

**UNIVERSIDADE FEDERAL DE MINAS GERAIS  
INSTITUTO DE CIÊNCIAS EXATAS  
PROGRAMA DE MESTRADO EM ESTATÍSTICA**

**THAIZE VIEIRA MARTINS**

**INTERVALOS DE PREVISÃO BOOTSTRAP PARA MODELOS ESTRUTURAIIS**

**Belo Horizonte**

**Mai / 2011**

**THAIZE VIEIRA MARTINS**

**INTERVALOS DE PREVISÃO BOOTSTRAP PARA MODELOS ESTRUTURAIS**

Dissertação apresentada junto ao curso de Mestrado em Estatística da Universidade Federal de Minas Gerais, na área de concentração de Probabilidade e Estatística com a linha de pesquisa em Séries Temporais.

Orientadora: Professora Glaura da Conceição Franco

**Belo Horizonte**

**Maio / 2011**

**THAIZE VIEIRA MARTINS**

**INTERVALOS DE PREVISÃO BOOTSTRAP PARA MODELOS ESTRUTURAIIS**

Dissertação apresentada junto ao curso de Mestrado em Estatística da Universidade Federal de Minas Gerais, na área de concentração de Probabilidade e Estatística com a linha de pesquisa em Séries Temporais.

Orientadora: Professora Glaura da Conceição Franco

**COMISSÃO EXAMINADORA**

---

Professora Ela Mercedes Medrano de Toscano  
Universidade Federal de Minas Gerais

---

Professora Glaura da Conceição Franco  
Universidade Federal de Minas Gerais

**Belo Horizonte, 02 de maio de 2011**

## **Agradecimentos**

Agradeço primeiramente a Deus, à minha família e ao meu noivo Paulo, pelo incentivo, paciência, compreensão e carinho sempre presentes. Destaco ainda, a imensa gratidão pela minha amiga Fernanda, que foi mais do que amiga e companheira desde os preparativos para ingressar no curso, fazendo com que essa caminhada fosse mais fácil e prazerosa.

Agradeço em especial à professora Glaura que norteou todo o processo de elaboração do trabalho acadêmico em questão e me orientou de forma exemplar, transmitindo seus valiosos conhecimentos estatísticos. Agradeço também ao professor Frederico que contribuiu muito para a programação deste projeto e que, juntamente com a minha orientadora, confiou no meu potencial e contribuiu para a minha formação.

Aos colegas do curso, agradeço sinceramente pela convivência, amizade e companheirismo.

Ao Professor Wanderley Ramalho e aos colegas de trabalho da Fundação IPEAD que sempre me apoiaram e me incentivaram para concluir o curso da melhor forma possível.

Agradeço à instituição de ensino, que permeou a aquisição da titulação alcançada por meio do curso ofertado com profissionais dedicados e de grande sabedoria.

Para finalizar, meus sinceros agradecimentos a todos aqueles que de alguma forma doaram um pouco de si para que a conclusão deste trabalho se tornasse possível.

### **Dedicatória**

Dedico esta Dissertação ao meu noivo Paulo, que desde os preparativos para a prova de seleção vem me apoiando e torcendo pelo meu sucesso com uma paciência, dedicação e um amor imensurável tornando-se, junto com a minha família, os principais responsáveis por essa grande realização na minha vida.

*“Quando você quer alguma coisa, todo o universo conspira para que você realize o seu desejo.”*

**Paulo Coelho**

## Resumo

Este trabalho dedica-se à implementação da metodologia de construção de intervalos de previsão *bootstrap* não-paramétrico para as formas de espaço de estados (EE), baseando-se na proposta apresentada em Rodriguez e Ruiz (2009). Os EE são uma forma alternativa de reescrever os modelos estruturais, os quais decompõem as séries temporais em seus componentes não-observáveis (nível, tendência e sazonalidade). Particularmente, este trabalho tem a finalidade de estender a metodologia apresentada por Rodriguez e Ruiz (2009) para modelos estruturais mais complexos, implementando-os na linguagem Ox e comparando os resultados das simulações com o método tradicional de construção de intervalos de previsão assintóticos e também com o procedimento *bootstrap* paramétrico.

Palavras-chave: Forma de espaço de estados, *bootstrap*, sazonalidade, filtro de Kalman.

## **Abstract**

This work is dedicated to the implementation of the methodology to calculate nonparametric bootstrap prediction intervals in state space form (SS), based on the work of Rodriguez and Ruiz (2009). The SS are an alternative way of rewriting the structural models, which decompose the time series in their non-observable components (level, trend and seasonality). Mainly, this work has the interest of extending the methodology proposed by Rodriguez and Ruiz (2009) to more complex structural models, implementing the algorithms in the Ox language and comparing the simulation results with the traditional method of constructing asymptotic prediction intervals and also with the parametric bootstrap procedure.

**Keywords:** State space form, bootstrap, seasonality, Kalman filter.



## Índice de Figuras

Figura 1: Cobertura média dos intervalos construídos para as séries simuladas do MNL .....	37
Figura 2: Amplitude média dos intervalos construídos para as séries simuladas do MNL.....	38
Figura 3: Cobertura média dos intervalos construídos para as séries simuladas do MTL .....	42
Figura 4: Amplitude média dos intervalos construídos para as séries simuladas do MTL .....	42
Figura 5: Cobertura média dos intervalos construídos para as séries simuladas do MEB .....	45
Figura 6: Amplitude média dos intervalos construídos para as séries simuladas do MEB .....	45
Figura 7: Gráfico com os dados da inflação mensal brasileira de jan/1995 – mar/2011.....	46
Figura 8: Histograma da série IPCA (jan/1995 – mar/2010).....	47
Figura 9: Ajuste da série IPCA à uma distribuição Normal (jan/1995 – mar/2010) .....	48
Figura 10: Análise de tendência e sazonalidade da série IPCA (jan/1995 – mar/2010).....	48
Figura 11: FAC e FACP da série IPCA (jan/1995 – mar/2010).....	49
Figura 12: Intervalos de confiança para os métodos de previsão utilizados na série do IPCA .....	50

## Índice de Tabelas

Tabela 1: Estimativas dos hiperparâmetros para as séries simuladas do MNL.....	34
Tabela 2: Intervalos de previsão para as séries simuladas do MNL.....	35
Tabela 3: Estimativas dos hiperparâmetros para as séries simuladas do MTL .....	40
Tabela 4: Intervalos de previsão para as séries simuladas do MTL .....	40
Tabela 5: Estimativas dos hiperparâmetros para as séries simuladas do MEB .....	43
Tabela 6: Intervalos de previsão para as séries simuladas do MEB .....	44
Tabela 7: Estatísticas descritivas da série do IPCA (jan/1995 – mar/2010).....	47
Tabela 8: Estimativas pontuais para os hiperparâmetros da série do IPCA .....	49
Tabela 9: Intervalos de previsão para a série do IPCA (período previsto: abril/2010 – março/2011) ..	50

## Sumário

1. Introdução.....	12
2. Aspectos Metodológicos .....	15
2.1. Modelos Estruturais.....	15
2.1.1. Modelo de Nível Local (MNL).....	16
2.1.2. Modelo de Tendência Linear Local (MTL) .....	16
2.1.3. Modelo Estrutural Básico (MEB).....	17
2.2. Forma de Espaço de Estados (EE).....	17
2.3. Filtro de Kalman.....	19
2.4. Estimação dos hiperparâmetros .....	21
3. <i>Bootstrap</i> .....	22
3.1. Bootstrap em Modelos Estruturais .....	22
3.2. Bootstrap Paramétrico .....	23
3.3. Bootstrap Não-paramétrico .....	23
4. Previsão .....	26
4.1. Previsão pontual .....	26
4.1.1. Previsões pontuais pelo método bootstrap paramétrico .....	26
4.1.2. Previsões pontuais no método bootstrap não-paramétrico.....	28
4.1.3. Previsões pontuais para o cálculo do intervalo assintótico.....	29
4.2. Intervalos de Previsão .....	31
4.2.1. Intervalos de Previsão Assintóticos.....	31
4.2.2. Intervalos de previsão Bootstrap Percentílico .....	31
5. Resultados da simulação .....	33
5.1 Resultados para o MNL.....	34
5.2 Resultados para o MTL .....	39
5.3 Resultados para o MEB .....	42
6. Aplicação a uma série real.....	46
7. Considerações Finais .....	51
Referências Bibliográficas .....	53
Apêndice 1 – Matriz de Covariância.....	55
Apêndice 2 – Equações de Previsão <i>Bootstrap</i> Paramétrico.....	57
Apêndice 3 – Equações de Previsão <i>Bootstrap</i> Não-Paramétrico .....	60
Apêndice 4 – Equações de Previsão Assintótica.....	67
Apêndice 5 – Resultados Rodriguez e Ruiz (2009).....	77
Apêndice 6 – Algoritmo .....	78

## 1. Introdução

A análise de séries temporais é uma área da Estatística dedicada ao estudo de observações obtidas em intervalos regulares de tempo, durante um período específico. Na análise de uma série temporal, primeiramente deseja-se modelar o fenômeno estudado para, a partir daí, descrever o comportamento da série e fazer estimativas.

As séries temporais surgem nas mais variadas áreas de aplicação, como: finanças, marketing, seguros, meteorologia, ciências econômicas, ciências sociais, energia, dentre outras, sendo seu estudo importante no auxílio do processo de tomada de várias decisões. Essas observações podem ser analisadas de forma a tentar entender a estrutura probabilística que as gerou e também no intuito de procurar construir um modelo estatístico a partir do qual seja possível prever valores futuros para a série. Para maiores detalhes, ver Morettin *et al.* (2004).

Dentre as várias metodologias existentes para tratamento de séries temporais destacam-se os modelos estruturais, os quais decompõem as séries em seus componentes não-observáveis de tendência, sazonalidade, ciclo e ruído aleatório. Essa modelagem é abordada tanto no enfoque clássico, quanto no enfoque Bayesiano. Sob o ponto de vista clássico os modelos são conhecidos como modelos estruturais, enquanto que no contexto Bayesiano eles são chamados de modelos dinâmicos. Vale ressaltar que neste trabalho será utilizado somente o enfoque clássico.

A ideia de decompor a série temporal surgiu nos trabalhos de Holt (1957) e Winters (1960), os quais desenvolveram as técnicas de alisamento exponencial. No entanto, somente na década de 60 surgiram alguns trabalhos formalizando a metodologia de modelos estruturais, dentre os quais destacam os de Muth (1960), Theil & Wage (1964) e Nervole & Wage (1964). Já na década de 70 surgiram os primeiros modelos de previsão Bayesianos com Harrison & Stevens (1971, 1976) e no final da década de 80, Andrew Harvey (Harvey, 1989, 1993) e West & Harrison (1997) colaboraram para o avanço dos modelos estruturais, fornecendo uma base sólida para a modelagem de séries temporais por meio do enfoque estrutural.

Para realizar a estimação dos modelos estruturais é preciso encontrar as estimativas das variâncias dos ruídos relacionados a cada componente não-observável, chamadas de

hiperparâmetros. As duas ferramentas básicas para a análise e estimação desses modelos são a forma de espaço de estados (EE) (Harvey, 1989) e o filtro de Kalman (Kalman, 1960).

A forma de espaço de estados foi uma maneira encontrada para se reescrever as expressões da modelagem estrutural, reduzindo-se o número de equações, o que facilita o processo de estimação dos hiperparâmetros. Dessa forma é possível representar a modelagem estrutural com apenas duas equações: a equação das observações (ou de medida) e a equação do estado (ou transição). Além disso, os EE permitem a utilização do filtro de Kalman para a estimação sequencial do vetor de estados e das previsões requeridas.

O filtro de Kalman faz a atualização do vetor de estado e a construção da função de verossimilhança, tornando possível a estimação dos hiperparâmetros do modelo. Dessa forma, ele também permite obter as previsões dos valores futuros das séries, no entanto, vale ressaltar que ele trabalha com a suposição de que as observações têm distribuição normal.

A fim de fazer inferências sobre os hiperparâmetros dos modelos, os intervalos de confiança podem ser construídos por meio de diferentes metodologias. Uma delas é dada pela distribuição assintótica dos hiperparâmetros (Harvey, 1989), que está bem desenvolvida para modelos Gaussianos. Já quando a amostra é pequena ou a distribuição dos ruídos não é conhecida, uma alternativa é a aplicação da técnica *bootstrap* (Efron, 1979). Stoffer & Wall (1991) propuseram um procedimento que possibilita a aplicação da técnica *bootstrap* na forma de espaço de estados, a qual será utilizada neste trabalho.

Após escrever o modelo na forma de espaço de estados e estimar os hiperparâmetros, o interesse é calcular os valores futuros da série com os seus respectivos intervalos de previsão. Por meio do filtro de Kalman é possível obter os valores previstos e a partir daí construir os seus intervalos de previsão assintóticos que assumem que os ruídos apresentam distribuição normal.

A construção dos intervalos de previsão assintóticos apresenta duas limitações:

- 1<sup>a</sup>) Ela não incorpora a incerteza causada pela estimação dos hiperparâmetros;
- 2<sup>a</sup>) A normalidade assumida para as observações futuras pode ser inexata.

Para superar essas limitações Rodriguez e Ruiz (2009) propuseram para os EE o procedimento *bootstrap* não-paramétrico para construir intervalos de previsão diretamente das observações, aplicando-o no modelo mais simples, o qual ajusta apenas o nível da série (Modelo de Nível Local).

