opção: O intervalo de amostragem é mº = 24; o comprimento da sequência de observações na zona amarela é hº = 3; a largura do limite de alerta é Wº = 0.8; a largura do limite de controle é $C^o = 1.6$; e a probabilidade de eficiência que gera o menor custo é 0.84. Esses resultados fornecem o custo médio de $1.376, que é melhor do que o obtido anteriormente.

Considerando na prática um ajuste 100%, será sempre melhor para o processo, porém pode ser mais caro. O objetivo do segundo modelo tornando pe um parâmetro de otimização, foi encontrar uma probabilidade que apesar de não tornar o sistema tão bom quanto novo, apresenta um custo menor, pois o objetivo do trabalho é minimizar o custo. Desta forma, os modelos fornecem a possibilidade de analisar os tipos de serviço para ajuste do processo, e como as matrizes são dinâmicas é possível trabalhar com os programas da forma mais conveniente.

# Referências Bibliográficas


BORGES, W. et al. An analysis of Taguchi’s on-line quality monitoring procedure for attributes with diagnosis errors. *Applied Stochastic Models in Business and Industry*, **17**, 261-276. 2001

DASGUPTA, T.; MANDAL, A. Estimation of process parameters to determine the optimum diagnosis interval for control of defective items. Technometrics, 50, 167-181. 2008.

DING, J.; GONG, L. The effect of testing equipment shift on optimal decisions in a repetitive testing process. European Journal of Operational Research, 186, 330-350. 2008.

GREENBERG, B. S.; STOKES, S. L. Repetitive Testing in the Presence of Inspection Errors. *Technometrics*, **37**, 102-111. 1995

HO, L.L. et al. An alternative model for on-line quality monitoring for variables. International Journal of Production Economics, 107, 202-222. 2007.

JIANG, W.; TSUI, K. L. An economic model for integrated APC and SPC control charts. IIE Transactions, 32, 505-513. 2000.

NANDI, S. N.; SREEHARI, M. Some improvements in Taguchi's economic method allowing continued quality deterioration in production process. *Communications in Statistics - Theory and Methods* ; **28**(5): 1169-1181. 1999.

NAYEBPOUR, M.R.; WOODALL, W.H. An analysis of Taguchi’s on-line quality monitoring procedure for attributes. *Technometrics*, 35, 53-60, 1993.

QUININO, R.C.; SUYAMA, E. Número ótimo de classificações independentes com erro na avaliação da conformidade de produtos. *Pesquisa Operacional*, **22**(1), 1-8. 2002.

QUININO, R. & HO, L.L.Repetitive tests as an economic alternative procedure to control attributes with diagnosis errors. *European Journal of Operation Research*, **155**, 209-225. 2004.

TAGUCHI, G. et al. Quality Engineering in Production in Systems. New York: Mcgraw Hill, 1989

TAGUCHI, G.; CHOWDHURY, S.; WU, Y. Taguchi’s Quality Engineering – Handbook. John Wiley & Sons, Inc. New Jersey. 2004

TRINDADE, A. et al. Monitoring process for attributes with quality deterioration and diagnosis errors. *Applied Stochastic Models in Business and Industry*, **23** (4), 339-358. 2007.

WANG, C. H.; SHEU, S. H. Determining the optimal production-maintenance policy with inspection errors: using a Markov chain. Comput. Operat. Res, 30: 1-17. 2003.

