

# Reduzindo o Custo Computacional do Algoritmo das Projeções Sucessivas para Problemas de Classificação Utilizando o Lema de Inversão de Matrizes

**Anderson da Silva Soares**

Programa de Pós-Graduação em Engenharia Eletrônica e Computação, ITA  
12228-900, São José dos Campos, SP  
E-mail: engsoares@gmail.com

**Roberto Kawakami Harrop Galvão**

Divisão de Engenharia Eletrônica, ITA  
12228-900, São José dos Campos, SP  
E-mail: kawakami@ita.br

**Resumo:** *Técnicas de seleção de variáveis como o Algoritmo das Projeções Sucessivas (APS) podem ser empregadas em problemas de classificação com o objetivo de reduzir a dimensionalidade do problema. Neste trabalho é proposto o uso do lema de inversão de matrizes para reduzir o custo computacional do APS. O ganho computacional obtido pela modificação é ilustrado em um exemplo de detecção de falhas de um sistema industrial simulado. Dados do software de simulação Tennessee Eastman Process foram utilizados com o sistema operando com e sem falha. Utilizando o APS com a modificação proposta, foi possível obter uma economia de até 63% do tempo computacional sem perda de desempenho em termos de taxa de acerto de classificação.*

**Palavras-Chave:** *Algoritmo das Projeções Sucessivas, classificação de padrões, seleção de variáveis, lema de inversão de matrizes, eficiência computacional.*

## 1 Introdução

A seleção de variáveis em problemas de classificação tem por objetivo obter um subconjunto de atributos não-redundantes entre si que melhor representem os objetos a serem classificados [8].

O Algoritmo das Projeções Sucessivas (APS) foi proposto em [1] como uma técnica de seleção de variáveis para minimizar problemas de colinearidade em Regressão Linear Múltipla. O algoritmo foi posteriormente adaptado para tratar problemas de classificação por Análise Discriminante Linear [2], tendo proporcionado taxas de acerto superiores às obtidas por métodos baseados em análise de componentes principais ou algoritmos genéticos [2, 9]. O APS compreende duas fases [2]. Dada uma matriz  $\mathbf{X}$  de dados de treinamento com  $N$  linhas (objetos) e  $K$  colunas (variáveis), a fase 1 consiste de operações de projeção envolvendo as colunas de  $\mathbf{X}$ . Estas projeções são utilizadas para gerar  $K$  cadeias de variáveis candidatas. Na fase 2, cada uma das cadeias é usada para definir subconjuntos de variáveis a serem avaliados por meio de uma métrica baseada na distância de Mahalanobis [4]. Os subconjuntos são gerados incluindo-se um número progressivamente maior de variáveis, na ordem estabelecida dentro da cadeia.

Em termos de custo computacional, a fase 2 do APS requer um tempo de execução consideravelmente maior que o da fase 1. Isso se deve à necessidade de calcular, para cada subconjunto de variáveis, a inversa da matriz de covariância conjunta a ser usada no cálculo da distância de Mahalanobis. Vale ressaltar que, em aplicações envolvendo um número elevado de variáveis, o tempo de execução pode se tornar proibitivo. Por essa razão, propõe-se neste trabalho uma técnica de atualização da inversa da matriz de covariância conjunta por meio do lema de inversão de matrizes para reduzir o custo computacional da fase 2 do APS.

Para ilustração apresenta-se um estudo de caso do problema de detecção de falhas. Uma falha é entendida como uma operação anormal do sistema em estudo. A detecção de falhas pode ser formulada como um problema de classificação no qual devem ser distinguidas duas classes:  $\omega_0$  : Operação normal e  $\omega_1$  : Falha.

O restante do texto está organizado da seguinte forma. Na seção 2 é feita uma revisão sobre o algoritmo APS em sua versão para problemas de classificação. Na seção 3 é apresentada a proposta de modificação utilizando o lema de inversão de matrizes na fase 2 do APS. Na seção 4 é descrito o estudo de caso de detecção de falhas e as configurações para medida do tempo de execução antes e depois da alteração proposta. Na seção 5 são apresentados os resultados de classificação e de custo computacional. Finalmente na seção 6 são resumidas as conclusões do trabalho.