Diante da problemática apresentada, o foco principal deste trabalho é implementar a metodologia descrita anteriormente e estendê-la para modelos estruturais mais complexos,

que incorporam componentes de tendência e sazonalidade. Para implementar esse método desenvolvemos um algoritmo na linguagem Ox e, em seguida, realizamos simulações para comparar os nossos resultados com aqueles obtidos no artigo. Adicionalmente, também serão implementados outros métodos de construção de intervalos de previsão, como o assintótico e o *bootstrap* paramétrico, com o objetivo de identificar qual o método mais adequado para contruir intervalos de previsão.

Atualmente existem pacotes computacionais, como o SsfPack do Ox (Koopman, Shephard & Doornik, 1999), que já fazem a estimação dos modelos estruturais e possibilitam a implementação de novas rotinas.

Esta dissertação está organizada da seguinte forma: no Capítulo 2 é apresentado a metodologia utilizada para a elaboração deste trabalho; no Capítulo 3 encontra-se a definição do método *Bootstrap*; o Capítulo 4 é dedicada à discussão dos métodos de cálculo para previsão e os tipos de intervalos adotados neste trabalho; no Capítulo 5 são apresentados os resultados obtidos com as simulações; no Capítulo 6 o método estudado é exemplificado por meio de sua aplicação a uma série real, e, por fim, no Capítulo 7 têm-se as considerações finais.

## 2. Aspectos Metodológicos

Uma série temporal consiste na coleção de dados feita sequencialmente ao longo do tempo, cuja ordem é fundamental, uma vez que as observações vizinhas são geralmente dependentes. Esse conjunto de informações também pode ser definido como a trajetória de um processo estocástico<sup>1</sup>.

O maior interesse dessa área de estudo é analisar e modelar a dependência que as observações apresentam. Para isso, primeiramente são estudadas as características das séries temporais.

Este capítulo apresenta a teoria existente na área de séries temporais sobre modelos estruturais e a forma de espaço de estados utilizada na elaboração deste trabalho.

### 2.1. Modelos Estruturais

A premissa básica dos modelos estruturais consiste em representar a variabilidade de uma série temporal por meio de componentes não-observáveis de tendência, sazonalidade, ciclo e ruído aleatório. Essa ideia de decompor as séries surgiu nos trabalhos de Holt (1957) e Winters (1960) e se consolidou no final da década de 80 com Andrew Harvey (Harvey, 1989, 1993) e West & Harison (1997), como foi discutido no Capítulo 1.

A principal vantagem desses modelos é dada pelo fato de que em vez de ajustar funções que descrevem os componentes não-observáveis aos dados históricos, estes procedimentos procuram identificar os componentes básicos na série e o modelo resultante é obtido a partir da composição desses elementos.

Neste capítulo, os trabalhos de Franco (1998) e Franco *et al.* (2009) foram utilizados como base para o entendimento da definição dos modelos estruturais, suas representações e os métodos de estimação. Para uma dada série temporal,  $y_t$ , os modelos estruturais são escritos da seguinte forma:

$$y_t = \mu_t + \gamma_t + \delta_t + \varepsilon_t \quad t = 1, 2, \dots, T \quad (2.1)$$

---

<sup>1</sup> Um processo estocástico é uma família de variáveis aleatórias indexadas no tempo, que são controladas por leis probabilísticas e assumem valores no espaço de índices T, o qual pode ser discreto ou contínuo.

sendo,

$\mu_t$ : Componente estocástico de Tendência.

$\gamma_t$ : Componente estocástico de Sazonalidade.

$\delta_t$ : Componente estocástico de Ciclo.

$\varepsilon_t$ : Ruído Aleatório, onde  $\varepsilon_t \sim N(0, \sigma_\varepsilon^2)$  e são independentes entre si.

Vale ressaltar que os modelos são construídos baseados nas suposições de independência, homocedasticidade e normalidade dos erros, sendo os principais tipos de modelos apresentados a seguir.

### **2.1.1. Modelo de Nível Local (MNL)**

O MNL, também conhecido como modelo de passeio aleatório adicionado de um ruído, é o modelo mais simples, pois não há inclinação positiva ou negativa. Isto é, a série se movimenta ao longo do tempo sem uma trajetória fixa, pois o nível atual é igual ao nível no período anterior acrescido de um ruído. A forma do modelo é dada por:

$$y_t = \mu_t + \varepsilon_t, \quad \varepsilon_t \sim N(0, \sigma_\varepsilon^2) \quad (2.2)$$

$$\mu_t = \mu_{t-1} + \eta_t, \quad \eta_t \sim N(0, \sigma_\eta^2) \quad (2.3)$$

com  $\varepsilon_t, \eta_t$  não-correlacionados e  $t = 1, 2, \dots, T$ .

### **2.1.2. Modelo de Tendência Linear Local (MTL)**

O MTL é utilizado para nomear as séries temporais que possuem um movimento crescente ou decrescente (tendência) e que também têm um comportamento de passeio aleatório, como pode ser visualizado a seguir:

$$y_t = \mu_t + \varepsilon_t, \quad \varepsilon_t \sim N(0, \sigma_\varepsilon^2) \quad (2.4)$$

$$\mu_t = \mu_{t-1} + \beta_{t-1} + \eta_t, \quad \eta_t \sim N(0, \sigma_\eta^2) \quad (2.5)$$

$$\beta_t = \beta_{t-1} + \xi_t, \quad \xi_t \sim N(0, \sigma_\xi^2) \quad (2.6)$$



com  $\varepsilon_t, \eta_t, \xi_t$  não-correlacionados e  $t = 1, 2, \dots, T$ .

### 2.1.3. Modelo Estrutural Básico (MEB)

O MEB é freqüentemente usado para modelar séries que contenham o componente sazonal, trata-se do MTL acrescido da sazonalidade e é definido na forma:

$$y_t = \mu_t + \gamma_t + \varepsilon_t, \quad \varepsilon_t \sim N(0, \sigma_\varepsilon^2) \quad (2.7)$$

$$\mu_t = \mu_{t-1} + \beta_{t-1} + \eta_t, \quad \eta_t \sim N(0, \sigma_\eta^2) \quad (2.8)$$

$$\beta_t = \beta_{t-1} + \xi_t, \quad \xi_t \sim N(0, \sigma_\xi^2) \quad (2.9)$$

$$\gamma_t = -\gamma_{t-1} - \dots - \gamma_{t-s+1} + \omega_t, \quad \omega_t \sim N(0, \sigma_\omega^2) \quad (2.10)$$

com  $\varepsilon_t, \eta_t, \xi_t, \omega_t$  não-correlacionados e  $t = 1, 2, \dots, T$ .

Neste modelo o  $s$  é o período sazonal da série. O período pode ser diário ( $s = 7$ ), mensal ( $s = 12$ ) ou trimestral ( $s = 4$ ).

Vale lembrar que as variâncias dos ruídos dos modelos estruturais e outros parâmetros fixos existentes nesses modelos são chamados de hiperparâmetros, cuja estimação é feita pelo método de máxima verossimilhança.

A seguir é possível compreender como os modelos estruturais podem ser escritos na forma de espaço de estados e como o filtro de Kalman faz a estimação dos hiperparâmetros do modelo e as previsões.

## 2.2. Forma de Espaço de Estados (EE)

Para realizar a estimação dos modelos estruturais é preciso encontrar as estimativas dos hiperparâmetros. A forma de espaço de estados invariante no tempo reescreve as expressões de um modelo estrutural com apenas duas equações: a equação das observações (ou de medida) e a equação do estado (ou transição), dadas respectivamente por:

$$y_t = Z\alpha_t + \varepsilon_t \quad \varepsilon_t \sim N(0, H), \quad (2.11)$$

$$\alpha_t = T\alpha_{t-1} + R\eta_t \quad \eta_t \sim N(0, Q), \quad t = 1, \dots, T \quad (2.12)$$

sendo,

$y_t$  : Série temporal, vetor  $T \times 1$ ;

$\alpha_t$  : Vetor de estados  $k \times 1$ ;

$\varepsilon_t$  : Ruídos não correlacionados com média 0 e variância  $H$ , vetor  $T \times 1$ ;

$\eta_t$  : Ruídos serialmente não correlacionados de média 0 e matriz de covariância  $Q$ , vetor  $g \times 1$ ;

$Z$ : Matriz  $k \times 1$ ;

$T$ : Matriz  $k \times k$ ;

$R$ : Matriz  $k \times g$ ,  $g < k$ ;

$H$ : Matriz  $1 \times 1$ ;

$Q$ : Matriz  $g \times g$ .

Sabe-se também que  $\varepsilon_t$  e  $\eta_t$  são independentes entre si e  $Z, T$  e  $R$  são chamados matrizes do sistema e são conhecidas após a definição da modelagem a ser aplicada à série. Para o MNL as dimensões são:  $Z_{1 \times 1}$ ,  $T_{1 \times 1}$ ,  $R_{1 \times 1}$ ,  $H_{1 \times 1}$  e  $Q_{1 \times 1}$  já no MTL temos:  $Z_{1 \times 2}$ ,  $T_{2 \times 2}$ ,  $R_{2 \times 2}$ ,  $H_{1 \times 1}$  e  $Q_{2 \times 2}$ , e finalmente, para o MEB as dimensões são:  $Z_{1 \times (s+1)}$ ,  $T_{(s+1) \times (s+1)}$ ,  $R_{(s+1) \times (s+1)}$ ,  $H_{1 \times 1}$  e  $Q_{(s+1) \times (s+1)}$ . Adicionalmente, ressalta-se que  $\alpha_0 \sim N(a_0, P_0)$ ,  $E(\varepsilon'_t \eta_z) = 0$  e  $E(\varepsilon'_t \alpha_0) = E(\eta'_t \alpha_0) = 0$ ,  $\forall z, t = 1, 2, \dots, T$ .

Uma das grandes vantagens desses modelos decorre de sua natureza Markoviana e Gaussiana, que permite que estimativas do vetor de estados possam ser obtidas recursivamente. Adicionalmente, várias estruturas para séries temporais podem ser obtidas por meio das equações apresentadas anteriormente. Por exemplo, pode-se adicionar variáveis explicativas, variáveis indicadoras para incluir informação sobre quebras estruturais, estruturas não-lineares, dentre outras.

Cabe destacar que neste trabalho assumir-se-á  $d_t = 0$  e  $c_t = 0$ , indicando que serão modeladas séries univariadas sem a presença de outras covariáveis influentes nesta modelagem.

Aplicando-se a forma de espaço de estados para o MNL, as matrizes do modelo ficam com as seguintes quantidades:

$$Z = 1, \quad T = 1, \quad R = 1, \quad \alpha_t = \mu_t, \quad H = \sigma_\varepsilon^2, \quad Q = \sigma_\eta^2,$$

Já as matrizes do modelo MTL assumem os seguintes valores:

$$Z = [1 \ 0], \quad T = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \alpha_t = \begin{bmatrix} \mu_t \\ \beta_t \end{bmatrix}, \quad H = \sigma_\varepsilon^2, \quad Q = \begin{bmatrix} \sigma_\eta^2 & 0 \\ 0 & \sigma_\xi^2 \end{bmatrix},$$

Por sua vez, o MEB possui as seguintes matrizes:

$$Z = [1 \ 0 \ 1 \ 0 \ \dots \ 0]_{1 \times (s+1)} \quad T = \begin{bmatrix} 1 & 1 & & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & -1 & -1 & \dots & -1 & -1 \\ 0 & 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & 1 & \dots & 0 & 0 \\ 0 & 0 & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 & 0 \end{bmatrix}_{(s+1) \times (s+1)}$$

$$R = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_{(s+1) \times (s+1)} \quad \alpha_t = \begin{bmatrix} \mu_t \\ \beta_t \\ \gamma_t \\ \gamma_{t-1} \\ \vdots \\ \gamma_{t-s+2} \end{bmatrix} \quad H = \sigma_\varepsilon^2 \quad Q = \begin{bmatrix} \sigma_\eta^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_\xi^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_\omega^2 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}_{(s+1) \times (s+1)}$$

### 2.3. Filtro de Kalman

Para fazer previsões em um modelo na forma de espaço de estados a partir de observações passadas, ou seja, até o tempo  $t$ , é necessário que o vetor de estado  $\alpha_t$  seja estimado. Os estimadores do vetor de estado são obtidos por meio do filtro de Kalman (1960).

O filtro de Kalman são equações recursivas que atualizam sequencialmente os estimadores do vetor de estado. Suas etapas básicas são previsão, atualização e suavização.

Sabendo-se que  $Y_{t-1}$  é o vetor das observações até o instante  $t-1$ ,  $E(\alpha_0) = a_0$  e  $Cov(\alpha_0, \alpha'_0) = \mathbf{P}_0$ , tem-se que:

$$(i) (\alpha_t | Y_{t-1}) \sim N(a_{t|t-1}, \mathbf{P}_{t|t-1}), \text{ sendo } a_{t|t-1} = E(\alpha_t | Y_{t-1}) = \mathbf{T}a_{t-1} + c_t, \quad (2.13)$$

$$(ii) \mathbf{P}_{t|t-1} = Var(\alpha_t | Y_{t-1}) = \mathbf{T}\mathbf{P}_{t-1}\mathbf{T}' + \mathbf{R}\mathbf{Q}\mathbf{R}', \quad (2.14)$$

$$(iii) (y_t | Y_{t-1}) \sim N(\tilde{y}_{t|t-1}, F_t), \text{ sendo } \tilde{y}_{t|t-1} = E(y_t | Y_{t-1}) = Z_t a_{t|t-1} + d_t, \quad (2.15)$$

$$(iv) F_t = Var(y_t | Y_{t-1}) = Z_t' \mathbf{P}_{t|t-1} Z_t + H. \quad (2.16)$$

Então, pode-se notar que a média e a matriz de covariância do vetor de estado no tempo  $t$  são respectivamente iguais a  $a_{t+1|t}$  e  $\mathbf{P}_{t+1|t}$ .