# Anexo 1

Macro desenvolvida no *software Matlab* para o exemplo numérico, considerando a probabilidade de eficiência PE = 1.

```
%==========================================================================
%      Exemplo Numérico: exemplo adaptado de Taguchi et al. (1989), Taguchi et
%      al. (2004) e Trindade et al. (2007).
%==========================================================================

clear all
tic;
format long
con=1;

cn=20; %custo de envio de n-conf
cd=2 ; %custo de destruir peça
mi2=1; %média fora de controle
ca=900; %custo ajuste
ci=0.25; %Custo de Inspeção
pi=0.001; %probabilidade de mudança

mi1=0; %média sob controle
dp=0.5; %desvio padrão
LE=1.5; %Limite de Especificação
pd=1; %relativo ao que chamamos de preventivo.
pc=1; %relativo ao que chamamos corretivo

r=eps;
cc=1;
for m=2:1:50
       for h=1:1:5
            for A=0.5:0.1:0.9
                for B=1.2:0.1:1.8

L1=m;
L2=m;

 cac=ca*pc;
 cap=ca*pd;

%+++++++++++++++++++++++++++++++++++++++++
% mp= Matriz de probabilidade de transição
%+++++++++++++++++++++++++++++++++++++++++
mp=zeros(3*(h+2),3*(h+2));
```

```
qm=(1-pi)^m;
qL1=(1-pi)^L1;
qL2=(1-pi)^L2;
clc1=1-(cdf('norm',B,mi1,dp)-cdf('norm',-B,mi1,dp));
clc2=1-(cdf('norm',B,mi2,dp)-cdf('norm',-B,mi2,dp));

clb1=(cdf('norm',B,mi1,dp)-cdf('norm',A,mi1,dp))+(cdf('norm',-A,mi1,dp)-cdf('norm',-
B,mi1,dp));

clb2=(cdf('norm',B,mi2,dp)-cdf('norm',A,mi2,dp))+(cdf('norm',-A,mi2,dp)-cdf('norm',-
B,mi2,dp));

cla1=(cdf('norm',A,mi1,dp)-cdf('norm',-A,mi1,dp));
cla2=(cdf('norm',A,mi2,dp)-cdf('norm',-A,mi2,dp));

cont=1;
for i=0:2
        for j=-1:h
        a1(cont)=i;
        a2(cont)=i;
        b1(cont)=j;
        b2(cont)=j;
        cont=cont+1;
        end
end

for i=1:3*(h+2)
        z1=a1(i);
        z2=b1(i);

        for j=1:3*(h+2)
        z11=a2(j);
        z22=b2(j);

        if z1==0 & z2==-1 & z11==0 & z22==-1
        mp(i,j)=qL2*clc1;
        end

        if z1==0 & (z2==h | z2==0) & z11==0 & z22==-1
        mp(i,j)=qL1*clc1;
        end

        if z1==0 & z2==-1 & z11==0 & z22==0
        mp(i,j)=qL2*cla1;
        end

        if z1==0 & (z2==h | z2==0) & z11==0 & z22==0
        mp(i,j)=qL1*cla1;
        end

        if z1==0 & z2==-1 & z11==0 & z22==1
        mp(i,j)=qL2*clb1;
        end
```

```
if z1==0 & (z2==h | z2==0) & z11==0 & z22==1
mp(i,j)=qL1*clb1;
end

if z1==0 & z2>0 & z2<h & z11==0 & z22==z2+1
mp(i,j)=qm*clb1;
end

if z1==0 & z2>0 & z2<h & z11==0 & z22==-1
mp(i,j)=qm*clc1;
end

if z1==0 & z2>0 & z2<h & z11==0 & z22==0
mp(i,j)=qm*cla1;
end

%=============================================
if z1==0 & z2==-1 & z11==1 & z22==-1
mp(i,j)=(1-qL2)*clc2;
end

 if z1==0 & (z2==h | z2==0) & z11==1 & z22==-1
mp(i,j)=(1-qL1)*clc2;
end

if z1==0 & z2==-1 & z11==1 & z22==0
mp(i,j)=(1-qL2)*cla2;
end

if z1==0 & (z2==h | z2==0) & z11==1 & z22==0
mp(i,j)=(1-qL1)*cla2;
end

if z1==0 & z2==-1 & z11==1 & z22==1
mp(i,j)=(1-qL2)*clb2;
end

if z1==0 & (z2==h | z2==0) & z11==1 & z22==1
mp(i,j)=(1-qL1)*clb2;
end

if z1==0 & z2>0 & z2<h & z11==1 & z22==z2+1
mp(i,j)=(1-qm)*clb2;
end

if z1==0 & z2>0 & z2<h & z11==1 & z22==-1
mp(i,j)=(1-qm)*clc2;
end

if z1==0 & z2>0 & z2<h & z11==1 & z22==0
mp(i,j)=(1-qm)*cla2;
end
```

```
%================================================
 if z1==1 & z2==-1 & z11==0 & z22==-1
mp(i,j)=qL2*pc*clc1;
end

if z1==1 & z2==-1 & z11==0 & z22==0
mp(i,j)=qL2*pc*cla1;
end

if z1==1 & z2==-1 & z11==0 & z22==1
mp(i,j)=qL2*pc*clb1;
end

 if z1==1 & z2==-1 & z11==1 & z22==-1
mp(i,j)=(1-qL2)*pc*clc2;
end

 if z1==1 & z2==-1 & z11==1 & z22==0
mp(i,j)=(1-qL2)*pc*cla2;
end

if z1==1 & z2==-1 & z11==1 & z22==1
mp(i,j)=(1-qL2)*pc*clb2;
end

 if z1==1 & z2==-1 & z11==2 & z22==-1
mp(i,j)=(1-pc)*clc2;
end

if z1==1 & z2==-1 & z11==2 & z22==0
mp(i,j)=(1-pc)*cla2;
end

if z1==1 & z2==-1 & z11==2 & z22==1
mp(i,j)=(1-pc)*clb2;
end

 if z1==1 & z2==0 & z11==2 & z22==0
mp(i,j)=cla2;
end

if z1==1 & z2==0 & z11==2 & z22==1
mp(i,j)=clb2;
end

if z1==1 & z2==0 & z11==2 & z22==-1
mp(i,j)=clc2;
end

if z1==1 & z2>0 & z2<h & z11==2 & z22==z2+1
mp(i,j)=clb2;
end
```

```
 if z1==1 & z2>0 & z2<h & z11==2 & z22==0
mp(i,j)=cla2;
end

 if z1==1 & z2>0 & z2<h & z11==2 & z22==-1
mp(i,j)=clc2;
end

 if z1==1 & z2==h & z11==0 & z22==-1
mp(i,j)=pd*qL1*clc1;
end

if z1==1 & z2==h & z11==0 & z22==0
mp(i,j)=pd*qL1*cla1;
end

if z1==1 & z2==h & z11==0 & z22==1
mp(i,j)=pd*qL1*clb1;
end

if z1==1 & z2==h & z11==1 & z22==-1
mp(i,j)=pd*(1-qL1)*clc2;
end

if z1==1 & z2==h & z11==1 & z22==0
mp(i,j)=pd*(1-qL1)*cla2;
end

if z1==1 & z2==h & z11==1 & z22==1
mp(i,j)=pd*(1-qL1)*clb2;
end

%extra

if z1==1 & z2==h & z11==2 & z22==-1
mp(i,j)=(1-pd)*clc2;
end

if z1==1 & z2==h & z11==2 & z22==0
mp(i,j)=(1-pd)*cla2;
end

if z1==1 & z2==h & z11==2 & z22==1
mp(i,j)=(1-pd)*clb2;
end

 %extra

%==================================================
 if z1==2 & z2==-1 & z11==0 & z22==-1
mp(i,j)=qL2*pc*clc1;
end
```

```
if z1==2 & z2==-1 & z11==0 & z22==0
mp(i,j)=qL2*pc*cla1;
end

if z1==2 & z2==-1 & z11==0 & z22==1
mp(i,j)=qL2*pc*clb1;
end

 if z1==2 & z2==-1 & z11==1 & z22==-1
 mp(i,j)=(1-qL2)*pc*clc2;
 end

 if z1==2 & z2==-1 & z11==1 & z22==0
 mp(i,j)=(1-qL2)*pc*cla2;
 end

 if z1==2 & z2==-1 & z11==1 & z22==1
 mp(i,j)=(1-qL2)*pc*clb2;
 end

 if z1==2 & z2==-1 & z11==2 & z22==-1
 mp(i,j)=(1-pc)*clc2;
 end

 if z1==2 & z2==-1 & z11==2 & z22==0
 mp(i,j)=(1-pc)*cla2;
 end

 if z1==2 & z2==-1 & z11==2 & z22==1
 mp(i,j)=(1-pc)*clb2;
 end

 if z1==2 & z2==0 & z11==2 & z22==0
 mp(i,j)=cla2;
 end

 if z1==2 & z2==0 & z11==2 & z22==1
 mp(i,j)=clb2;
 end

 if z1==2 & z2==0 & z11==2 & z22==-1
 mp(i,j)=clc2;
 end

 if z1==2 & z2>0 & z2<h & z11==2 & z22==z2+1
 mp(i,j)=clb2;
 end

 if z1==2 & z2>0 & z2<h & z11==2 & z22==0
 mp(i,j)=cla2;
 end

 if z1==2 & z2>0 & z2<h & z11==2 & z22==-1