## 2 O Algoritmo APS para Problemas de Classificação

O algoritmo APS faz uso de dois conjuntos de dados: um conjunto de treinamento com  $N$  objetos e outro conjunto de validação com  $N_v$  objetos. Na fase 1, a partir de uma variável inicial  $x_k (k = 1, \dots, K)$ , cada elemento da cadeia é escolhido de modo a possuir a menor colinearidade em relação às variáveis anteriores, até que se complete um tamanho  $M$  pré-estabelecido. Como resultado, obtêm-se  $K$  cadeias de  $M$  variáveis cada uma. Detalhes desse procedimento estão descritos em [1, 2]. Vale salientar que  $M$  é limitado pela seguinte restrição:

$$M \leq \min(N - c, K) \quad (1)$$

em que  $c$  é a quantidade de classes consideradas no problema [4].

Na fase 2 o algoritmo APS utiliza o conjunto de validação para avaliar os subconjuntos extraídos das cadeias geradas na fase 1, conforme a seqüência de passos explicitada abaixo.

---

### 1 Avaliação dos subconjuntos de variáveis - Fase 2 do APS

---

1. Faça  $k = 1$
2. Enquanto  $k < K$
3.     Faça  $m = 1$
4.     Enquanto  $m < M$
5.         Seja  $\mathcal{X}_{km}$  um subconjunto de variáveis formado pelos  $m$  primeiros elementos da  $k$ -ésima cadeia gerada na fase 1.
6.         Calcule a inversa  $\mathbf{S}_{km}^{-1}$  da matriz de covariância conjunta com as variáveis contidas em  $\mathcal{X}_{km}$ .
7.         Empregando as variáveis contidas em  $\mathcal{X}_{km}$ , calcule a distância de Mahalanobis de cada objeto do conjunto de validação ao centro de cada uma das classes consideradas, de acordo com a Equação (2):

$$r_{km}^2(\mathbf{x}_n, \bar{\mathbf{x}}_j) = (\mathbf{x}_n - \bar{\mathbf{x}}_j)^T \mathbf{S}_{km}^{-1} (\mathbf{x}_n - \bar{\mathbf{x}}_j); \quad n = 1, \dots, N_v; \quad j = 1, \dots, c \quad (2)$$

em que  $x_n$  corresponde ao  $n$ -ésimo objeto do conjunto de validação e  $\bar{x}_j$  corresponde à média amostral da  $j$ -ésima classe.

8.         Calcule o custo associado a cada objeto do conjunto de validação conforme a Equação (3):

$$g_{n,km} = \frac{r_{km}^2(\mathbf{x}_n, \bar{\mathbf{x}}_{In})}{\min_{Ij \neq In} r_{km}^2(\mathbf{x}_n, \bar{\mathbf{x}}_{Ij})}; \quad n = 1, \dots, N_v \quad (3)$$

em que  $In \in \{1, \dots, c\}$  é o índice da classe do objeto  $\mathbf{x}_n$

9.         Calcule o custo médio  $G_{km}$  conforme a Equação (4)

$$G_{km} = \frac{1}{N_v} \sum_{n=1}^{N_v} g_{n,km} \quad (4)$$

10.     Faça  $m = m + 1$
  11.     Fim Enquanto  $m < M$
  12.     Faça  $k = k + 1$
  13.     Fim Enquanto  $k < K$
-

A matriz de covariância conjunta  $\mathbf{S}_{km}$  do passo 6 é calculada como média ponderada das matrizes de covariância. O custo definido na Equação (3) é a razão das distâncias de Mahalanobis do objeto  $\mathbf{x}_n$  ao centro da classe correta e ao centro da classe incorreta mais próxima. Tal expressão pode ser interpretada como o risco de classificação incorreta do objeto em questão [2].