Com a recursividade, pode-se obter os seguintes resultados importantes, que são as equações de previsão:

$$a_t = a_{t|t-1} + \mathbf{P}_{t|t-1} \mathbf{Z} \mathbf{F}^{-1} (y_t - \tilde{y}_{t|t-1}), \quad (2.17)$$

$$\mathbf{P}_t = \mathbf{P}_{t|t-1} - \mathbf{P}_{t|t-1} \mathbf{Z} \mathbf{F}^{-1} \mathbf{Z}' \mathbf{P}_{t|t-1}. \quad (2.18)$$

As equações apresentadas anteriormente são chamadas de equações de atualização, pois com elas é possível atualizar os estimadores  $a_{t|t-1}$  e  $\mathbf{P}_{t|t-1}$  do vetor de estado no instante  $t$ . Para facilitar o manuseio das fórmulas, chama-se de  $v_t$  o erro de previsão ou inovações, sendo que  $E(v_t) = 0$  e  $Var(v_t) = F_t$ , e chama-se de  $\mathbf{K}_t$  a matriz de ganho. Assim, tem-se que:

$$v_t = y_t - \tilde{y}_{t|t-1} \Rightarrow v_t = y_t - \mathbf{Z}' a_{t|t-1} - d_t, \quad (2.19)$$

$$\mathbf{K}_t = \mathbf{T} \mathbf{P}_{t|t-1} \mathbf{Z} \mathbf{F}^{-1}. \quad (2.20)$$

Então, substituindo  $v_t$  e o  $\mathbf{K}_t$  nas equações de previsão é possível obter as seguintes equações de atualização:

$$a_{t|t} = a_{t|t-1} + \mathbf{T}^{-1} \mathbf{K}_t v_t, \quad (2.21)$$

$$\mathbf{P}_{t|t} = \mathbf{P}_{t|t-1} - \mathbf{P}_{t|t-1} \mathbf{T}^{-1} \mathbf{K}_t \mathbf{Z}' \mathbf{P}_{t|t-1}. \quad (2.22)$$

Para a inicialização do filtro de Kalman toma-se  $a_0 = 0$  ou  $a_0 = E(Y_t)$  e  $\mathbf{P}_0$  suficientemente grande (Harvey, 1989), nesta dissertação foi utilizado  $\mathbf{P}_0 = 10.000.000$ .

A forma da matriz  $P_t$  para cada um dos modelos MNL, MTL e MEB está detalhada no Apêndice 1, sendo apresentado a seguir apenas as fórmulas gerais.

MNL:  $P_t = \text{cov}(\alpha_t) = P_t^{\mu_t}$

MTL:  $P_t = \text{Cov}(\alpha_t) = \begin{bmatrix} P_t^{\mu_t} & P_t^{\mu_t, \beta_t} \\ P_t^{\mu_t, \beta_t} & P_t^{\beta_t} \end{bmatrix}$ , sendo  $P_t^{\mu_t, \beta_t} = \text{Cov}(\mu_t, \beta_t)$ .

$$\text{MEB: } P_t = \text{Cov}(\alpha_t) = \begin{bmatrix} P_t^{\mu_t} & P_t^{\mu_t, \beta_t} & 0 & 0 \\ P_t^{\mu_t, \beta_t} & P_t^{\beta_t} & 0 & 0 \\ 0 & 0 & P_t^{\gamma_t} & P_t^{\gamma_t, \gamma_{t-1}} \\ 0 & 0 & P_t^{\gamma_t, \gamma_{t-1}} & P_t^{\gamma_{t-1}} \end{bmatrix}, \text{ sendo } P_t^{\gamma_t, \gamma_{t-1}} = \text{Cov}(\gamma_t, \gamma_{t-1}).$$

## 2.4. Estimação dos hiperparâmetros

A função de verossimilhança pode ser calculada a partir das quantidades obtidas pelo Filtro de Kalman. Dessa forma o vetor de hiperparâmetros ( $\psi$ ) estimados é obtido maximizando a função de log-verossimilhança, a qual é dada a seguir:

$$\log L(\psi) = -\frac{T}{2} \log(2\pi) - \frac{1}{2} \sum_{t=1}^T \log |F_t| - \frac{1}{2} \sum_{t=1}^T \nu_t' F_t^{-1} \nu_t \quad (2.23)$$

Pelo fato de a função de verossimilhança ser uma função não linear do vetor de hiperparâmetros, esta estimação não será realizada analiticamente, mas via método numérico, o qual se encontra implementado em ambiente Ox (Doornik 1999). Adicionalmente, vale destacar que o método de otimização utilizado neste trabalho é o BFGS (em referência aos seus autores, Broyden-Fletcher-Goldfarb-Shanno); para maiores detalhes ver Press *et al.* (1988).

### 3. *Bootstrap*

O *bootstrap* é um método de reamostragem de dados introduzido por Efron (1979), usado especialmente na inferência estatística em situações de amostras pequenas, como por exemplo, quando há interesse na construção de intervalos de confiança, testes de hipóteses, estimação de vícios, seleção de modelos e outros.

O *bootstrap* é um método em que são realizadas reamostragens (com reposição) dos dados de uma amostra de tamanho  $n$  finito, tentando aproximar a distribuição de uma determinada função das observações pela distribuição empírica dos dados.

Na prática, o tamanho mais utilizado para cada amostra *bootstrap* é o tamanho  $n$  da amostra original. Se o interesse está voltado para uma estatística  $T$  conjunta, dada por  $T(X_1, X_2, \dots, X_n)$ , torna-se necessário gerar repetidas amostras *bootstrap* de tamanho  $n$ , denotadas por  $X^{*1}, X^{*2}, \dots, X^{*B}$ , onde  $B$  é o número de reamostragens *bootstrap*.

A estimativa numa amostra *bootstrap* se aproxima do valor real quando  $B \rightarrow \infty$ , ou seja, quando as reamostragens são realizadas várias vezes, sendo que o valor de  $B$  é escolhido de acordo com a finalidade para a qual o método *bootstrap* está sendo usado.

Existem basicamente duas maneiras de se realizar o *bootstrap*: paramétrica e não-paramétrica. Na primeira, reamostram-se observações da distribuição geradora dos dados e na última, a reamostragem é feita na própria amostra obtida.

#### 3.1. *Bootstrap* em Modelos Estruturais

Lembrando que o objetivo deste estudo é a estimação do vetor de hiperparâmetros  $\Psi$  e a previsão de valores futuros utilizando modelos estruturais, as duas técnicas *bootstrap* serão utilizadas para auxiliar na estimação dos valores das variâncias desconhecidas dos ruídos dos modelos e no cálculo dos valores previstos.

Sabe-se que para fazer inferências usando o *bootstrap* é necessário que a suposição de independência seja válida. Levando em consideração que as observações em séries temporais são correlacionadas, uma das possibilidades é aplicar o *bootstrap* nas inovações padronizadas do modelo ajustado. Desta forma, o *bootstrap* será utilizado para reamostrar os dados originais, repetidas vezes e, a partir de tais reamostras, estimar por máxima verossimilhança

os hiperparâmetros, conseguindo assim uma série de estimativas *bootstrap* para os mesmos, lembrando que para o MNL,  $\psi = (\sigma_\eta^2, \sigma_\varepsilon^2)$ , para o MTL,  $\psi = (\sigma_\eta^2, \sigma_\xi^2, \sigma_\varepsilon^2)$  e para o MEB,  $\psi = (\sigma_\eta^2, \sigma_\xi^2, \sigma_\omega^2, \sigma_\varepsilon^2)$ .

### 3.2. *Bootstrap* Paramétrico

No *bootstrap* paramétrico são feitas reamostragens (com reposição) de uma distribuição conhecida da qual os dados são obtidos. Neste caso, existem  $X_1, X_2, \dots, X_n$  variáveis aleatórias independentes com função de distribuição comum conhecida  $F$  e tem-se  $\theta$  como um vetor de parâmetros desconhecidos. Então as amostragens serão realizadas diretamente em  $F_\theta$ , conseguindo assim uma amostra *bootstrap* que, usando a notação de Efron & Tibshirani (1993), é denotada por  $X^* = X_1^*, X_2^*, \dots, X_n^*$ .

O procedimento do *bootstrap paramétrico* aplicado a modelos estruturais consiste primeiramente em estimar os hiperparâmetros do modelo pelo método de máxima verossimilhança, baseados na série original. Como os ruídos das equações de observações e de estado (3.11 e 3.12) seguem a distribuição normal com média zero e variância  $H$  e  $Q$ , respectivamente, o *bootstrap* paramétrico é facilmente executado substituindo-se estes hiperparâmetros pelos seus valores estimados e amostrando-se valores desta distribuição. Assim, a série *bootstrap* ( $y_t^*$ ) é calculada recursivamente para  $t = 1, \dots, T$ ,  $B$  vezes, utilizando-se os ruídos estimados e valores iniciais pré-determinados para os componentes do modelo.

$$y_t^* = Za_t^* + \hat{\varepsilon}_t^* \quad (3.1)$$

$$a_t^* = Ta_{t-1}^* + R\hat{\eta}_t^* \quad (3.2)$$

Sendo  $\hat{\varepsilon}_t^* \sim N(0, \hat{H})$  e  $\hat{\eta}_t^* \sim N(0, \hat{Q})$ .

### 3.3. *Bootstrap* Não-paramétrico

Como o próprio nome diz, no *bootstrap* não-paramétrico não se assume uma distribuição para os dados. As amostragens são feitas a partir da amostra original de tamanho  $n$ , sendo necessária a suposição de que cada dado tenha identicamente uma massa de probabilidade igual a  $1/n$ , designada também, por uma suposta distribuição empírica  $\hat{F}$ . Neste caso, a

amostra *bootstrap* também será feita com reposição, retirando-se um conjunto de  $n$  observações da amostra original e que, seguindo a notação de Efron & Tibshirani (1993), também será denotada por  $X^* = X_1^*, X_2^*, \dots, X_n^*$ .

Como o *bootstrap* não-paramétrico não depende da distribuição que os dados seguem (distribuição desconhecida), o mesmo pode ser usado para qualquer conjunto de dados, tendo então maior aplicabilidade que o *bootstrap* paramétrico.

Para a realização do ***bootstrap não-paramétrico*** foi utilizada a proposta do Stoffer & Wall (1991). Tal proposta consiste primeiramente em escrever os modelos na forma de EE, em seguida estimam-se os hiperparâmetros do modelo por máxima verossimilhança e calculam os valores das inovações ( $v_t$ ) e sua variância ( $F_t$ ) por meio do filtro de Kalman. Com isso, as inovações são padronizadas ( $e_t$ ), subtraindo-se a sua média e dividindo pela raiz da variância estimada.

$$e_t = \frac{\hat{v}_t - \bar{v}}{\sqrt{F_t}}, \text{ sendo } \bar{v} = \frac{\sum_{j=1}^n \hat{v}_j}{n} \quad (3.3)$$

Logo, esses resíduos padronizados são reamostrados, gerando a série de resíduos padronizados *bootstrap* ( $e_t^*$ ). Assim, a série *bootstrap* ( $y_t^*$ ) é construída recursivamente com as equações do filtro de Kalman escritas em função dos resíduos *bootstrap*. Ressalta-se que neste estudo a padronização utilizada nos resíduos foi apenas o erro dividido pela raiz de  $F_t$ , a fim de aproximar da metodologia adotada no artigo Rodriguez e Ruiz (2009).

A seguir são apresentadas as equações do filtro de Kalman em função das inovações.

$$a_{t+1|t} = T_t a_{t|t-1} + K_t v_t \quad (3.4)$$

$$v_t = y_t - \tilde{y}_{t|t-1} \Rightarrow y_t = v_t + z_t' a_{t|t-1} \quad (3.5)$$

Definindo o vetor  $S_t = \begin{bmatrix} a_{t+1|t} \\ y_t \end{bmatrix}$ , tem-se:

$$S_t = A_t S_{t-1} + B_t e_t, \quad t = 1, 2, \dots, T \quad (3.6)$$

$$\text{sendo } A_t = \begin{bmatrix} T & 0 \\ Z & 0 \end{bmatrix} \text{ e } B_t = \begin{bmatrix} K\sqrt{F_t} \\ \sqrt{F_t} \end{bmatrix}. \quad (3.7)$$



De um modo geral, a nova série  $y_t^*$  poderá ser obtida resolvendo-se a equação (3.7), substituindo  $e_t$  por  $e_t^*$  e utilizando os valores estimados de  $F_t$  e  $K_t$ .

Para o MNL, por exemplo, a série  $y_t^*$  é dada por:

$$y_t^* = \hat{a}_{t|t-1} + \sqrt{\hat{F}_t} e_t^* \quad (3.8)$$

$$\hat{a}_{t+1|t} = \hat{a}_{t|t-1} + \hat{K}_t \sqrt{\hat{F}_t} e_t^* \quad (3.9)$$

Obtida, então, a série *bootstrap*  $y_t^*$ , o filtro de Kalman será utilizado novamente com essa nova série e através do método de máxima verossimilhança será possível obter as estimativas *bootstrap* dos hiperparâmetros. Este procedimento é chamado de forma de inovação (IF), por Rodriguez e Ruiz (2009).