```

```
mp(i,j)=clc2;
end

 if z1==2 & z2==h & z11==0 & z22==-1
mp(i,j)=pd*qL1*clc1;
end

if z1==2 & z2==h & z11==0 & z22==0
mp(i,j)=pd*qL1*cla1;
end

if z1==2 & z2==h & z11==0 & z22==1
mp(i,j)=pd*qL1*clb1;
end

if z1==2 & z2==h & z11==1 & z22==-1
mp(i,j)=pd*(1-qL1)*clc2;
end

if z1==2 & z2==h & z11==1 & z22==0
mp(i,j)=pd*(1-qL1)*cla2;
end

if z1==2 & z2==h & z11==1 & z22==1
mp(i,j)=pd*(1-qL1)*clb2;
end

 %extra

if z1==2 & z2==h & z11==2 & z22==-1
mp(i,j)=(1-pd)*clc2;
end

if z1==2 & z2==h & z11==2 & z22==0
mp(i,j)=(1-pd)*cla2;
end

if z1==2 & z2==h & z11==2 & z22==1
mp(i,j)=(1-pd)*clb2;
end

 %extra
%=====================================================
        end
    end
 %=====================================================
 mp0=mp;
 tamanho=size(mp);
 AA=transpose(mp)-eye(tamanho(1,1));
 AA(tamanho(1,1),:)=ones(1,tamanho(1,1));
 BB=zeros(tamanho(1,1),1);
 BB(tamanho(1,1),1)=1;
```

```
y=AA\BB;
y=y';

 CUSTO=zeros(1,3*(h+2));
mL=zeros(1,3*(h+2));
conta=1;
   for i=0:2
       for j=-1:h

ta=size(y);
ta1=ta(1,2);
ta2=ta1/3;

if i==0 & (j==-1 | j==h)

  if j==-1
 pa1=0;
      for m1=1:1:ta1
    pa=mp0(m1,1)*y(m1);
    pa1=pa+pa1;
    end

  pa2=0;
      for m1=[2 ta2 ta2+2 2*ta2 2*ta2+2 3*ta2]
    pa=mp0(m1,1)*y(m1);
    pa2=pa+pa2;
    end
    r1=pa2/(pa1+r); %L1

  pa3=0;
      for m1=[1 ta2+1 2*ta2+1]
      pa=mp0(m1,1)*y(m1);
      pa3=pa+pa3;
      end
    r2=pa3/(pa1+r); %L2

    end

    if j==h
      pa1=0;
             for m1=1:1:ta1
           pa=mp0(m1,ta2)*y(m1);
           pa1=pa+pa1;
           end

      pa2=0;
             for m1=[2 ta2 ta2+2 2*ta2 2*ta2+2 3*ta2]
           pa=mp0(m1,ta2)*y(m1);
           pa2=pa+pa2;
           end
           r1=pa2/(pa1+r);

     pa3=0;
```

```
            for m1=[1 ta2+1 2*ta2+1]
                pa=mp0(m1,ta2)*y(m1);
                pa3=pa+pa3;
                end
                r2=pa3/(pa1+r); %L2
                  end
                r3=1-r2-r1;

                    if j==-1

costaa=(ci+cd+ cac+(1-cdf('norm',LE,mi1,dp)+cdf('norm',-LE,mi1,dp))*(m-1)*cn)*r3;
costab=(ci+cd+ cac+(1-cdf('norm',LE,mi1,dp)+cdf('norm',-LE,mi1,dp))*(L2-1)*cn)*r2;
costac=(ci+cd+ cac+(1-cdf('norm',LE,mi1,dp)+cdf('norm',-LE,mi1,dp))*(L1-1)*cn)*r1;
costa=costaa+costab+costac;

                        CUSTO(conta)=costa;
                        mL(conta)=m*r3+L1*r1+L2*r2;
                    end

                    if j==h

costaa=(ci+cd+ cap+(1-cdf('norm',LE,mi1,dp)+cdf('norm',-LE,mi1,dp))*(m-1)*cn)*r3;
costab=(ci+cd+ cap+(1-cdf('norm',LE,mi1,dp)+cdf('norm',-LE,mi1,dp))*(L2-1)*cn)*r2;
costac=(ci+cd+ cap+(1-cdf('norm',LE,mi1,dp)+cdf('norm',-LE,mi1,dp))*(L1-1)*cn)*r1;
costa=costaa+costab+costac;

                        CUSTO(conta)=costa;
                        mL(conta)=m*r3+L1*r1+L2*r2;
                    end

                end

                if i==0 & j>=0 & j<h

                    pa1=0;
                    for m1=1:1:ta1
                        pa=mp0(m1,j+2)*y(m1);
                        pa1=pa+pa1;
                    end

                    pa2=0;

                    for m1=[2 ta2 ta2+2 2*ta2 2*ta2+2 3*ta2]
                        pa=mp0(m1,j+2)*y(m1);
                        pa2=pa+pa2;
                    end
                    r1=pa2/(pa1+r);

                    pa3=0;
                    for m1=[1 ta2+1 2*ta2+1]
                        pa=mp0(m1,j+2)*y(m1);
                        pa3=pa+pa3;
                    end
```

```
                    r2=pa3/(pa1+r); %L2

                    r3=1-r2-r1;

costaa=(ci+cd+(1-cdf('norm',LE,mi1,dp)+cdf('norm',-LE,mi1,dp))*(m-1)*cn)*r3;
costab=(ci+cd+(1-cdf('norm',LE,mi1,dp)+cdf('norm',-LE,mi1,dp))*(L2-1)*cn)*r2;
costac=(ci+cd+(1-cdf('norm',LE,mi1,dp)+cdf('norm',-LE,mi1,dp))*(L1-1)*cn)*r1;
costa=costaa+costab+costac;

                    CUSTO(conta)=costa;

                    mL(conta)=m*r3+L1*r1+L2*r2;
                end

                if i==1 & (j==-1 |j==h)

                    if j==-1
                        pa1=0;
                        for m1=1:1:ta1
                            pa=mp0(m1,ta2+1)*y(m1);
                            pa1=pa+pa1;
                        end

                        pa2=0;
                        for m1=[2 ta2 ta2+2 2*ta2 2*ta2+2 3*ta2]
                            pa=mp0(m1,ta2+1)*y(m1);
                            pa2=pa+pa2;
                        end
                        r1=pa2/(pa1+r); %L1

                        pa3=0;
                        for m1=[1 ta2+1 2*ta2+1]
                            pa=mp0(m1,ta2+1)*y(m1);
                            pa3=pa+pa3;
                        end
                        r2=pa3/(pa1+r); %L2

                    end

                    if j==h

                        pa1=0;
                        for m1=1:1:ta1
                            pa=mp0(m1,2*ta2)*y(m1);
                            pa1=pa+pa1;
                        end

                        pa2=0;

                        for m1=[2 ta2 ta2+2 2*ta2 2*ta2+2 3*ta2]
                            pa=mp0(m1,2*ta2)*y(m1);
                            pa2=pa+pa2;
                        end
```

```
        r1=pa2/(pa1+r); %L1
        pa3=0;
        for m1=[1 ta2+1 2*ta2+1]
            pa=mp0(m1,2*ta2)*y(m1);
            pa3=pa+pa3;
        end
        r2=pa3/(pa1+r); %L2

    end

    r3=1-r2-r1;

    p1=(1-cdf('norm',LE,mi1,dp)+cdf('norm',-LE,mi1,dp));
    p2=(1-cdf('norm',LE,mi2,dp)+cdf('norm',-LE,mi2,dp));

    n1=0;
    for vt=1:m
 pt=((1-pi)^(vt-1))*pi*((vt-1)*p1*cn+(m-vt)*p2*cn)/(1-qm);
        n1 =pt+n1;
    end

    n2=0;
    for vt=1:L1
pt=((1-pi)^(vt-1))*pi*((vt-1)*p1*cn+(L1-vt)*p2*cn)/(1-qL1);
        n2 =pt+n2;
    end


    n3=0;
    for vt=1:L2
pt=((1-pi)^(vt-1))*pi*((vt-1)*p1*cn+(L2-vt)*p2*cn)/(1-qL2);