Ao final da fase 2, a escolha do melhor sub-conjunto de variáveis dá-se pelo critério de menor custo médio dos subconjuntos analisados. Vale salientar que o cálculo da Equação (2) é efetuado  $M \times K$  vezes. Com os sucessivos incrementos no número de variáveis (laço “Enquanto  $m < M$ ”), há um acréscimo na dimensão da matriz  $\mathbf{S}_{km}$ . Portanto o custo computacional para se obter a inversa  $\mathbf{S}_{km}^{-1}$ , pode tornar-se elevado à medida que  $m$  é incrementado. Para efeito de comparação, nos testes utilizando os dados e configurações descritos na seção 4, a fase 2 demanda um tempo de execução dez vezes maior do que a fase 1, sendo a inversão de  $\mathbf{S}_{km}$  o passo de maior custo computacional dentro da fase 2 do algoritmo.

### 3 Técnica Proposta

É possível reduzir o custo computacional da fase 2 do APS empregando-se uma abordagem seqüencial para se obter a inversa da matriz  $\mathbf{S}_{k(m+1)}$  com  $m + 1$  variáveis a partir da inversa de  $\mathbf{S}_{km}$ . Para tanto, é proposto o uso do Lema de Inversão de Matrizes.

O lema de inversão de matrizes é uma identidade matemática muito útil no cálculo da inversa de uma matriz, e que surge regularmente na área de controle de sistemas em aplicações do filtro de Kalman e do método dos mínimos quadrados recursivo [6, 7]. Sejam  $\mathbf{A}$ ,  $\mathbf{C}$  matrizes quadradas não-singulares e  $\mathbf{B}$ ,  $\mathbf{D}$  matrizes tais que  $\mathbf{BCD}$  tem a mesma dimensão de  $\mathbf{A}$ . Então o chamado lema de inversão de matrizes é dado na forma [5]

$$(\mathbf{A} + \mathbf{BCD})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{DA}^{-1}\mathbf{B} + \mathbf{C}^{-1})^{-1}\mathbf{DA}^{-1} \quad (5)$$

Seja  $\mathbf{S}_{km}^{-1}$  a inversa da matriz de covariância conjunta para o subconjunto de variáveis  $\{x_{k1}, x_{k2}, \dots, x_{km}\}$  correspondente aos  $m$  primeiros elementos da  $k$ -ésima cadeia. Ao adicionar  $x_{k(m+1)}$ , acrescenta-se um vetor  $\mathbf{s} = [s_1, s_2, \dots, s_N]^T$  em  $\mathbf{S}_{km}$ , de modo que

$$\mathbf{S}_{k(m+1)} = \begin{bmatrix} \mathbf{S}_m & \tilde{\mathbf{s}} \\ \tilde{\mathbf{s}}^T & r \end{bmatrix} \quad (6)$$

em que  $\tilde{\mathbf{s}} = [s_1, s_2, \dots, s_{N-1}]^T$  e  $r = s_N$ . Conforme demonstrado no apêndice A, é possível obter  $\mathbf{S}_{k(m+1)}^{-1}$  como:

$$\mathbf{S}_{k(m+1)}^{-1} = \begin{bmatrix} (\mathbf{S}_m^{-1} - \tilde{\mathbf{s}}_r^{-1}\tilde{\mathbf{s}}^T)^{-1} & -\frac{1}{q}\mathbf{S}_m^{-1}\tilde{\mathbf{s}} \\ -r\tilde{\mathbf{s}}^T(\mathbf{S}_m^{-1} - \tilde{\mathbf{s}}_r^{-1}\tilde{\mathbf{s}}^T)^{-1} & \frac{1}{q} \end{bmatrix}^{-1} \quad (7)$$

em que  $q = r - \tilde{\mathbf{s}}^T\mathbf{S}_m^{-1}\tilde{\mathbf{s}}$ .