## 4. Previsão

Um dos objetivos principais de se modelar séries temporais é prever valores futuros. Portanto, o desenvolvimento das funções de previsão torna-se um importante tópico na análise de séries temporais.

Intervalos de previsão de valores futuros baseados em toda informação disponível na amostra podem ser obtidos por diferentes técnicas. Neste estudo foram utilizados três diferentes procedimentos para obter intervalos de confiança para previsões, sendo eles o intervalo percentílico baseado no *bootstrap* paramétrico, o intervalo percentílico baseado no procedimento adotado no artigo Rodriguez e Ruiz (2009), chamado de *bootstrap* não-paramétrico e o procedimento de previsão assintótico. Os detalhes de cada método são descritos a seguir.

### 4.1. Previsão pontual

Os valores futuros das séries são gerados pelo modelo, substituindo o vetor de hiperparâmetros pelos seus estimadores de máxima verossimilhança (EMV). Ressalta-se que neste trabalho as equações para o MEB foram calculadas apenas para  $s=3$ , devido à complexidade de implementar este modelo.

A forma das previsões para os modelos estudados neste trabalho são apresentadas nas próximas subseções. Para o cálculo dos intervalos de previsão *bootstrap* é necessário obter as previsões *bootstrap*. Estas previsões são obtidas com a série original e os EMV's dos hiperparâmetros das séries *bootstrap*, conforme sugerido nos trabalhos de Pascual, Romo e Ruiz (2001), Rodriguez e Ruiz (2009) e Wal e Stoffer (2002).

#### 4.1.1. Previsões pontuais pelo método *bootstrap* paramétrico

Para obter os valores previstos pelo método *bootstrap* paramétrico, basta seguir os seguintes passos:

1º) Estimar os hiperparâmetros do modelo pela função de máxima verossimilhança com base na série real, obtendo  $\hat{\psi}$ .

2º) Amostrar as séries dos ruídos  $\{\varepsilon_t \sim N(0, \hat{\sigma}_\varepsilon^2)\}$  com tamanho  $T+k$ , a partir da distribuição Normal com média zero e variância substituída pelos hiperparâmetros estimados no primeiro passo.

3º) Construir a série *bootstrap*  $\{\hat{y}_t^* ; 1 \leq t \leq T\}$  pelo método *bootstrap* paramétrico (Capítulo 3.2).

4º) Estimar os hiperparâmetros *bootstrap* correspondentes pela função de máxima verossimilhança,  $\hat{\psi}^*$ , a partir da série *bootstrap*  $\hat{y}_t^*$ . Esse passo introduz a incerteza dos parâmetros da previsão.

5º) Obter o vetor de estados  $\hat{a}_{T|T-1}^*$  a partir do  $\hat{\psi}^*$  e as observações originais, obtendo a réplica *bootstrap* para o vetor de estados no tempo  $n$ , a qual incorpora a incerteza causada pela estimação.

6º) Obter as previsões condicionais *bootstrap*  $k$  passos a frente  $\{\hat{y}_{T+k|T}^* ; 1 \leq k \leq K\}$  pelas expressões:

$$\hat{a}_{T+k|T}^* = \hat{T}^* \hat{a}_{T|T-1}^* + R \hat{\eta}_t^* \quad (4.1)$$

$$\hat{y}_{T+k|T}^* = Z \hat{a}_{T|T-1}^* + \hat{\varepsilon}_{T+k}^* \quad (4.2)$$

Os passos 2 até 6 são repetido  $B$  vezes, obtendo-se  $B$  réplicas *bootstrap*.

As Expressões 4.1 e 4.2 foram calculadas para os diferentes EE considerados neste trabalho. A seguir, destaca-se a fórmula geral de previsão para cada modelo estudado, sendo que a derivação das mesmas é apresentada no Apêndice 2.

#### MNL

$$\hat{y}_{T+k}^* = \hat{\mu}_T^* + \sum_{j=1}^k \hat{\eta}_{T+j}^* + \hat{\varepsilon}_{T+k}^* \quad (4.3)$$

#### MTL

$$\hat{y}_{T+k}^* = \hat{\mu}_T^* + k \hat{\beta}_T^* + \sum_{j=1}^k [\hat{\eta}_{T+j}^* + (k-j) \hat{\xi}_{T+j}^*] + \hat{\varepsilon}_{T+k}^* \quad (4.4)$$

### MEB (s=3)

$$\hat{y}_{T+k}^* = \hat{\mu}_T^* + k\hat{\beta}_T^* + \sum_{j=1}^k [\hat{\eta}_{T+j}^* + (k-j)\hat{\xi}_{T+j}^*] - \hat{\gamma}_{T+k-1}^* - \hat{\gamma}_{T+k-2}^* + \hat{\omega}_{T+k}^* + \hat{\varepsilon}_{T+k}^* \quad (4.5)$$

#### 4.1.2. Previsões pontuais no método bootstrap não-paramétrico

No artigo base deste trabalho é proposto o procedimento *bootstrap* para obter os intervalos de predição das observações futuras dos modelos estruturais na forma de EE. Este procedimento também é baseado na forma de inovação (IF).

A seguir são apresentados os passos para a implementação desta metodologia, chamada de SSB por Rodriguez e Ruiz (2009):

1º) Estimar os parâmetros do modelo pelo método QML (*quasi-maximum likelihood*),  $\hat{\psi}$ , e obter as inovações padronizadas  $\{\hat{v}_t^s; 1 \leq t \leq T\}$ .

2º) Obter a sequência de inovações padronizadas  $\{\hat{v}_t^{s*}; 1 \leq t \leq T+k\}$  pela amostragem aleatória com reposição das inovações padronizadas ( $\hat{v}_t^s$ ).

3º) Computar a réplica *bootstrap* não-paramétrica  $\{\hat{y}_t^*; 1 \leq t \leq T\}$  por meio da forma de inovação (IF) usando  $\hat{v}_t^{s*}$  e os hiperparâmetros  $\hat{\psi}$  estimados. A seguir, estimam-se os hiperparâmetros *bootstrap* correspondentes,  $\hat{\psi}^*$ .

4º) Rodar o filtro de Kalman com o  $\hat{\psi}^*$  e as observações originais, obtendo a réplica *bootstrap* para o vetor de estados no tempo  $T$ , a qual incorpora a incerteza causada pela estimação do vetor de estados  $\hat{a}_{T|T-1}^*$ .

5º) Obter as previsões condicionais *bootstrap*  $k$  passos a frente  $\{\hat{y}_{T+k|T}^*; 1 \leq k \leq K\}$  pelas expressões:

$$\hat{a}_{T+k|T}^* = \hat{T}^{*k} \hat{a}_{T|T-1}^* + \sum_{j=0}^{k-1} \hat{T}^{*k-1-j} \hat{K}_{T+j}^* \hat{F}_{T+j}^{*-1} \hat{v}_{T+j}^* \quad (4.6)$$

$$\hat{y}_{T+k|T}^* = \hat{Z}^* \hat{T}^{*k} \hat{a}_{T|T-1}^* + \hat{Z}^* \sum_{j=0}^{k-1} \hat{T}^{*k-1-j} \hat{K}_{T+j}^* \hat{F}_{T+j}^{*-1} \hat{v}_{T+j}^* + \hat{v}_{T+k}^* \quad (4.7)$$

Os passos 2 até 5 são repetido  $B$  vezes, obtendo-se  $B$  réplicas *bootstrap*.

As expressões 4.6 e 4.7 também foram calculadas para os diferentes EE considerados neste trabalho. A seguir, destaca-se a fórmula geral de previsão para cada modelo estudado, sendo que a derivação das mesmas é apresentada no Apêndice 3.

#### MNL

$$\hat{\mu}_{T+k|T}^* = \hat{\mu}_{T|T-1}^* + \sum_{j=0}^{k-1} \hat{K}_{T+j}^* \hat{v}_{T+j}^* \quad (4.8)$$

$$\hat{y}_{T+k|T}^* = \hat{\mu}_{T+k|T}^* + \hat{v}_{T+k}^* \quad (4.9)$$

#### MTL

$$\hat{\mu}_{T+k|T}^* = \hat{\mu}_{T|T-1}^* + k \hat{\beta}_{T|T-1}^* + \sum_{j=0}^{k-1} \left( \hat{K}_{T+j}^{*\mu} + (k-1-j) \hat{K}_{T+j}^{*\beta} \right) \hat{v}_{T+j}^* \quad (4.10)$$

$$\hat{\beta}_{T+k|T}^* = \hat{\beta}_{T|T-1}^* + \sum_{j=0}^{k-1} \hat{K}_{T+j}^{*\beta} \hat{v}_{T+j}^* \quad (4.11)$$

$$\hat{y}_{T+k|T}^* = \hat{\mu}_{T+k|T}^* + \hat{v}_{T+k}^* \quad (4.12)$$

#### MEB (s=3)

$$\hat{\mu}_{T+k|T}^* = \hat{\mu}_{T|T-1}^* + k \hat{\beta}_{T|T-1}^* + \sum_{j=0}^{k-1} \left( \hat{K}_{T+j}^{*\mu} + (k-1-j) \hat{K}_{T+j}^{*\beta} \right) \hat{v}_{T+j}^* \quad (4.13)$$

$$\hat{\beta}_{T+k|T}^* = \hat{\beta}_{T|T-1}^* + \sum_{j=0}^{k-1} \hat{K}_{T+j}^{*\beta} \hat{v}_{T+j}^* \quad (4.14)$$

$$\hat{y}_{T+k|T}^* = -\hat{y}_{T+k-1|T}^* - \hat{y}_{T+k-2|T}^* + K_{T+k-1}^{\gamma} \hat{v}_{T+k-1}^* \quad (4.15)$$

$$\hat{y}_{T+k|T}^* = \hat{\mu}_{T+k|T}^* + \hat{y}_{T+k|T}^* + \hat{v}_{T+k}^* \quad (4.16)$$

#### **4.1.3. Previsões pontuais para o cálculo do intervalo assintótico**

Para obter a previsão  $k$  passos à frente com a forma de espaço de estados são utilizadas as seguintes equações (Franco *et al.* (2009):

$$\tilde{y}_{T+k|T}(\psi) = Z' \left[ \left( \prod_{i=1}^k T \right) a_T \right] \quad (4.17)$$

$$a_{T+k} = \left( \prod_{i=1}^k T \right) a_T + \sum_{i=1}^k \left( \prod_{j=1}^{k-i} T \right) R' \eta_{T+i} \quad (4.18)$$

A variância do valor previsto é dada por:

$$\text{Var}(\tilde{y}_{T+k} | Y_{T,\psi}) = Z' \left( \prod_{i=1}^k T \right) P_T \left( \prod_{i=1}^k T \right) Z + Z' \sum_{i=1}^k \left( \prod_{j=1}^{k-i} T \right) RQR' \left( \prod_{j=1}^{k-i} T \right) Z + H \quad (4.19)$$

Utilizando o EMV para obter as estimativas dos hiperparâmetros, as Equações (4.17) e (4.19) podem ser escritas como:

$$\tilde{y}_{T+k|T} = \hat{Z} \hat{T}^k \hat{a}_T \quad (4.20)$$

$$F_{T+k|T} = \hat{Z} \hat{T}^k \hat{P}_T (\hat{T}^k)' \hat{Z} + \hat{Z}' \sum_{j=0}^{k-1} \hat{T}^j RQR' (\hat{T}^j)' \hat{Z} + H \quad (4.21)$$

A seguir, destaca-se a fórmula geral de previsão para cada modelo estudado, sendo que a derivação das mesmas é apresentada no Apêndice 4.

### MNL

$$F_{T+k|T} = \hat{P}_T^{\mu t} + k\sigma_\eta^2 + \sigma_\varepsilon^2 \quad (4.22)$$

$$\tilde{y}_{T+k|T} = \hat{\mu}_T \quad (4.23)$$

### MTL

$$F_{T+k|T} = P_T^{\mu t} + k^2 P_T^{\beta t} + 2k P_t^{\mu t, \beta t} + k\sigma_\eta^2 + \sum_{i=1}^k (i-1)^2 \sigma_\xi^2 + \sigma_\varepsilon^2 \quad (4.24)$$

$$\tilde{y}_{T+k|T} = \hat{\mu}_T + k\hat{\beta}_T \quad (4.25)$$

### MEB (s=3)

Assuma que  $x$  seja um valor retornado da seguinte condição: para  $i=1$  até  $k$ , se o resto da divisão entre  $i$  e o grau de sazonalidade for diferente de 0,  $x = x + 1$ .