        n3 =pt+n3;
    end

    if j==-1

        costaa=(n1+ci+cd+cac)*r3;
        costab=(n2+ci+cd+cac)*r1;
        costac=(n3+ci+cd+cac)*r2;

        costa=costaa+costab+costac;

        CUSTO(conta)=costa;
        mL(conta)=m*r3+L1*r1+L2*r2;
    end

    if j==h

        costaa=(n1+ci+cd+cap)*r3;
        costab=(n2+ci+cd+cap)*r1;
        costac=(n3+ci+cd+cap)*r2;

        costa=costaa+costab+costac;
```

```
            CUSTO(conta)=costa;
            mL(conta)=m*r3+L1*r1+L2*r2;
        end

    end

if i==1 & j>=0 & j<h

        pa1=0;
        for m1=1:1:ta1
            pa=mp0(m1,j+ta2+2)*y(m1);
            pa1=pa+pa1;
        end

        pa2=0;

        for m1=[2 ta2 ta2+2 2*ta2 2*ta2+2 3*ta2]
            pa=mp0(m1,j+ta2+2)*y(m1);
            pa2=pa+pa2;
        end
        r1=pa2/(pa1+r);
        pa3=0;
        for m1=[1 ta2+1 2*ta2+1]
            pa=mp0(m1,j+ta2+2)*y(m1);
            pa3=pa+pa3;
        end
        r2=pa3/(pa1+r); %L2

        r3=1-r2-r1;

        p1=(1-cdf('norm',LE,mi1,dp)+cdf('norm',-LE,mi1,dp));
        p2=(1-cdf('norm',LE,mi2,dp)+cdf('norm',-LE,mi2,dp));

        n1=0;
        for vt=1:m
            pt=((1-pi)^(vt-1))*pi*((vt-1)*p1*cn+(m-vt)*p2*cn)/(1-qm);
            n1 =pt+n1;
        end

        n2=0;
        for vt=1:L1
            pt=((1-pi)^(vt-1))*pi*((vt-1)*p1*cn+(L1-vt)*p2*cn)/(1-qL1);
            n2 =pt+n2;
        end

        n3=0;
        for vt=1:L2
            pt=((1-pi)^(vt-1))*pi*((vt-1)*p1*cn+(L2-vt)*p2*cn)/(1-qL2);
            n3 =pt+n3;
        end

            costaa=(n1+ci+cd)*r3;
```

```
                    costab=(n2+ci+cd)*r1;
                    costac=(n3+ci+cd)*r2;

                    costa=costaa+costab+costac;

                    CUSTO(conta)=costa;
                    mL(conta)=m*r3+L1*r1+L2*r2;
 end

%********************

            if i==2 & (j==-1 | j==h)
                if j==-1
                    pa1=0;
                    for m1=1:1:ta1
                        pa=mp0(m1,2*ta2+1)*y(m1);
                        pa1=pa+pa1;
                    end

                    pa2=0;
                    for m1=[2 ta2 ta2+2 2*ta2 2*ta2+2 3*ta2]
                        pa=mp0(m1,2*ta2+1)*y(m1);
                        pa2=pa+pa2;
                    end
                    r1=pa2/(pa1+r); %L1

                    pa3=0;
                    for m1=[1 ta2+1 2*ta2+1]
                        pa=mp0(m1,2*ta2+1)*y(m1);
                        pa3=pa+pa3;
                    end
                    r2=pa3/(pa1+r); %L2

                end
                if j==h

                    pa1=0;
                    for m1=1:1:ta1
                        pa=mp0(m1,3*ta2)*y(m1);
                        pa1=pa+pa1;
                    end
                    pa2=0;
                    for m1=[2 ta2 ta2+2 2*ta2 2*ta2+2 3*ta2]
                        pa=mp0(m1,3*ta2)*y(m1);
                        pa2=pa+pa2;
                    end
                    r1=pa2/(pa1+r);
                    pa3=0;
                    for m1=[1 ta2+1 2*ta2+1]
                        pa=mp0(m1,3*ta2)*y(m1);
                        pa3=pa+pa3;
                    end
                    r2=pa3/(pa1+r); %L2
```

```
                        end
                        r3=1-r2-r1;

                        if j==-1

costaa=(ci+cd+ cac+(1-cdf('norm',LE,mi2,dp)+cdf('norm',-LE,mi2,dp))*(m-1)*cn)*r3;
costab=(ci+cd+ cac+(1-cdf('norm',LE,mi2,dp)+cdf('norm',-LE,mi2,dp))*(L2-1)*cn)*r2;
costac=(ci+cd+ cac+(1-cdf('norm',LE,mi2,dp)+cdf('norm',-LE,mi2,dp))*(L1-1)*cn)*r1;
costa=costaa+costab+costac;

                            CUSTO(conta)=costa;
                            mL(conta)=m*r3+L1*r1+L2*r2;
                        end
                        if j==h

costaa=(ci+cd+ cap+(1-cdf('norm',LE,mi2,dp)+cdf('norm',-LE,mi2,dp))*(m-1)*cn)*r3;
costab=(ci+cd+ cap+(1-cdf('norm',LE,mi2,dp)+cdf('norm',-LE,mi2,dp))*(L2-1)*cn)*r2;
costac=(ci+cd+ cap+(1-cdf('norm',LE,mi2,dp)+cdf('norm',-LE,mi2,dp))*(L1-1)*cn)*r1;
costa=costaa+costab+costac;

                            CUSTO(conta)=costa;
                            mL(conta)=m*r3+L1*r1+L2*r2;
                        end
                    end

                    if i==2 & j>=0 & j<h
                        pa1=0;
                        for m1=1:1:ta1
                            pa=mp0(m1,j+2*ta2+2)*y(m1);
                            pa1=pa+pa1;
                        end

                        pa2=0;
                        for m1=[2 ta2 ta2+2 2*ta2 2*ta2+2 3*ta2]
                            pa=mp0(m1,j+2*ta2+2)*y(m1);
                            pa2=pa+pa2;
                        end
                        r1=pa2/(pa1+r);
                        pa3=0;
                        for m1=[1 ta2+1 2*ta2+1]
                            pa=mp0(m1,j+2*ta2+2)*y(m1);
                            pa3=pa+pa3;
                        end
                        r2=pa3/(pa1+r); %L2
                        r3=1-r2-r1;

costaa=(ci+cd+(1-cdf('norm',LE,mi2,dp)+cdf('norm',-LE,mi2,dp))*(m-1)*cn)*r3;
costab=(ci+cd+(1-cdf('norm',LE,mi2,dp)+cdf('norm',-LE,mi2,dp))*(L2-1)*cn)*r2;
costac=(ci+cd+(1-cdf('norm',LE,mi2,dp)+cdf('norm',-LE,mi2,dp))*(L1-1)*cn)*r1;
costa=costaa+costab+costac;

                        CUSTO(conta)=costa;
```

```
                    mL(conta)=m*r3+L1*r1+L2*r2;
                end

                conta =conta+1;
            end
        end

        VC=y*CUSTO';
        tciclo=y*mL';
        EK=VC/(tciclo-1);

       vcusto(cc)=EK;
       bm(cc)=m;
       bh(cc)=h;
       bA(cc)=A;
       bB(cc)=B;
cc=cc+1;
   end
       end
       end
   end
  [customin,posmin]=min(vcusto);
   m0=bm(posmin)
   h0=bh(posmin)
   A0=bA(posmin)
   B0=bB(posmin)
   custominA=customin
   toc;
```