Aplicando o lema de inversão de matrizes (Equação 5) para resolver  $(\mathbf{S}_m^{-1} - \tilde{\mathbf{s}}_r^{-1}\tilde{\mathbf{s}}^T)^{-1}$ , em que  $\mathbf{A} = \mathbf{S}_{km}^{-1}$ ,  $\mathbf{B} = \frac{1}{r}$ ,  $\mathbf{C} = \tilde{\mathbf{s}}$  e  $\mathbf{D} = \tilde{\mathbf{s}}^T$  obtém-se:

$$\mathbf{S}_{k(m+1)}^{-1} = \begin{bmatrix} \mathbf{S}_m^{-1} + \frac{1}{q}\mathbf{S}_m^{-1}\tilde{\mathbf{s}}\tilde{\mathbf{s}}^T\mathbf{S}_m^{-1} & -\frac{1}{q}\mathbf{S}_m^{-1}\tilde{\mathbf{s}} \\ -\frac{1}{q}\tilde{\mathbf{s}}^T\mathbf{S}_m^{-1} & \frac{1}{q} \end{bmatrix}^{-1} \quad (8)$$

Logo, é possível obter  $\mathbf{S}_{k(m+1)}^{-1}$  a partir de  $\mathbf{S}_{km}^{-1}$  sem a necessidade de refazer todo o cálculo da inversa a medida que  $m$  é incrementado. Com isso, o custo computacional da fase 2 do algoritmo APS pode ser reduzido.

### 4 Materiais e Métodos

Para fins de medida de tempo computacional, foi utilizado um notebook com processador *AMD Athlon Dual-Core* de 1.90Hz e 2GB de memória RAM. O ambiente de desenvolvimento foi o *Matlab* em sua versão 2008.

Em particular, foram utilizados os mesmos dados do trabalho apresentado em [10] gerados por meio de simulação do *Tennessee Eastman process* (TEP). O software é reconhecido na literatura de detecção de falhas como problema de *benchmark* e utiliza técnicas de controle multivariável de processos [3]. O sistema em questão pode ser brevemente descrito como uma arranjo reator-separador-reciclo para geração de diferentes produtos em uma determinada proporção. O objetivo é a produção de materiais que serão referenciados como produtos PA, PC, PD e PE.

O APS é empregado para escolher variáveis com atrasos de tempo a serem considerados. As medidas das grandezas da tabela 1 são realizadas com intervalos regulares de tempo, com período de amostragem  $T$ . Como entrada do classificador emprega-se os valores passados de  $w_1, w_2, \dots, w_n$ . O objeto a ser classificado no instante de tempo  $t$  seria então

$$\mathbf{x}(t) = \begin{bmatrix} w_1(t) & w_1(t-T) & w_1(t-(p-1)T) & \dots \\ w_2(t) & w_1(t-T) & w_1(t-(p-1)T) & \dots \\ w_n(t) & w_n(t-T) & w_n(t-(p-1)T) \end{bmatrix}$$

sendo  $p$  uma constante escolhida de modo a definir a janela de tempo a ser empregada.

Tabela 1: Grandezas monitoradas na simulação.

Grandeza	Descrição
$w_1$	Vazão de alimentação para o produto PA.
$w_2$	Vazão de alimentação para o produto PD.
$w_3$	Vazão de alimentação para o produto PE.
$w_4$	Vazão de alimentação para os produtos PA e PC.
$w_5$	Vazão de reciclo do compressor.
$w_6$	Vazão de alimentação do reator
$w_7$	Pressão do reator.
$w_8$	Nível do reator.
$w_9$	Temperatura do reator.
$w_{10}$	Vazão do depurador.
$w_{11}$	Temperatura do separador de produtos.
$w_{12}$	Nível do separador de produtos.
$w_{13}$	Pressão do separador de produtos.
$w_{14}$	Vazão de líquido na saída do separador.
$w_{15}$	Nível da retificadora.
$w_{16}$	Pressão da retificadora.
$w_{17}$	Vazão de líquido na saída da retificadora.
$w_{18}$	Temperatura da retificadora.
$w_{19}$	Vazão de vapor para a retificadora.
$w_{20}$	Trabalho do compressor.
$w_{21}$	Temperatura de saída da água de resfriamento para o reator.
$w_{22}$	Temperatura de saída da água de resfriamento para o condensador.