**Se o resto da divisão entre  $k$  e o grau de sazonalidade for = 0**

$$F_{T+k|T} = P_T^{\mu t} + k^2 P_T^{\beta t} + 2k P_t^{\mu t, \beta t} + P_T^{\gamma t} + k\sigma_\eta^2 + \sum_{i=1}^k [(i-1)^2 \sigma_\xi^2] + x\sigma_\omega^2 + \sigma_\varepsilon^2 \quad (4.26)$$

$$\tilde{y}_{T+k|T} = \hat{\mu}_t + k\hat{\beta}_t + \hat{\gamma}_t \quad (4.27)$$

**Se o resto da divisão entre k e o grau de sazonalidade for = 1**

$$F_{T+k|T} = P_T^{\mu_t} + k^2 P_T^{\beta_t} + 2k P_t^{\mu_t, \beta_t} + P_T^{\gamma_t} + P_T^{\gamma_{t-1}} + 2P_T^{\gamma_t, \gamma_{t-1}} + k\sigma_\eta^2 + \sum_{i=1}^k [(i-1)^2 \sigma_\xi^2] + x\sigma_\omega^2 + \sigma_\varepsilon^2 \quad (4.28)$$

$$\tilde{y}_{T+k|T} = \hat{\mu}_t + k\hat{\beta}_t - \hat{\gamma}_t - \hat{\gamma}_{t-1} \quad (4.29)$$

**Se o resto da divisão entre k e o grau de sazonalidade for = 2**

$$F_{T+k|T} = P_T^{\mu_t} + k^2 P_T^{\beta_t} + 2k P_t^{\mu_t, \beta_t} + P_T^{\gamma_{t-1}} + k\sigma_\eta^2 + \sum_{i=1}^k [(i-1)^2 \sigma_\xi^2] + x\sigma_\omega^2 + \sigma_\varepsilon^2 \quad (4.30)$$

$$\tilde{y}_{T+k|T} = \hat{\mu}_t + k\hat{\beta}_t + \hat{\gamma}_{t-1} \quad (4.31)$$

**4.2. Intervalos de Previsão**

A seguir são apresentados os métodos de construção de intervalos de previsão comparados neste trabalho: o método assintótico (**Assintótico**), o intervalo percentílico baseado no *bootstrap* paramétrico (**Boot Par.**) e o intervalo percentílico baseado no *bootstrap* não-paramétrico (**Boot NP**).

**4.2.1. Intervalos de Previsão Assintóticos**

Utilizando propriedades assintóticas, baseadas na distribuição Normal (Harvey, 1989), tem-se que um intervalo de confiança assintótico de  $100 * (1 - \alpha)\%$  para  $\tilde{y}_{T+k}$  é dado por:

$$\tilde{y}_{T+k|T} \pm z_{\alpha/2} \sqrt{\text{Var}(\tilde{y}_{T+k|T} | Y_{T,\psi})} \quad (4.32)$$

onde  $z_{\alpha/2}$  é o percentil  $\alpha/2$  da distribuição Normal-padrão e a  $\text{Var}(\tilde{y}_{T+k|T} | Y_{T,\psi})$  é a variância do valor previsto, dada por  $F_{T+k|T}$ .

**4.2.2. Intervalos de previsão Bootstrap Percentílico**

Neste trabalho são utilizados os intervalos de previsão *bootstrap* percentílico tanto para o método de previsão *Bootstrap* Paramétrico quanto para o método *Bootstrap* Não-paramétrico.

Para construir os intervalos percentílicos temos o seguinte procedimento:

1º) Gerar  $B$  amostras *bootstrap* da série original, calculando para cada replicação, os valores previstos para o modelo.

2º) Para um passo  $k$  previsto, ordena-se as previsões de todas as réplicas *bootstrap*, para esse passo  $k$ , em ordem crescente de valor.

3º) Após a ordenação das previsões, toma-se o percentil  $100\left(\frac{\alpha}{2}\right)^o$  como o limite inferior do intervalo para o valor real do passo  $k$  e o percentil de ordem  $100\left(1-\frac{\alpha}{2}\right)^o$  como o limite superior do intervalo. Obtendo um intervalo de  $100*(1-\alpha)\%$  de confiança.

No artigo Rodriguez e Ruiz (2009) os autores estudaram as propriedades dos intervalos de predição *bootstrap* percentílico para amostras finitas, com base no procedimento de previsão *Bootstrap* Não-paramétrico, e as compararam com os intervalos assintóticos e com os intervalos obtidos no estudo de Wall e Stoffer (2002) para modelos de nível local, assim como os métodos de previsão utilizados em cada caso. Como resultado, eles mostraram que o procedimento de previsão *Bootstrap* Não-paramétrico apresentou propriedades melhores para amostras finitas quando comparados com o procedimento *bootstrap* utilizado por Wall e Stoffer (2002). Adicionalmente, os autores mostraram que o método utilizado por eles é mais adequado do que o assintótico, principalmente na presença de ruídos não-Gaussianos. Para finalizar, concluíram também que os intervalos *bootstrap* percentílicos adotados por eles e pelos autores Wall e Stoffer (2002) para uma mesma série temporal se apresentaram muito similares, no entanto o tempo computacional despendido por Rodriguez e Ruiz (2009) foi muito menor.



## 5. Resultados da simulação

Neste estudo rotinas computacionais na linguagem Ox (ver algoritmo no Apêndice 6), para os procedimentos descritos nas seções anteriores foram implementadas para o MNL, MTL e MEB (com  $s=3$ ) variando o tamanho da série temporal e os valores reais dos hiperparâmetros. O objetivo é comparar as taxas de cobertura dos intervalos de previsão *bootstrap* paramétrico (Boot Par.), *bootstrap* não-paramétrico (Boot NP) e assintótico (Assintótico), utilizando um nível nominal de 95%.

A princípio, estabeleceram-se os valores dos hiperparâmetros que seriam adotados para simular os EE. Com tais valores pré-definidos, as previsões foram obtidas pelos três métodos descritos no Capítulo 4. Para os procedimentos *bootstrap* fixou-se  $B=1.000$  reamostragens.

O número de replicações Monte Carlo foi fixado em 1.000, ou seja, todo o processo de previsão e construção de intervalos de predição foi feito para 1.000 séries diferentes do mesmo modelo, e, além disso, variou-se o tamanho das séries em  $n=50$ ,  $n=100$  e  $n=500$  e foram obtidas previsões 1, 6 e 12 passos à frente.

Para cada série simulada calculou-se a média das 1.000 estimativas dos hiperparâmetros *bootstrap*, ordenou-se os valores previstos pelo método *bootstrap* para cada passo à frente, sendo os limites do intervalo de previsão *bootstrap* aqueles que ocupam as posições que deixam 2,5% dos valores abaixo dele e 2,5% dos valores acima dele. Após construir o intervalo de previsão *bootstrap*, foi construído o intervalo de previsão assintótico e para ambos verificou-se se o valor real da série simulada (guardado no início do processo) pertencia ao intervalo, ou se estava abaixo do limite inferior ou acima do limite superior. Todas essas informações foram armazenadas para cada série simulada.

Obteve-se como resultado final as médias dos estimadores de máxima verossimilhança (EMV) dos hiperparâmetros, a média das 1.000 médias das estimativas *bootstrap* paramétrica e não-paramétrica dos hiperparâmetros, a média dos limites inferior e superior dos intervalos de previsão *bootstrap* paramétrico, *bootstrap* não-paramétrico e assintótico, para cada passo à frente, assim como a cobertura média desses intervalos.

A seguir são apresentados os resultados das simulações para cada modelo. Ressalta-se que um estudo mais aprofundado sobre estimação pode ser encontrado no Franco, Santos, Ribeiro & Cruz (2008).

## 5.1 Resultados para o MNL

Várias simulações foram feitas com o algoritmo construído para o modelo MNL, variando o tamanho da série, o valor dos hiperparâmetros, a razão sinal ruído  $q = \frac{\sigma_\eta^2}{\sigma_\varepsilon^2}$  e, por fim, o número de previsões desejadas ( $k$ ), a fim de comparar os resultados com os apresentados no artigo de Rodriguez e Ruiz (2009) (Apêndice 5). Os resultados das simulações para as estimativas pontuais dos hiperparâmetros e para as previsões são apresentados, respectivamente, nas Tabelas 1 e 2, considerando uma cobertura nominal de 95%:

**Tabela 1: Estimativas dos hiperparâmetros para as séries simuladas do MNL**

Tamanho da série	Razão sinal-ruído	Hiperparâmetros	Valor real	Valor Estimado		
				Boot Par.	Boot NP	EMV
50	0,1	$\sigma_\eta^2$	0,1	0,109	0,126	0,104
		$\sigma_\varepsilon^2$	1,0	0,994	0,976	1,002
	1	$\sigma_\eta^2$	1,0	1,014	1,117	1,022
		$\sigma_\varepsilon^2$	1,0	0,998	0,949	0,992
	2	$\sigma_\eta^2$	1,0	0,983	1,074	1,012
		$\sigma_\varepsilon^2$	0,5	0,510	0,463	0,494
100	0,1	$\sigma_\eta^2$	0,1	0,106	0,116	0,105
		$\sigma_\varepsilon^2$	1,0	0,996	0,983	0,996
	1	$\sigma_\eta^2$	1,0	1,027	1,086	1,026
		$\sigma_\varepsilon^2$	1,0	0,987	0,957	0,988
	2	$\sigma_\eta^2$	1,0	1,018	1,071	1,023
		$\sigma_\varepsilon^2$	0,5	0,492	0,463	0,489
500	0,1	$\sigma_\eta^2$	0,1	0,100	0,102	0,099
		$\sigma_\varepsilon^2$	1,0	1,003	1,000	1,003
	1	$\sigma_\eta^2$	1,0	0,997	1,008	0,997
		$\sigma_\varepsilon^2$	1,0	1,004	0,998	1,004
	2	$\sigma_\eta^2$	1,0	0,996	1,007	0,996
		$\sigma_\varepsilon^2$	0,5	0,503	0,497	0,503

A partir da Tabela 1, observa-se que os resultados das estimativas pontuais dos hiperparâmetros estão relativamente próximas dos valores reais, sendo que a diferença fica geralmente na segunda casa decimal. Os valores do Boot NP são os mais viciados, destacando-se que, para  $n=500$  os resultados obtidos pelo método Boot Par. e pelo EMV são

praticamente os mesmos, independente da razão sinal-ruído. Como esperado, as estimativas são menos viciadas à medida que o tamanho da série aumenta. A partir dos resultados da Tabela 1 podemos concluir que os valores obtidos nos procedimentos *bootstrap* o capacitam para ser utilizado como uma alternativa na construção de intervalos de confiança para os hiperparâmetros.

Uma nova análise pode ser feita ao comparar os resultados das estimativas de máxima verossimilhança entre as diferentes razões sinal-ruído. Essa comparação mostra que a razão sinal-ruído igual a 0,1 apresentou os melhores resultados, uma vez que as estimativas apresentaram sempre uma precisão de três casas decimais para os dois hiperparâmetros estudados e para qualquer tamanho de amostra.

**Tabela 2: Intervalos de previsão para as séries simuladas do MNL**

Tamanho da série	Razão sinal-ruído	K	Cobertura média 95%			Cobertura média nas caudas						Amplitude média		
			Boot Par.	Boot NP	Assintótico	Boot Par.		Boot NP		Assintótico		Boot Par.	Boot NP	Assintótico
						Abaixo	Acima	Abaixo	Acima	Abaixo	Acima			
50	0,1	1	0,926	0,946	0,941	0,032	0,042	0,025	0,029	0,024	0,035	4,275	4,890	4,513
		6	0,915	0,947	0,925	0,042	0,043	0,026	0,027	0,040	0,035	5,094	5,818	5,287
		12	0,913	0,938	0,919	0,040	0,047	0,029	0,033	0,039	0,042	5,880	6,744	6,049
	1	1	0,921	0,973	0,939	0,036	0,043	0,012	0,015	0,026	0,035	5,671	7,633	6,250
		6	0,928	0,942	0,935	0,031	0,041	0,024	0,034	0,028	0,037	10,362	12,032	10,689
		12	0,921	0,932	0,923	0,038	0,041	0,034	0,034	0,040	0,037	14,013	15,755	14,251
	2	1	0,925	0,979	0,941	0,032	0,043	0,008	0,013	0,025	0,034	4,859	6,713	5,275
		6	0,934	0,946	0,935	0,027	0,039	0,023	0,031	0,028	0,037	9,930	11,292	10,138
		12	0,927	0,925	0,932	0,035	0,038	0,039	0,036	0,035	0,033	13,698	15,038	13,854
100	0,1	1	0,929	0,963	0,957	0,034	0,037	0,018	0,019	0,020	0,023	4,179	4,834	4,559
		6	0,930	0,948	0,942	0,025	0,045	0,018	0,034	0,019	0,039	5,041	5,670	5,352
		12	0,920	0,934	0,930	0,040	0,040	0,030	0,036	0,035	0,035	5,890	6,547	6,153
	1	1	0,923	0,978	0,949	0,029	0,048	0,008	0,014	0,022	0,029	5,596	7,555	6,306
		6	0,925	0,947	0,941	0,035	0,040	0,026	0,027	0,027	0,032	10,395	11,847	10,815
		12	0,934	0,935	0,940	0,033	0,033	0,034	0,031	0,031	0,029	14,140	15,516	14,455
	2	1	0,930	0,980	0,948	0,028	0,042	0,006	0,014	0,023	0,029	4,836	6,717	5,323
		6	0,938	0,948	0,941	0,030	0,032	0,027	0,025	0,027	0,032	10,012	11,284	10,271
		12	0,938	0,941	0,943	0,032	0,030	0,029	0,030	0,029	0,028	13,865	15,037	14,059
500	0,1	1	0,909	0,955	0,953	0,042	0,049	0,022	0,023	0,024	0,023	4,119	4,756	4,586
		6	0,925	0,947	0,942	0,033	0,042	0,023	0,030	0,025	0,033	4,958	5,523	5,351
		12	0,930	0,950	0,946	0,034	0,036	0,026	0,024	0,028	0,026	5,796	6,309	6,142
	1	1	0,907	0,974	0,948	0,040	0,053	0,010	0,016	0,023	0,029	5,545	7,464	6,334
		6	0,935	0,961	0,945	0,032	0,033	0,018	0,021	0,026	0,029	10,345	11,542	10,788
		12	0,936	0,946	0,944	0,034	30,000	0,026	0,028	0,029	0,027	14,058	15,032	14,414
	2	1	0,913	0,982	0,938	0,040	0,047	0,008	0,010	0,027	0,035	4,797	6,641	5,347
		6	0,939	0,961	0,948	0,028	0,033	0,016	0,023	0,024	0,028	9,964	11,024	10,239
		12	0,935	0,944	0,940	0,034	0,031	0,027	0,029	0,032	0,028	13,782	14,642	14,010

A partir da Tabela 2, pode-se dizer que em geral, as coberturas médias dos intervalos ficaram mais próximas de 95% quando o tamanho da amostra era igual a 500, ou seja, em 95% das séries simuladas, os valores reais para os passos de previsão apresentados se

encontram dentro do intervalo. Adicionalmente, verifica-se que com o aumento do tamanho da amostra, diminui-se a amplitude dos intervalos de previsão, fixando o tipo de modelo utilizado e a razão sinal-ruído, independente da metodologia utilizada, e que esses intervalos se tornam mais simétricos.