# Anexo 2

Macro desenvolvida no *software Matlab* para o exemplo numérico, considerando a probabilidade de eficiência 0.5< PE < 1.0.

```
%==============================================================================
%       Exemplo Numérico: exemplo adaptado de Taguchi et al. (1989), Taguchi et
%       al. (2004) e Trindade et al. (2007).
%==============================================================================


clear all
tic;
format long
con=1;

% PE: probabilidade de eficiência do ajuste
pi=0.001;      % probabilidade de mudança de estado
mi1=0;         % média sob controle
mi2=1;         % média fora de controle
dp=0.5;        % desvio padrão
LE=1.5;        % Limite de Especificação

% CUSTOS
cn=20;         % custo de envio de um item não conforme
cd=2 ;         % custo de decarte da peça inspecionada
ci=0.25;       % Custo de Inspeção


pd=1; % relativo ao que chamamos de preventivo

pc=1; % relativo ao que chamamos corretivo

r=eps;
cc=1;
for m=2:1:50
    for h=1:1:5
        for A=0.5:0.1:0.9
            for B=1.2:0.1:1.8
                for PE=0.5:0.01:1.0
                    ca=200*(PE)^2+700*PE


                    cac=ca*pc;
                    cap=ca*pd;
```

```
%mp= Matriz de probabilidade de transição
%++++++++++++++++++++++++++++++++
mp=zeros(3*(h+2),3*(h+2));
qm=(1-pi)^m;     %PROBABILIDADE DE NÃO MUDAR

clc1=1-(cdf('norm',B,mi1,dp)-cdf('norm',-B,mi1,dp));
clc2=1-(cdf('norm',B,mi2,dp)-cdf('norm',-B,mi2,dp));

clb1=(cdf('norm',B,mi1,dp)-cdf('norm',A,mi1,dp))+(cdf('norm',-
A,mi1,dp)-cdf('norm',-B,mi1,dp));     %REGIÃO AMARELA MI1
clb2=(cdf('norm',B,mi2,dp)-cdf('norm',A,mi2,dp))+(cdf('norm',-
A,mi2,dp)-cdf('norm',-B,mi2,dp));     %REGIÃO AMARELA MI2


cla1=(cdf('norm',A,mi1,dp)-cdf('norm',-A,mi1,dp));
cla2=(cdf('norm',A,mi2,dp)-cdf('norm',-A,mi2,dp));

cont=1;
for i=0:2
    for j=-1:h
        a1(cont)=i;
        a2(cont)=i;
        b1(cont)=j;
        b2(cont)=j;
        cont=cont+1;
    end
end


for i=1:3*(h+2)
    z1=a1(i);
    z2=b1(i);
    for j=1:3*(h+2)
        z11=a2(j);
        z22=b2(j);

        if z1==0 & z2==-1 & z11==0 & z22==-1
            mp(i,j)=qm*clc1;
        end

        if z1==0 & (z2==h | z2==0) & z11==0 & z22==-1
            mp(i,j)=qm*clc1;
        end

        if z1==0 & z2==-1 & z11==0 & z22==0
            mp(i,j)=qm*cla1;
        end

        if z1==0 & (z2==h | z2==0) & z11==0 & z22==0
            mp(i,j)=qm*cla1;
        end
```

```
if z1==0 & z2==-1 & z11==0 & z22==1
    mp(i,j)=qm*clb1;
end

if z1==0 & (z2==h | z2==0) & z11==0 & z22==1
    mp(i,j)=qm*clb1;
end

if z1==0 & z2>0 & z2<h & z11==0 & z22==z2+1
    mp(i,j)=qm*clb1;
end

if z1==0 & z2>0 & z2<h & z11==0 & z22==-1
    mp(i,j)=qm*clc1;
end

if z1==0 & z2>0 & z2<h & z11==0 & z22==0
    mp(i,j)=qm*cla1;
end

%============================================
if z1==0 & z2==-1 & z11==1 & z22==-1
    mp(i,j)=(1-qm)*clc2;
end

if z1==0 & (z2==h | z2==0) & z11==1 & z22==-1
    mp(i,j)=(1-qm)*clc2;
end

if z1==0 & z2==-1 & z11==1 & z22==0
    mp(i,j)=(1-qm)*cla2;
end

if z1==0 & (z2==h | z2==0) & z11==1 & z22==0
    mp(i,j)=(1-qm)*cla2;
end

if z1==0 & z2==-1 & z11==1 & z22==1
    mp(i,j)=(1-qm)*clb2;
end

if z1==0 & (z2==h | z2==0) & z11==1 & z22==1
    mp(i,j)=(1-qm)*clb2;
end

if z1==0 & z2>0 & z2<h & z11==1 & z22==z2+1
    mp(i,j)=(1-qm)*clb2;
end

if z1==0 & z2>0 & z2<h & z11==1 & z22==-1
    mp(i,j)=(1-qm)*clc2;
end
```

```
if z1==0 & z2>0 & z2<h & z11==1 & z22==0
    mp(i,j)=(1-qm)*cla2;
end


%==================================================

if z1==1 & z2==-1 & z11==0 & z22==-1
    mp(i,j)=qm*PE*clc1;
end


if z1==1 & z2==-1 & z11==0 & z22==0
    mp(i,j)=qm*PE*cla1;
end


if z1==1 & z2==-1 & z11==0 & z22==1
    mp(i,j)=qm*PE*clb1;
end


if z1==1 & z2==-1 & z11==1 & z22==-1
    mp(i,j)=(1-qm)*PE*clc2;
end


if z1==1 & z2==-1 & z11==1 & z22==0
    mp(i,j)=(1-qm)*PE*cla2;
end


if z1==1 & z2==-1 & z11==1 & z22==1
    mp(i,j)=(1-qm)*PE*clb2;
end


if z1==1 & z2==-1 & z11==2 & z22==-1
    mp(i,j)=(1-PE)*clc2;
end


if z1==1 & z2==-1 & z11==2 & z22==0
    mp(i,j)=(1-PE)*cla2;
end


if z1==1 & z2==-1 & z11==2 & z22==1
    mp(i,j)=(1-PE)*clb2;
end

%=================================================
```

```
if z1==1 & z2==0 & z11==2 & z22==-1
    mp(i,j)=clc2;
end


if z1==1 & z2==0 & z11==2 & z22==0
    mp(i,j)=cla2;
end


if z1==1 & z2==0 & z11==2 & z22==1
    mp(i,j)=clb2;
end

%================================================

if z1==1 & z2>0 & z2<h & z11==2 & z22==-1
    mp(i,j)=clc2;
end


if z1==1 & z2>0 & z2<h & z11==2 & z22==z2+1
    mp(i,j)=clb2;
end