O simulador é capaz de simular condições de falha e operação normal por meio do monitoramento de 22 grandezas descritas na Tabela 1. Os dados se referem a 8 tipos de falhas no processo de produção por meio da manipulação de algumas variáveis conforme a Tabela 2. Cada caso de falha inclui 100 objetos obtidos a partir de operação normal e 100 objetos a partir de operação com falha. Os dados foram separados em três conjuntos: treinamento e validação a serem utilizados no algoritmo APS e ao final o classificador é obtido por Análise Discriminante Linear aplicado a um terceiro conjunto de teste de classificação onde será medida a acurácia do classificador.

#### 4.1 Métricas de Avaliação

No processo de detecção de falhas quatro possíveis resultados são previstos: Verdadeiro Positivo (VP) que consiste em detectar uma falha quando existe falha. Falso Positivo (FP) que consiste

Tabela 2: Falhas simuladas.

Código da falha	Grandeza manipulada
IDV1	Razão de alimentação entre PA e PC
IDV2	Composição do produto PB
IDV3	Temperatura de alimentação do PD
IDV4	Temperatura da água no reator
IDV5	Temperatura da água no condensador
IDV6	Perda de alimentação do produto PA
IDV7	Perda de pressão
IDV8	Composição dos produtos PA, PB, PC

em detectar uma falha quando não existe falha. Verdadeiro Negativo (VN) que consiste em classificar a operação do sistema como normal quando a operação é normal, ou seja, sem falhas e Falso Negativo (FN) que consiste em classificar a operação do sistema como normal quando existe uma falha.

As taxas de VP, FP, VN e FN podem ser usadas para calcular a sensibilidade do classificador de falhas como a sensibilidade  $SEN(\%) = \frac{VP}{VP+FN} \times 100\%$  e a especificidade como  $ESP(\%) = \frac{VN}{VN+FP} \times 100\%$ . A sensibilidade indica o quão bom é a classificação para identificar falhas, enquanto a especificidade indica a confiabilidade do classificador em detectar operação normal. Adicionalmente a taxa de acerto como  $TA(\%) = \frac{VP}{VP+FP} \times 100\%$  também será utilizada.

## 5 Resultados

A Tabela 3 apresenta os resultados de detecção para oito tipos de falhas utilizando o APS. Para cada tipo de falha um classificador foi treinado. Os resultados são referentes ao conjunto de teste. Para a maioria das falhas analisadas (IDV1, IDV2, IDV6, IDV7, IDV8) a detecção foi realizada com taxas de acerto, sensibilidade e especificidade elevadas.

Tabela 3: Resultados da detecção de falhas utilizando o APS.

APS-Classificação								
	IDV1	IDV2	IDV3	IDV4	IDV5	IDV6	IDV7	IDV8
TA(%)	85	100	60	60	40	100	100	100
SEN(%)	70	100	60	50	50	100	100	100
ESP(%)	100	100	60	70	30	100	100	100

Como o APS permite a escolha do número máximo de variáveis ( $M$ ) a serem testadas na fase 2 do algoritmo, foi conduzido um experimento variando  $M$  de 1 ao máximo permitido conforme a Equação (1). O objetivo consiste em comparar o custo computacional da implementação original da fase 2 do APS com a implementação utilizando o lema de inversão de matrizes.

A Figura 1 mostra o resultado comparativo do tempo de execução gasto na implementação original do APS e na implementação proposta na seção 3 para os dados da falha IDV1. Para o caso de  $M = 76$  a modificação na fase 2 do APS resultou em uma redução de 63% do tempo computacional. Resultados similares foram obtidos ao se tratar as demais falhas.