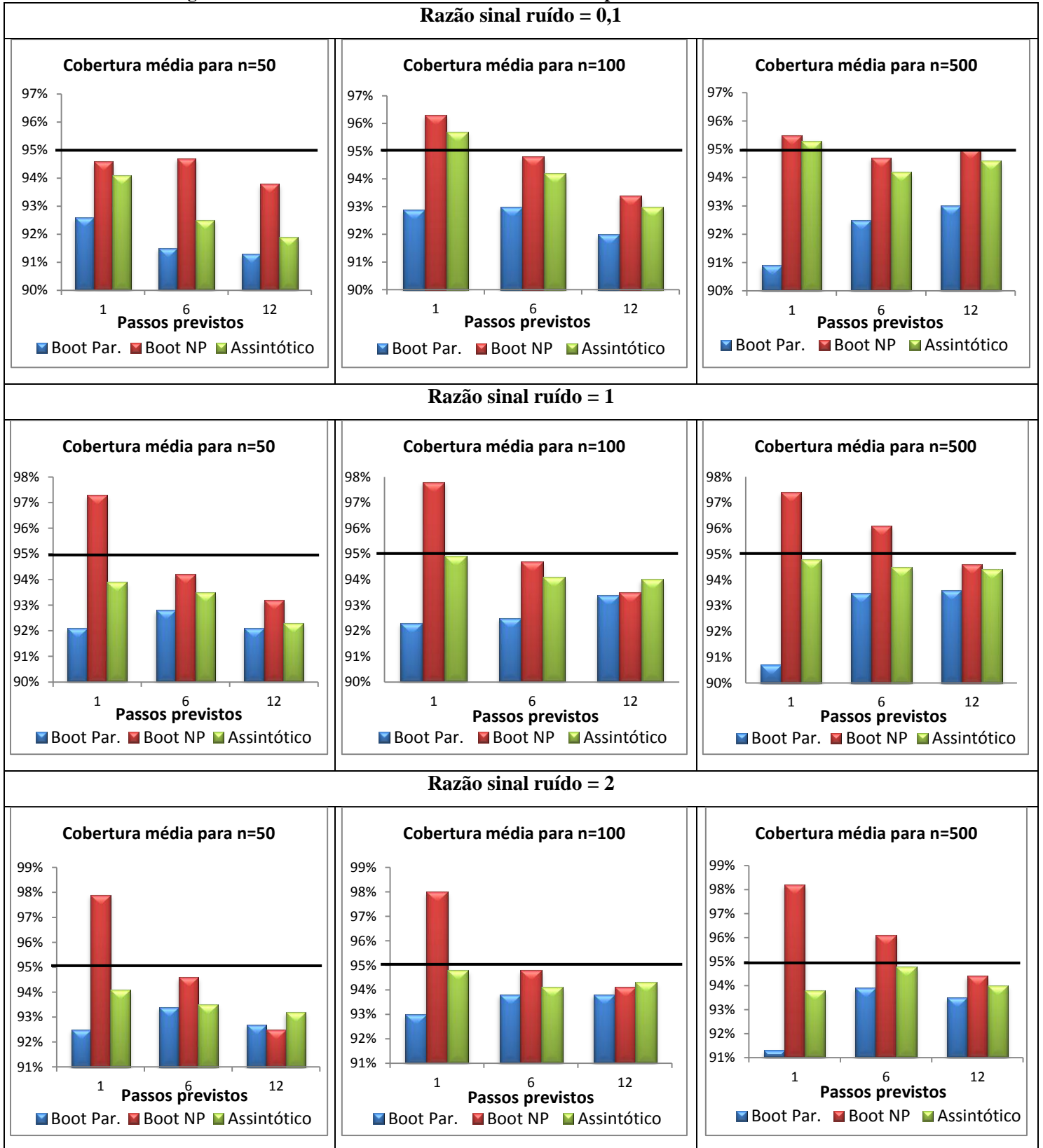
Ao analisar os resultados para a razão sinal-ruído igual a 0,1 pode-se concluir que os intervalos de previsão obtidos pelo método *bootstrap* não-paramétrico apresentaram uma cobertura média mais próxima de 95% para os passos previstos, sendo melhores que os demais métodos de construção de intervalos de previsão. Ressalta-se ainda o fato deles apresentarem a maior amplitude em todos os passos previstos e também serem os mais simétricos.

Especificando a análise para as demais razões sinal-ruído ( $q=1$  e  $q=2$ ), observa-se que os intervalos assintóticos apresentaram resultados melhores que os intervalos *bootstrap*.

Com a Tabela 2 também é possível verificar que os intervalos *bootstrap* paramétrico não se mostraram eficientes em nenhum caso, uma vez que apresentaram as menores coberturas nominais, dadas pelos baixos valores de amplitude. Adicionalmente, nota-se que quando  $q=0,1$  os intervalos assintóticos diminuem muito a sua cobertura nominal para valores altos de  $k$ , principalmente para as séries de tamanho 50 e 100. Em geral, conclui-se que independente da razão sinal-ruído adotada, do tamanho da série e do valor de  $k$ , todos os intervalos mostraram-se simétricos, e também, que sempre as maiores amplitudes foram observadas nos intervalos *bootstrap* não-paramétricos e as menores amplitudes nos intervalos *bootstrap* paramétrico.

Para melhor visualizar os resultados obtidos com a simulação do MNL foram construídos os gráficos da Figura 1, os quais apresentam a cobertura dos intervalos de previsão para  $k=1$ ,  $k=6$  e  $k=12$ , considerando o tamanho da série e a razão sinal ruído( $q$ ).

Figura 1: Cobertura média dos intervalos construídos para as séries simuladas do MNL

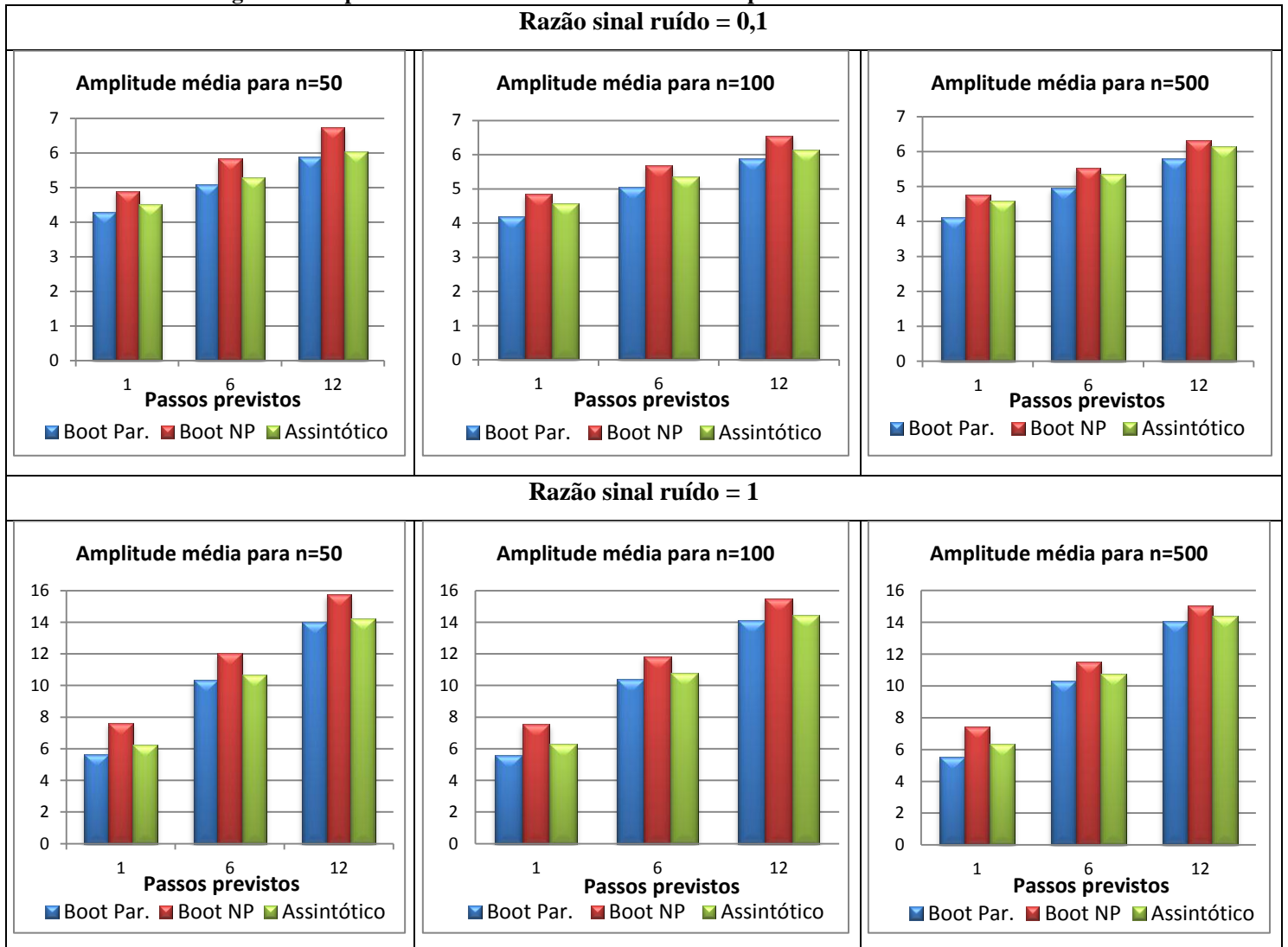


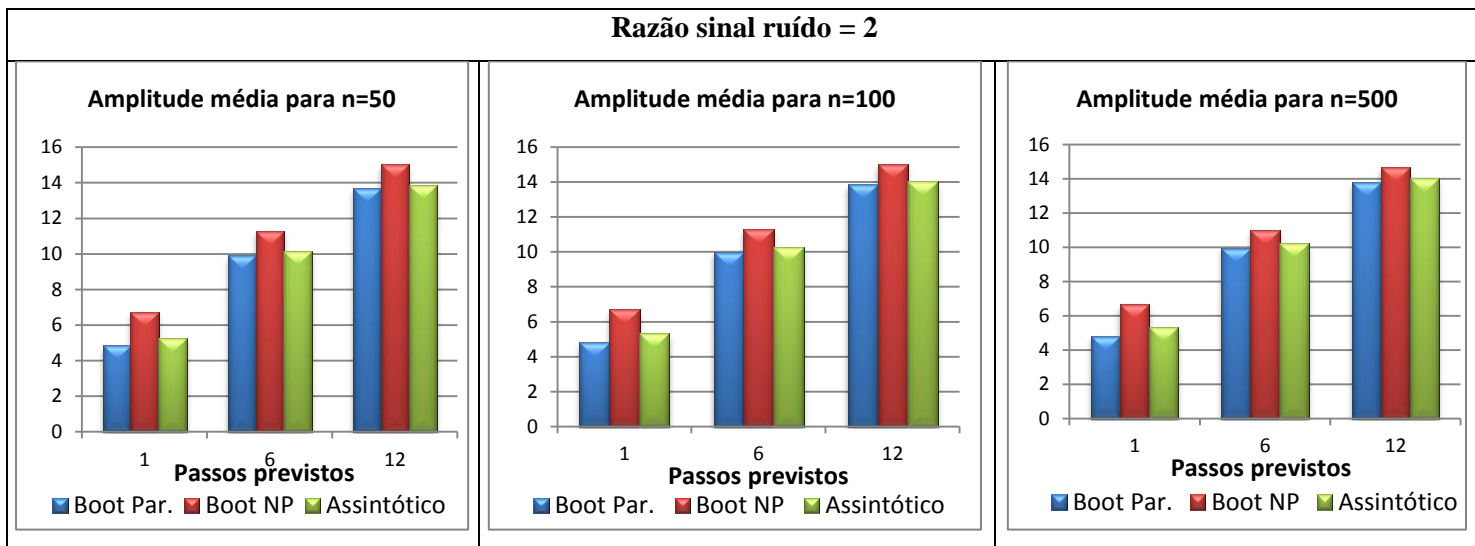
Ao comparar com os resultados apresentados no artigo Rodriguez e Ruiz (2009) (Apêndice 5), observa-se que essas simulações apresentam valores mais distantes de 95%,

sendo o mínimo observado igual a 90,07% e o máximo igual a 98,00%. Enquanto no artigo, as coberturas variaram de 92,8% a 94,8%. Entretanto, os resultados obtidos com os intervalos *bootstrap* não-paramétricos neste estudo, quando  $q=0,1$ , foram melhores que os obtidos no artigo supracitado, uma vez que se aproximaram mais da cobertura nominal de 95%.