if z1==1 & z2>0 & z2<h & z11==2 & z22==0
    mp(i,j)=cla2;
end

%================================================

if z1==1 & z2==h & z11==0 & z22==-1
    mp(i,j)=PE*qm*clc1;
end


if z1==1 & z2==h & z11==0 & z22==0
    mp(i,j)=PE*qm*cla1;
end


if z1==1 & z2==h & z11==0 & z22==1
    mp(i,j)=PE*qm*clb1;
end


if z1==1 & z2==h & z11==1 & z22==-1
    mp(i,j)=PE*(1-qm)*clc2;
end
```

```
if z1==1 & z2==h & z11==1 & z22==0
    mp(i,j)=PE*(1-qm)*cla2;
end


if z1==1 & z2==h & z11==1 & z22==1
    mp(i,j)=PE*(1-qm)*clb2;
end


if z1==1 & z2==h & z11==2 & z22==-1
    mp(i,j)=(1-PE)*clc2;
end


if z1==1 & z2==h & z11==2 & z22==0
    mp(i,j)=(1-PE)*cla2;
end


if z1==1 & z2==h & z11==2 & z22==1
    mp(i,j)=(1-PE)*clb2;
end

%=================================================

if z1==2 & z2==-1 & z11==0 & z22==-1
    mp(i,j)=qm*PE*clc1;
end


if z1==2 & z2==-1 & z11==0 & z22==0
    mp(i,j)=qm*PE*cla1;
end


if z1==2 & z2==-1 & z11==0 & z22==1
    mp(i,j)=qm*PE*clb1;
end


if z1==2 & z2==-1 & z11==1 & z22==-1
    mp(i,j)=(1-qm)*PE*clc2;
end


if z1==2 & z2==-1 & z11==1 & z22==0
    mp(i,j)=(1-qm)*PE*cla2;
end


if z1==2 & z2==-1 & z11==1 & z22==1
    mp(i,j)=(1-qm)*PE*clb2;
```

```
end


if z1==2 & z2==-1 & z11==2 & z22==-1
    mp(i,j)=(1-PE)*clc2;
end


if z1==2 & z2==-1 & z11==2 & z22==0
    mp(i,j)=(1-PE)*cla2;
end


if z1==2 & z2==-1 & z11==2 & z22==1
    mp(i,j)=(1-PE)*clb2;
end

%=================================================

if z1==2 & z2==0 & z11==2 & z22==-1
    mp(i,j)=clc2;
end


if z1==2 & z2==0 & z11==2 & z22==0
    mp(i,j)=cla2;
end


if z1==2 & z2==0 & z11==2 & z22==1
    mp(i,j)=clb2;
end

%=================================================


if z1==2 & z2>0 & z2<h & z11==2 & z22==-1
    mp(i,j)=clc2;
end


if z1==2 & z2>0 & z2<h & z11==2 & z22==z2+1
    mp(i,j)=clb2;
end


if z1==2 & z2>0 & z2<h & z11==2 & z22==0
    mp(i,j)=cla2;
end

%=================================================

if z1==2 & z2==h & z11==0 & z22==-1
```

```
            mp(i,j)=PE*qm*clc1;
        end


        if z1==2 & z2==h & z11==0 & z22==0
            mp(i,j)=PE*qm*cla1;
        end


        if z1==2 & z2==h & z11==0 & z22==1
            mp(i,j)=PE*qm*clb1;
        end


        if z1==2 & z2==h & z11==1 & z22==-1
            mp(i,j)=PE*(1-qm)*clc2;
        end


        if z1==2 & z2==h & z11==1 & z22==0
            mp(i,j)=PE*(1-qm)*cla2;
        end


        if z1==2 & z2==h & z11==1 & z22==1
            mp(i,j)=PE*(1-qm)*clb2;
        end


        if z1==2 & z2==h & z11==2 & z22==-1
            mp(i,j)=(1-PE)*clc2;
        end


        if z1==2 & z2==h & z11==2 & z22==0
            mp(i,j)=(1-PE)*cla2;
        end


        if z1==2 & z2==h & z11==2 & z22==1
            mp(i,j)=(1-PE)*clb2;
        end

        %=====================================================


    end
end
%==============
mp0=mp;
tamanho=size(mp);
AA=transpose(mp)-eye(tamanho(1,1));
AA(tamanho(1,1),:)=ones(1,tamanho(1,1));
```

```
BB=zeros(tamanho(1,1),1);
BB(tamanho(1,1),1)=1;

y=AA\BB;
y=y';



CUSTO=zeros(1,3*(h+2));
mL=zeros(1,3*(h+2));
conta=1;
for i=0:2
    for j=-1:h

        ta=size(y);
        ta1=ta(1,2);
        ta2=ta1/3;

        if i==0 & (j==-1 | j==h)



            if j==-1
                pa1=0;
                for m1=1:1:ta1
                    pa=mp0(m1,1)*y(m1);
                    pa1=pa+pa1;
                end


                pa2=0;
                for m1=[2 ta2 ta2+2 2*ta2 2*ta2+2 3*ta2]
                    pa=mp0(m1,1)*y(m1);
                    pa2=pa+pa2;
                end
                r1=pa2/(pa1+r); %m



                pa3=0;
                for m1=[1 ta2+1 2*ta2+1]
                    pa=mp0(m1,1)*y(m1);
                    pa3=pa+pa3;
                end
                r2=pa3/(pa1+r); %m

            end

            if j==h

                pa1=0;
                for m1=1:1:ta1
                    pa=mp0(m1,ta2)*y(m1);
```

```
                    pa1=pa+pa1;
                end


                pa2=0;

                for m1=[2 ta2 ta2+2 2*ta2 2*ta2+2 3*ta2]
                    pa=mp0(m1,ta2)*y(m1);
                    pa2=pa+pa2;
                end
                r1=pa2/(pa1+r);

                pa3=0;
                for m1=[1 ta2+1 2*ta2+1]
                    pa=mp0(m1,ta2)*y(m1);
                    pa3=pa+pa3;
                end
                r2=pa3/(pa1+r); %m


            end

            r3=1-r2-r1;

            if j==-1

costaa=(ci+cd+ cac+(1-cdf('norm',LE,mi1,dp)+cdf('norm',-LE,mi1,dp))*(m-1)*cn)*r3;