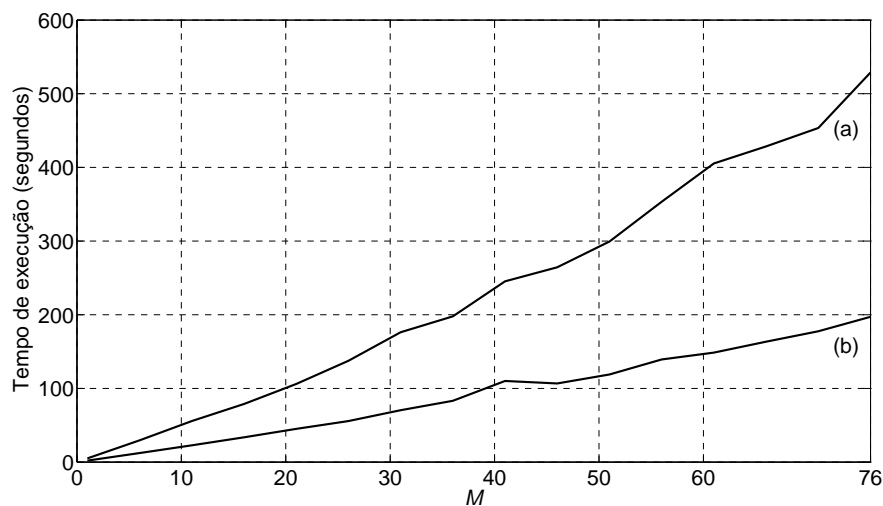


Figura 1: Custo computacional utilizando a implementação original (a) e a implementação com lema de inversão de matrizes (b) para os dados da falha IDV1.

## 6 Conclusões

Este trabalho propôs uma nova implementação do algoritmo de seleção de variáveis APS por meio do uso do lema de inversão de matrizes. Esta implementação foi utilizada na fase 2 do algoritmo, principal limitante em termos de custo computacional. Como exemplo de aplicação, um problema de detecção de falhas em dados de um processo industrial simulado foi utilizado. Como resultado, o ganho computacional na fase 2 do APS chegou a 67%.

Como continuidade do trabalho sugere-se o tratamento do custo computacional do laço “Enquanto  $k < K$ ” no algoritmo apresentado na seção 2 eliminando subconjuntos de variáveis que já foram avaliadas.

## Agradecimentos

Os autores agradecem o apoio financeiro da FAPESP (bolsa de doutorado 2007/57803-7 e projeto temático 2006/58850-6), CNPq (bolsa de produtividade em pesquisa) e CAPES/PROCAD (processo 0081/05-1).

## Referências

- [1] M. C. U. Araújo, T. C. B. Saldanha, R. K. H. Galvão, T. Yoneyama, H. C. Chame, V. Visani. The successive projections algorithm for variable selection in spectroscopic multicomponent analysis. *Chemometrics and Intelligent Laboratory Systems*, 57(2):65–73, 2001.
- [2] M. J. C. Pontes, R. K. H. Galvão, M. C. U. Araújo, P. N. T. Nogueira, O. D. P. Neto, G. E. José, T. C. B. Saldanha. The Successive Projections Algorithm for Spectral Variable Selection in Classification Problems. *Chemometrics and Intelligent Laboratory Systems*, 78(1):11-18, 2005.
- [3] J. J. Downs, E. F. Vogel. A plant-wide industrial-process control problem, *Computers and Chemical Engineering*. *Computers & Chemical Engineering*, 17(3):245-255, 1993.
- [4] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley, New York, 2 edition, 2001.

- [5] W. Hager. Updating the inverse of a matrix. *SIAM Review*, 31(2):221–239, 1989.
- [6] S. Haykin. *Adaptive Filter Theory*. Prentice Hall, 2002.
- [7] J. S. R. Jang, C. T. Sun, E. Mizutani. *Neuro-Fuzzy and Soft Computing*. Prentice Hall, 1997.
- [8] N. Kwak, C. Choi. Input feature selection for classification. *IEEE Transaction on Neural Networks*, 13(1):143-159, 2002.
- [9] F. F. G. Neto, G. Marinoa, M. C. U. Araújo, R. K. H. Galvão, M. J. C. Pontesa, E. P. de Medeiros, R. S. Lima. Classification of edible vegetable oils using square wave voltammetry with multivariate data analysis. *Talanta*, 77(5):1660-1666, 2009.
- [10] X. Pei, Y. Yamashita, M. Yoshida, and S. Matsumoto. Fault detection in chemical processes using discriminant analysis. *Journal of Chemical Engineering of Japan*, 41(1):25–31, 2008.