Além dos resultados apresentados anteriormente, também foi feito um estudo sobre a amplitude dos intervalos de previsão para cada razão sinal-ruído ( $q$ ), apresentado na Figura 2.

**Figura 2: Amplitude média dos intervalos construídos para as séries simuladas do MNL**





Observa-se com os gráficos apresentados anteriormente que independente da razão sinal-ruído e do procedimento de previsão adotado, as amplitudes aumentam com o aumento do número de previsões e tendem a diminuir com o aumento do tamanho da série.

## 5.2 Resultados para o MTL

Para o MTL também foi feito um estudo com diferentes simulações variando apenas o tamanho da série e o número de previsões desejadas ( $k$ ). Os valores dos hiperparâmetros foram fixados em  $\sigma_{\eta}^2 = 0,50$ ,  $\sigma_{\xi}^2 = 0,01$  e  $\sigma_{\varepsilon}^2 = 1,00$ .

Ressalta-se que o artigo de Rodriguez e Ruiz (2009) não apresentou nenhum resultado para este modelo, sendo esse estudo uma ampliação proposta neste trabalho.

Os resultados das simulações para as estimativas pontuais dos hiperparâmetros e para as previsões são apresentados, respectivamente, nas Tabelas 3 e 4, considerando uma cobertura nominal de 95%:

**Tabela 3: Estimativas dos hiperparâmetros para as séries simuladas do MTL**

Tamanho da série	Hiperparâmetros	Valor real	Valor Estimado		
			Boot Par.	Boot NP	EMV
50	$\sigma_{\eta}^2$	0,50	0,467	0,465	0,488
	$\sigma_{\xi}^2$	0,01	0,011	0,015	0,011
	$\sigma_{\varepsilon}^2$	1,00	1,003	1,025	1,002
100	$\sigma_{\eta}^2$	0,50	0,486	0,465	0,489
	$\sigma_{\xi}^2$	0,01	0,010	0,013	0,010
	$\sigma_{\varepsilon}^2$	1,00	1,001	1,020	1,002
500	$\sigma_{\eta}^2$	0,50	0,501	0,488	0,501
	$\sigma_{\xi}^2$	0,01	0,010	0,011	0,010
	$\sigma_{\varepsilon}^2$	1,00	0,999	1,007	0,999

A partir da Tabela 3, observa-se que as estimativas são muito semelhantes, com uma precisão de duas casas decimais para o  $\sigma^2_{\xi}$  e o  $\sigma^2_{\varepsilon}$  independente do tamanho da série. Destaca-se que para  $n=500$  os resultados obtidos pelo método Boot Par. e pelo EMV são os mesmos. Novamente, podemos concluir que os valores obtidos nos procedimentos *bootstrap* estão bem próximos das estimativas de máxima verossimilhança, portanto este método pode ser utilizado na construção de intervalos de confiança para os hiperparâmetros.

**Tabela 4: Intervalos de previsão para as séries simuladas do MTL**

Tamanho da série	K	Cobertura média 95%			Cobertura média nas caudas						Amplitude média		
		Boot Par.	Boot NP	Assintótico	Boot Par.		Boot NP		Assintótico		Boot Par.	Boot NP	Assintótico
					Abaixo	Acima	Abaixo	Acima	Abaixo	Acima			
50	1	0,896	0,968	0,930	0,053	0,051	0,017	0,015	0,038	0,032	5,158	7,381	5,722
	6	0,863	0,943	0,892	0,075	0,062	0,036	0,021	0,063	0,045	9,424	13,983	10,767
	12	0,816	0,918	0,855	0,105	0,079	0,049	0,033	0,087	0,058	15,489	23,989	17,919
100	1	0,897	0,978	0,948	0,050	0,053	0,013	0,009	0,026	0,026	4,960	7,123	5,811
	6	0,861	0,961	0,920	0,077	0,062	0,018	0,021	0,038	0,042	9,393	13,398	11,150
	12	0,829	0,940	0,895	0,084	0,087	0,037	0,023	0,053	0,052	15,717	22,704	18,790
500	1	0,881	0,975	0,949	0,063	0,056	0,012	0,013	0,028	0,023	4,806	7,007	5,922
	6	0,870	0,981	0,961	0,060	0,070	0,011	0,008	0,018	0,021	8,467	13,155	11,734
	12	0,834	0,972	0,952	0,081	0,085	0,010	0,018	0,015	0,033	13,875	22,042	20,094

A partir da Tabela 4, pode-se dizer que em geral, os intervalos Boot NP e Assintótico apresentaram os melhores desempenhos, sendo que o Boot Par subestimou muito a cobertura nominal fixada, com valores inferiores a 90%. Adicionalmente, verifica-se que com o



aumento do tamanho da amostra, o intervalo assintótico apresenta resultados mais próximos de 95%. Além disto, a amplitude dos intervalos de previsão diminui, independente da metodologia utilizada, e eles se tornam mais simétricos.

Ao analisar os resultados para as séries de tamanho 50 pode-se concluir que os intervalos de previsão obtidos pelo método *bootstrap* não-paramétrico apresentaram uma cobertura média mais próxima de 95% para os passos previstos, sendo melhores que os demais métodos de construção de intervalos de previsão. Ressalta-se ainda o fato deles apresentarem a maior amplitude em todos os passos previstos e também serem os mais simétricos.

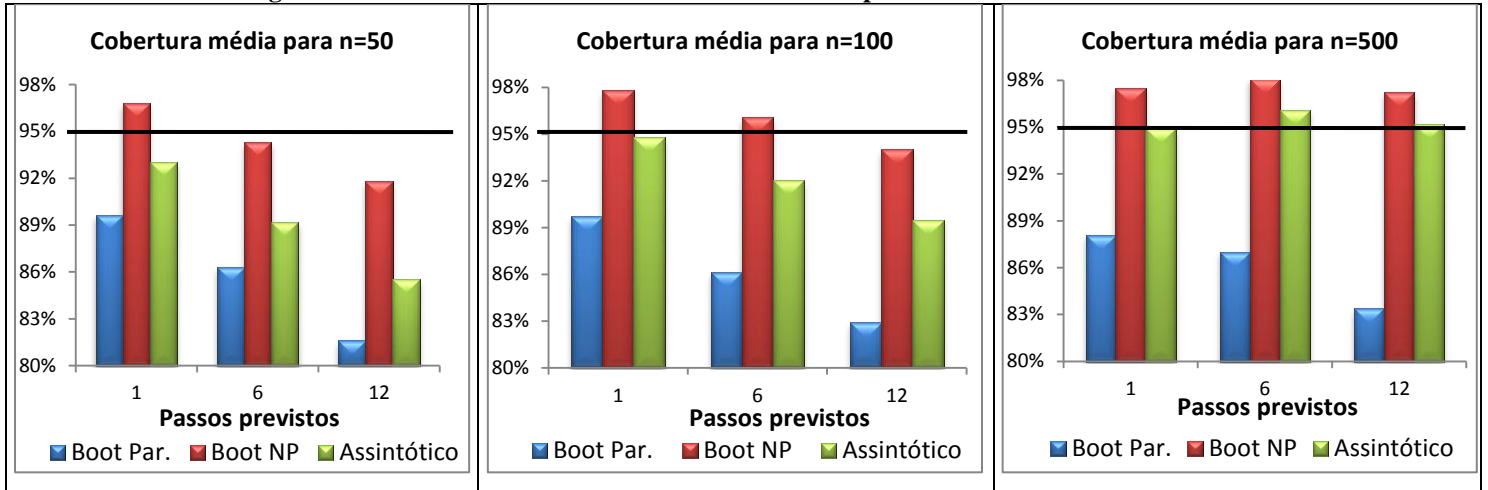
Especificando a análise para os demais tamanhos de série ( $n=100$  e  $n=500$ ), observa-se que os intervalos assintóticos apresentaram resultados melhores que os intervalos *bootstrap*. Adicionalmente, nota-se que os intervalos assintóticos diminuem muito a sua cobertura nominal para valores altos de  $k$ , para as séries de tamanho 50 e 100, no entanto o mesmo não ocorre quando  $n=500$ .

Com a Tabela 4 também é possível verificar que os intervalos *bootstrap* paramétrico não se mostraram eficientes em nenhum caso, assim como foi verificado nos resultados do MNL, uma vez que apresentaram as menores coberturas nominais, dadas pelos baixos valores de amplitude.

Conclui-se ainda, assim como visto nos resultados do MNL, que independente do tamanho da série e do valor de  $k$ , todos os intervalos mostraram-se ser simétricos, e também, que sempre as maiores amplitudes foram observadas nos intervalos *bootstrap* não-paramétricos e as menores amplitudes nos intervalos *bootstrap* paramétrico.

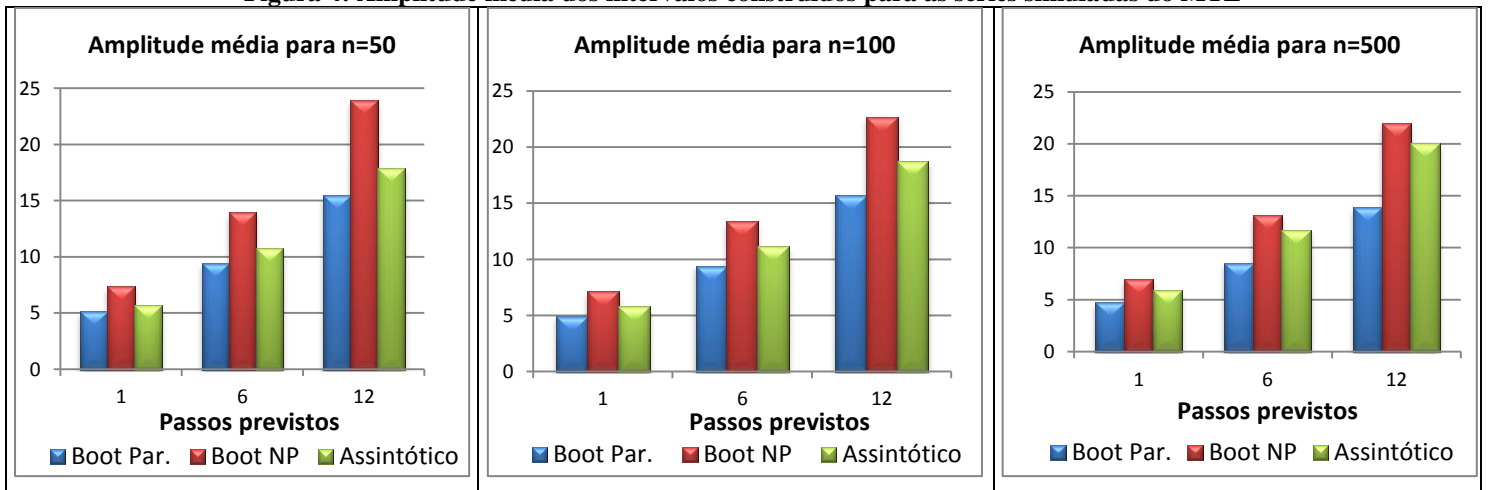
Para melhor visualizar os resultados obtidos com a simulação do MTL foram construídos os gráficos da Figura 3, os quais apresentam a cobertura dos intervalos de previsão para  $k=1$ ,  $k=6$  e  $k=12$ , considerando o tamanho da série.

**Figura 3: Cobertura média dos intervalos construídos para as séries simuladas do MTL**



Além dos resultados apresentados anteriormente, também foi feito um estudo sobre a amplitude dos intervalos de previsão, apresentado na Figura 4.

**Figura 4: Amplitude média dos intervalos construídos para as séries simuladas do MTL**



### 5.3 Resultados para o MEB

Para o MEB, assim como no MTL, foi feito um estudo com diferentes simulações variando apenas o tamanho da série e o número de previsões desejadas ( $k$ ). Os valores dos hiperparâmetros foram fixados em  $\sigma_\eta^2 = 0,50$ ,  $\sigma_\xi^2 = 0,01$ ,  $\sigma_\omega^2 = 0,10$  e  $\sigma_\varepsilon^2 = 1,00$ . Ressalta-se que o artigo de Rodriguez e Ruiz (2009) também não apresentou nenhum resultado para este modelo, sendo esse estudo mais uma ampliação da proposta dos autores.

Os resultados das simulações para as estimativas pontuais dos hiperparâmetros e para as previsões são apresentados, respectivamente, nas Tabelas 5 e 6, considerando uma cobertura nominal de 95%:

**Tabela 5: Estimativas dos hiperparâmetros para as séries simuladas do MEB**

Tamanho da série	Hiperparâmetros	Valor real	Valor Estimado		
			Boot Par.	Boot NP	EMV
50	$\sigma_n^2$	0,50	0,444	0,427	0,461
	$\sigma_\xi^2$	0,01	0,012	0,016	0,011
	$\sigma_\omega^2$	0,10	0,108	0,095	0,104
	$\sigma_\varepsilon^2$	1,00	0,985	1,119	1,008
100	$\sigma_n^2$	0,50	0,470	0,436	0,470
	$\sigma_\xi^2$	0,01	0,011	0,013	0,011
	$\sigma_\omega^2$	0,10	0,104	0,097	0,103
	$\sigma_\varepsilon^2$	1,00	1,010	1,090	1,017
500	$\sigma_\eta^2$	0,50	0,504	0,485	0,504
	$\sigma_\xi^2$	0,01	0,010	0,011	0,010
	$\sigma_\omega^2$	0,10	0,101	0,099	0,101
	$\sigma_\varepsilon^2$	1,00	0,991	1,015	0,991

A partir da Tabela 5, observa-se que as estimativas são semelhantes, com uma precisão de duas casas decimais para o sigma 2 Csi independente do tamanho da série. Destaca-se que para  $n=500$  os resultados obtidos pelo método Boot Par. e pelo EMV são os mesmos. Uma vez mais, conclui-se que os valores obtidos nos procedimentos *bootstrap* estão bem próximos das estimativas de máxima verossimilhança, podendo ser utilizado na construção de intervalos de confiança para os hiperparâmetros.