costab=(ci+cd+ cac+(1-cdf('norm',LE,mi1,dp)+cdf('norm',-LE,mi1,dp))*(m-1)*cn)*r2;
costac=(ci+cd+ cac+(1-cdf('norm',LE,mi1,dp)+cdf('norm',-LE,mi1,dp))*(m-1)*cn)*r1;
costa=costaa+costab+costac;

                CUSTO(conta)=costa;
                mL(conta)=m*r3+m*r1+m*r2;
            end

            if j==h


costaa=(ci+cd+ cap+(1-cdf('norm',LE,mi1,dp)+cdf('norm',-LE,mi1,dp))*(m-1)*cn)*r3;
costab=(ci+cd+ cap+(1-cdf('norm',LE,mi1,dp)+cdf('norm',-LE,mi1,dp))*(m-1)*cn)*r2;
costac=(ci+cd+ cap+(1-cdf('norm',LE,mi1,dp)+cdf('norm',-LE,mi1,dp))*(m-1)*cn)*r1;
costa=costaa+costab+costac;


                CUSTO(conta)=costa;
                mL(conta)=m*r3+m*r1+m*r2;
            end

        end


       if i==0 & j>=0 & j<h
```

```
                        pa1=0;
                        for m1=1:1:ta1
                            pa=mp0(m1,j+2)*y(m1);
                            pa1=pa+pa1;
                        end

                        pa2=0;

                        for m1=[2 ta2 ta2+2 2*ta2 2*ta2+2 3*ta2]
                            pa=mp0(m1,j+2)*y(m1);
                            pa2=pa+pa2;
                        end
                        r1=pa2/(pa1+r);

                        pa3=0;
                        for m1=[1 ta2+1 2*ta2+1]
                            pa=mp0(m1,j+2)*y(m1);
                            pa3=pa+pa3;
                        end
                        r2=pa3/(pa1+r); %m


                        r3=1-r2-r1;


costaa=(ci+cd+(1-cdf('norm',LE,mi1,dp)+cdf('norm',-LE,mi1,dp))*(m-1)*cn)*r3;
costab=(ci+cd+(1-cdf('norm',LE,mi1,dp)+cdf('norm',-LE,mi1,dp))*(m-1)*cn)*r2;
costac=(ci+cd+(1-cdf('norm',LE,mi1,dp)+cdf('norm',-LE,mi1,dp))*(m-1)*cn)*r1;
costa=costaa+costab+costac;

                        CUSTO(conta)=costa;

                        mL(conta)=m*r3+m*r1+m*r2;
                    end

                   if i==1 & (j==-1 |j==h)


                        if j==-1
                            pa1=0;
                            for m1=1:1:ta1
                                pa=mp0(m1,ta2+1)*y(m1);
                                pa1=pa+pa1;
                            end


                            pa2=0;
                            for m1=[2 ta2 ta2+2 2*ta2 2*ta2+2 3*ta2]
                                pa=mp0(m1,ta2+1)*y(m1);
                                pa2=pa+pa2;
                            end
                            r1=pa2/(pa1+r); %m
```

```
            pa3=0;
            for m1=[1 ta2+1 2*ta2+1]
                pa=mp0(m1,ta2+1)*y(m1);
                pa3=pa+pa3;
            end
            r2=pa3/(pa1+r); %m

        end

        if j==h

            pa1=0;
            for m1=1:1:ta1
                pa=mp0(m1,2*ta2)*y(m1);
                pa1=pa+pa1;
            end


            pa2=0;

            for m1=[2 ta2 ta2+2 2*ta2 2*ta2+2 3*ta2]
                pa=mp0(m1,2*ta2)*y(m1);
                pa2=pa+pa2;
            end
            r1=pa2/(pa1+r); %m
            pa3=0;
            for m1=[1 ta2+1 2*ta2+1]
                pa=mp0(m1,2*ta2)*y(m1);
                pa3=pa+pa3;
            end
            r2=pa3/(pa1+r); %m



        end

        r3=1-r2-r1;

        p1=(1-cdf('norm',LE,mi1,dp)+cdf('norm',-LE,mi1,dp));
        p2=(1-cdf('norm',LE,mi2,dp)+cdf('norm',-LE,mi2,dp));


        n1=0;
        for vt=1:m
pt=((1-pi)^(vt-1))*pi*((vt-1)*p1*cn+(m-vt)*p2*cn)/(1-qm);
            n1 =pt+n1;
        end

        n2=0;
        for vt=1:m
 pt=((1-pi)^(vt-1))*pi*((vt-1)*p1*cn+(m-vt)*p2*cn)/(1-qm);
            n2 =pt+n2;
        end
```

```
      n3=0;
      for vt=1:m
pt=((1-pi)^(vt-1))*pi*((vt-1)*p1*cn+(m-vt)*p2*cn)/(1-qm);
          n3 =pt+n3;
      end

      if j==-1


          costaa=(n1+ci+cd+cac)*r3;
          costab=(n2+ci+cd+cac)*r1;
          costac=(n3+ci+cd+cac)*r2;

          costa=costaa+costab+costac;

          CUSTO(conta)=costa;
          mL(conta)=m*r3+m*r1+m*r2;
      end

      if j==h


          costaa=(n1+ci+cd+cap)*r3;
          costab=(n2+ci+cd+cap)*r1;
          costac=(n3+ci+cd+cap)*r2;

          costa=costaa+costab+costac;



          CUSTO(conta)=costa;
          mL(conta)=m*r3+m*r1+m*r2;
      end

  end

  if i==1 & j>=0 & j<h


      pa1=0;
      for m1=1:1:ta1
          pa=mp0(m1,j+ta2+2)*y(m1);
          pa1=pa+pa1;
      end


      pa2=0;

      for m1=[2 ta2 ta2+2 2*ta2 2*ta2+2 3*ta2]
          pa=mp0(m1,j+ta2+2)*y(m1);
          pa2=pa+pa2;
```

```
    end
    r1=pa2/(pa1+r);
    pa3=0;
    for m1=[1 ta2+1 2*ta2+1]
        pa=mp0(m1,j+ta2+2)*y(m1);
        pa3=pa+pa3;
    end
    r2=pa3/(pa1+r); %m


    r3=1-r2-r1;