## A Demonstração da Equação (7)

Seja

$$\mathbf{S}_{k(m+1)} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \quad (9)$$

em que  $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$  tem dimensões  $(r \times r)$ ,  $(r \times s)$ ,  $(s \times r)$  e  $(s \times s)$  respectivamente. Seja ainda  $\mathbf{S}_{k(m+1)}^{-1}$  dado por

$$\mathbf{S}_{k(m+1)}^{-1} = \begin{bmatrix} \mathbf{E} & \mathbf{F} \\ \mathbf{G} & \mathbf{H} \end{bmatrix} \quad (10)$$

em que  $\mathbf{E}, \mathbf{F}, \mathbf{G}, \mathbf{H}$  têm dimensões  $(r \times r)$ ,  $(r \times s)$ ,  $(s \times r)$  e  $(s \times s)$  respectivamente.

Sabe-se que  $\mathbf{S}_{k(m+1)} \mathbf{S}_{k(m+1)}^{-1} = \mathbf{I}$  (matriz identidade), que na forma matricial equivale a

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{E} & \mathbf{F} \\ \mathbf{G} & \mathbf{H} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \quad (11)$$

que equivale a

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{E} & \mathbf{F} \\ \mathbf{G} & \mathbf{H} \end{bmatrix} = \begin{bmatrix} \mathbf{AE} + \mathbf{BG} = \mathbf{I} & \mathbf{AF} + \mathbf{BH} = \mathbf{0} \\ \mathbf{CE} + \mathbf{DG} = \mathbf{0} & \mathbf{CF} + \mathbf{DH} = \mathbf{I} \end{bmatrix} \quad (12)$$

Isolando  $\mathbf{G}$  na segunda linha da primeira coluna da Equação (12) tem-se que  $\mathbf{G} = -\mathbf{D}^{-1}\mathbf{CE}$ . Substituindo  $\mathbf{G}$  na primeira linha da primeira coluna da Equação (12) obtém-se  $\mathbf{E} = (\mathbf{A} - \mathbf{BD}^{-1}\mathbf{C})^{-1}$ . Isolando  $\mathbf{F}$  na primeira linha da segunda coluna da Equação (12) obtém-se  $\mathbf{F} = -\mathbf{A}^{-1}\mathbf{BH}$ . Substituindo  $\mathbf{F}$  de modo a isolar  $\mathbf{H}$  na segunda linha da segunda coluna da Equação (12) obtém-se  $\mathbf{H} = (\mathbf{D} - \mathbf{CA}^{-1}\mathbf{B})^{-1}$ . Substituindo  $\mathbf{E}, \mathbf{F}, \mathbf{G}$  e  $\mathbf{H}$  na Equação (10), segue que

$$\mathbf{S}_{k(m+1)}^{-1} = \begin{bmatrix} (\mathbf{A} - \mathbf{BD}^{-1}\mathbf{C})^{-1} & -\mathbf{ABH} \\ -\mathbf{D}^{-1}\mathbf{CE} & (\mathbf{D} - \mathbf{CA}^{-1}\mathbf{B})^{-1} \end{bmatrix} \quad (13)$$

No mesmo sentido é possível isolar  $\mathbf{H}$  e  $\mathbf{E}$  e substituir na Equação (13) de modo que

$$\mathbf{S}_{k(m+1)}^{-1} = \begin{bmatrix} (\mathbf{A} - \mathbf{BD}^{-1}\mathbf{C})^{-1} & -\mathbf{AB}(\mathbf{D} - \mathbf{CA}^{-1}\mathbf{B})^{-1} \\ -\mathbf{D}^{-1}\mathbf{C}(\mathbf{A} - \mathbf{BD}^{-1}\mathbf{C})^{-1} & (\mathbf{D} - \mathbf{CA}^{-1}\mathbf{B})^{-1} \end{bmatrix} \quad (14)$$