    p1=(1-cdf('norm',LE,mi1,dp)+cdf('norm',-LE,mi1,dp));
    p2=(1-cdf('norm',LE,mi2,dp)+cdf('norm',-LE,mi2,dp));


    n1=0;
    for vt=1:m
  pt=((1-pi)^(vt-1))*pi*((vt-1)*p1*cn+(m-vt)*p2*cn)/(1-qm);
        n1 =pt+n1;
    end

    n2=0;
    for vt=1:m
 pt=((1-pi)^(vt-1))*pi*((vt-1)*p1*cn+(m-vt)*p2*cn)/(1-qm);
        n2 =pt+n2;
    end


    n3=0;
    for vt=1:m
  pt=((1-pi)^(vt-1))*pi*((vt-1)*p1*cn+(m-vt)*p2*cn)/(1-qm);
        n3 =pt+n3;
    end

    costaa=(n1+ci+cd)*r3;
    costab=(n2+ci+cd)*r1;
    costac=(n3+ci+cd)*r2;

    costa=costaa+costab+costac;

    CUSTO(conta)=costa;
    mL(conta)=m*r3+m*r1+m*r2;
end
%********************

if i==2 & (j==-1 | j==h)

    if j==-1
        pa1=0;
        for m1=1:1:ta1
            pa=mp0(m1,2*ta2+1)*y(m1);
            pa1=pa+pa1;
```

```
    end


    pa2=0;
    for m1=[2 ta2 ta2+2 2*ta2 2*ta2+2 3*ta2]
        pa=mp0(m1,2*ta2+1)*y(m1);
        pa2=pa+pa2;
    end
    r1=pa2/(pa1+r); %m



    pa3=0;
    for m1=[1 ta2+1 2*ta2+1]
        pa=mp0(m1,2*ta2+1)*y(m1);
        pa3=pa+pa3;
    end
    r2=pa3/(pa1+r); %m

end

if j==h

    pa1=0;
    for m1=1:1:ta1
        pa=mp0(m1,3*ta2)*y(m1);
        pa1=pa+pa1;
    end


    pa2=0;

    for m1=[2 ta2 ta2+2 2*ta2 2*ta2+2 3*ta2]
        pa=mp0(m1,3*ta2)*y(m1);
        pa2=pa+pa2;
    end
    r1=pa2/(pa1+r);
    pa3=0;
    for m1=[1 ta2+1 2*ta2+1]
        pa=mp0(m1,3*ta2)*y(m1);
        pa3=pa+pa3;
    end
    r2=pa3/(pa1+r); %m



end

r3=1-r2-r1;


if j==-1
```

```
costaa=(ci+cd+ cac+(1-cdf('norm',LE,mi2,dp)+cdf('norm',-LE,mi2,dp))*(m-1)*cn)*r3;
costab=(ci+cd+ cac+(1-cdf('norm',LE,mi2,dp)+cdf('norm',-LE,mi2,dp))*(m-1)*cn)*r2;
costac=(ci+cd+ cac+(1-cdf('norm',LE,mi2,dp)+cdf('norm',-LE,mi2,dp))*(m-1)*cn)*r1;
costa=costaa+costab+costac;


                                    CUSTO(conta)=costa;
                                    mL(conta)=m*r3+m*r1+m*r2;
                                end

                                if j==h


costaa=(ci+cd+ cap+(1-cdf('norm',LE,mi2,dp)+cdf('norm',-LE,mi2,dp))*(m-1)*cn)*r3;
costab=(ci+cd+ cap+(1-cdf('norm',LE,mi2,dp)+cdf('norm',-LE,mi2,dp))*(m-1)*cn)*r2;
costac=(ci+cd+ cap+(1-cdf('norm',LE,mi2,dp)+cdf('norm',-LE,mi2,dp))*(m-1)*cn)*r1;
costa=costaa+costab+costac;


                                    CUSTO(conta)=costa;
                                    mL(conta)=m*r3+m*r1+m*r2;
                                end

                            end


                            if i==2 & j>=0 & j<h

                                pa1=0;
                                for m1=1:1:ta1
                                    pa=mp0(m1,j+2*ta2+2)*y(m1);
                                    pa1=pa+pa1;
                                end


                                pa2=0;
                                for m1=[2 ta2 ta2+2 2*ta2 2*ta2+2 3*ta2]
                                    pa=mp0(m1,j+2*ta2+2)*y(m1);
                                    pa2=pa+pa2;
                                end
                                r1=pa2/(pa1+r);
                                pa3=0;
                                for m1=[1 ta2+1 2*ta2+1]
                                    pa=mp0(m1,j+2*ta2+2)*y(m1);
                                    pa3=pa+pa3;
                                end
                                r2=pa3/(pa1+r); %m


                                r3=1-r2-r1;
```

```
costaa=(ci+cd+(1-cdf('norm',LE,mi2,dp)+cdf('norm',-LE,mi2,dp))*(m-1)*cn)*r3;
costab=(ci+cd+(1-cdf('norm',LE,mi2,dp)+cdf('norm',-LE,mi2,dp))*(m-1)*cn)*r2;
costac=(ci+cd+(1-cdf('norm',LE,mi2,dp)+cdf('norm',-LE,mi2,dp))*(m-1)*cn)*r1;
costa=costaa+costab+costac;

                                CUSTO(conta)=costa;

                                mL(conta)=m*r3+m*r1+m*r2;
                            end

                            conta =conta+1;
                        end

                    end

                    VC=y*CUSTO';
                    tciclo=y*mL';
                    EK=VC/(tciclo-1);


                    vcusto(cc)=EK;

                    bm(cc)=m;
                    bh(cc)=h;
                    bA(cc)=A;
                    bB(cc)=B;
                    bPE(cc)=PE;
                    cc=cc+1;
                end

            end
        end
    end
end
[customin,posmin]=min(vcusto);

m0=bm(posmin)
h0=bh(posmin)
A0=bA(posmin)
B0=bB(posmin)
PE0=bPE(posmin)
costominA=customin



toc;

Resultados:

m0 =
```

```
    24


h0 =

     3


A0 =

   0.800000000000000


B0 =

   1.600000000000000


PE0 =

   0.840000000000000


custominA =

   1.376913846109292
```