**Tabela 6: Intervalos de previsão para as séries simuladas do MEB**

Tamanho da série	K	Cobertura média 95%			Cobertura média nas caudas						Amplitude média		
		Boot Par.	Boot NP	Assintótico	Boot Par.		Boot NP		Assintótico		Boot Par.	Boot NP	Assintótico
					Abaixo	Acima	Abaixo	Acima	Abaixo	Acima			
50	1	0,874	0,964	0,870	0,056	0,070	0,014	0,022	0,062	0,068	5,859	8,269	5,834
	6	0,862	0,955	0,907	0,060	0,078	0,023	0,022	0,041	0,052	9,721	14,680	11,291
	12	0,811	0,929	0,857	0,093	0,096	0,030	0,041	0,067	0,076	15,897	24,716	18,543
100	1	0,871	0,967	0,900	0,082	0,047	0,017	0,016	0,061	0,039	5,576	7,965	5,967
	6	0,869	0,950	0,918	0,069	0,062	0,026	0,024	0,048	0,034	9,664	14,071	11,698
	12	0,839	0,925	0,890	0,086	0,075	0,044	0,031	0,063	0,047	16,060	23,305	19,392
500	1	0,848	0,958	0,896	0,084	0,068	0,021	0,021	0,055	0,049	5,282	7,743	6,079
	6	0,842	0,970	0,948	0,088	0,070	0,021	0,009	0,034	0,018	8,920	13,803	12,292
	12	0,822	0,959	0,940	0,104	0,074	0,023	0,018	0,035	0,025	14,371	22,566	20,598

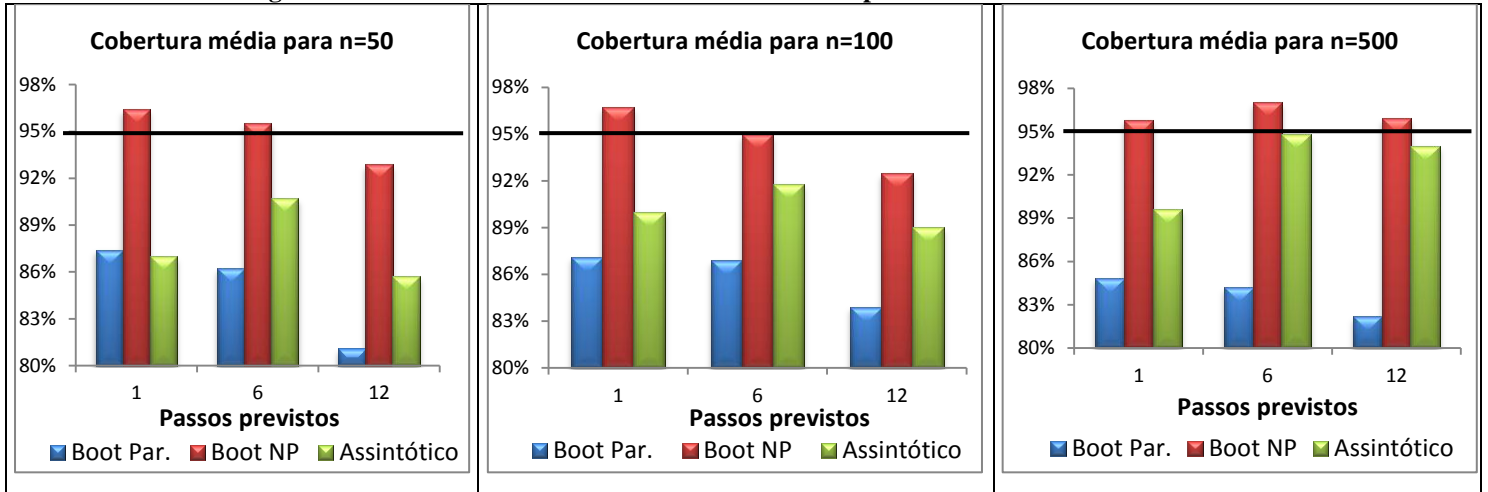
A partir da Tabela 6, pode-se dizer que em geral, as coberturas médias dos intervalos obtidos pelo método *bootstrap* não-paramétrico ficaram mais próximas de 95%. Adicionalmente, verifica-se que com o aumento do tamanho da amostra, diminui-se a amplitude dos intervalos de previsão, independente da metodologia utilizada.

Diferente dos resultados apresentados no modelo MNL e MTL os intervalos de previsão obtidos pelo método *bootstrap* não-paramétrico foram melhores que os demais métodos de construção de intervalos de previsão, independente do tamanho da série. Ressalta-se ainda o fato deles apresentarem a maior amplitude em todos os passos previstos e também serem os mais simétricos. No entanto, como visto nos outros modelos, o *bootstrap* paramétrico continuou apresentando os piores resultados.

Conclui-se ainda, assim como visto nos resultados do MNL e do MTL, que independente do tamanho da série e do valor de  $k$ , todos os intervalos mostraram-se ser simétricos, e também, que sempre as maiores amplitudes foram observadas nos intervalos *bootstrap* não-paramétricos e as menores amplitudes nos intervalos *bootstrap* paramétrico.

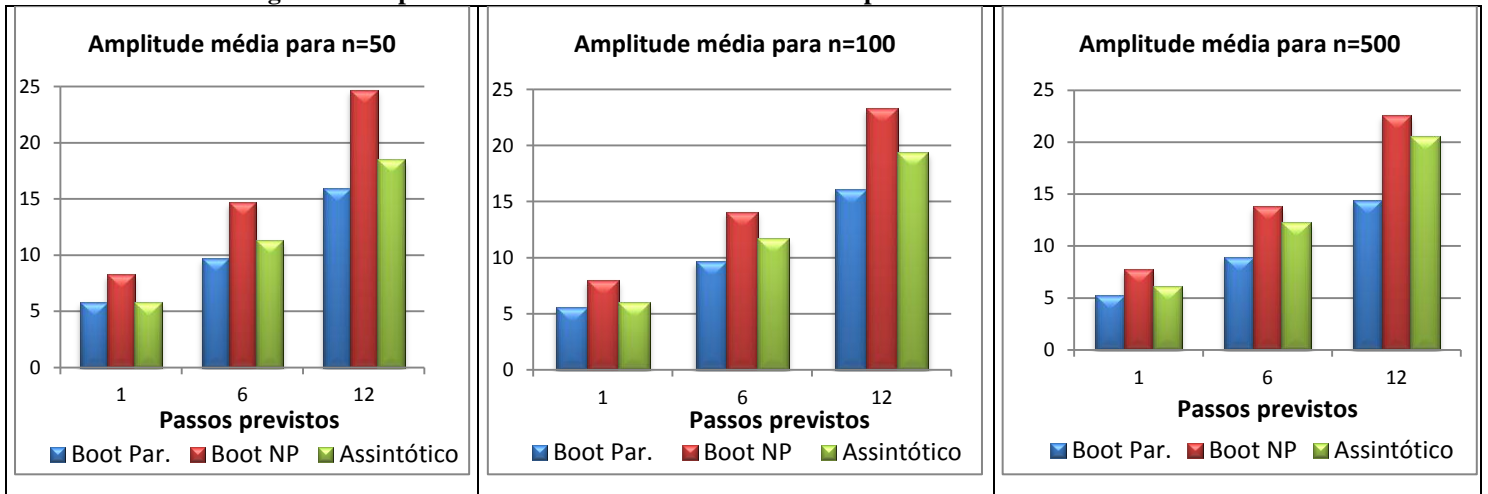
Para melhor visualizar os resultados obtidos com a simulação do MEB foram construídos os gráficos da Figura 5, os quais apresentam a cobertura dos intervalos de previsão para  $k=1$ ,  $k=6$  e  $k=12$ , considerando o tamanho da série.

**Figura 5: Cobertura média dos intervalos construídos para as séries simuladas do MEB**



Além dos resultados apresentados anteriormente, também foi feito um estudo sobre a amplitude dos intervalos de previsão, apresentado na Figura 6.

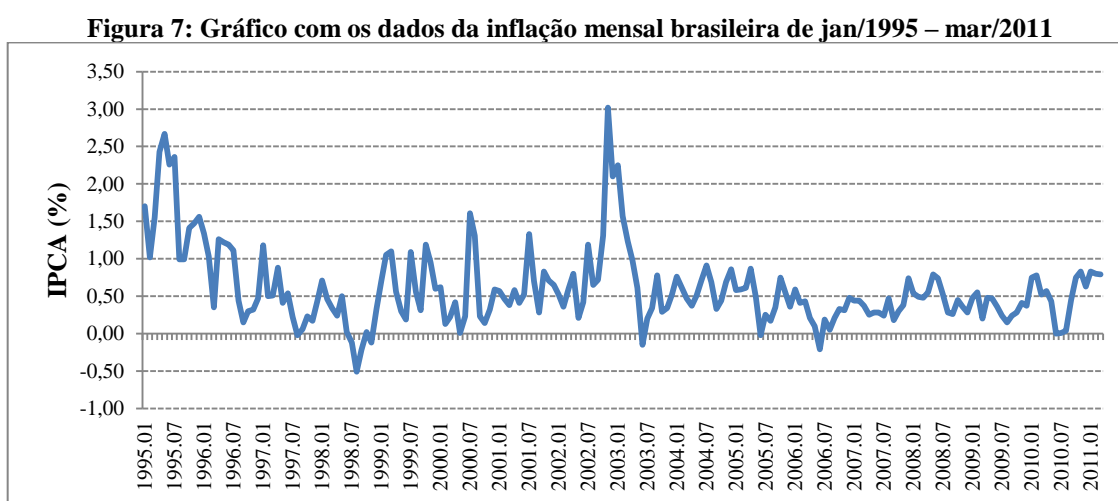
**Figura 6: Amplitude média dos intervalos construídos para as séries simuladas do MEB**



## 6. Aplicação a uma série real

Neste capítulo são estudados os dados mensais da inflação no Brasil por meio do Índice Nacional de Preços ao Consumidor Amplo (IPCA), no período de janeiro/1995 até março/2011, como forma de exemplificar a metodologia de construção de intervalos de previsão apresentada neste trabalho. Os dados foram disponibilizados no site do IPEA – Instituto de Pesquisas Econômicas Aplicadas, mas foram coletados e trabalhados pelo Instituto Brasileiro de Geografia e Estatística (IBGE), no departamento de Sistema Nacional de Índices e Preços ao Consumidor (IBGE/SNIPC).

A série de dados pode ser melhor visualizada na figura a seguir.



Fonte: Instituto Brasileiro de Geografia e Estatística, Sistema Nacional de Índices de Preços ao Consumidor (site do IPEA: <http://www.ipeadata.gov.br/Default.aspx>).

A partir da Figura 7 observa-se que a série IPCA varia de forma constante com o tempo, sendo estacionária em média. No entanto, nota-se a existência de alguns possíveis pontos discrepantes como o valor de 3,02% em nov/2002, 2,10% em dez/2002 e 2,25% em jan/2003, uma vez que os dados se distanciaram muito do nível que a série mantinha e também observa-se uma leve tendência decrescente no período de jan/1995 a nov/1998.

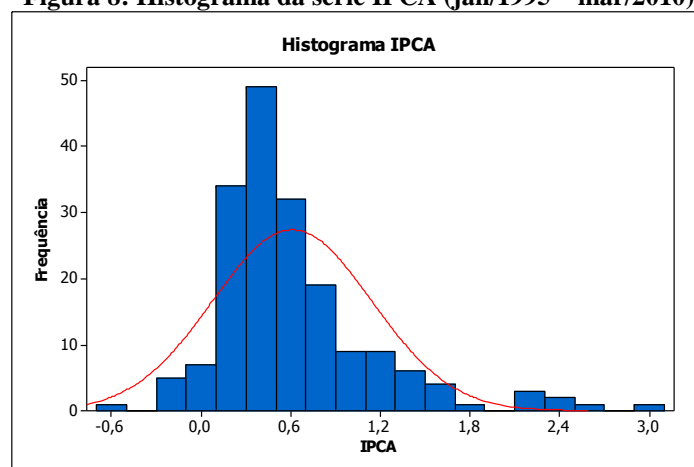
A fim de utilizar a metodologia desenvolvida neste estudo foram utilizados os dados de jan/1995 até mar/2010 para a modelagem da série e os dados de abr/2010 até mar/2011 foram guardados e comparados com os valores previstos obtidos com os diversos métodos para construção de intervalos de previsão. A seguir será apresentada uma análise exploratória da série IPCA no período da modelagem.

**Tabela 7: Estatísticas descritivas da série do IPCA (jan/1995 – mar/2010)**

Estatísticas Descritivas	
Nº de observações	183
Média	0,61
Desvio-padrão	0,53
Mínimo	-0,51
Mediana	0,48
Máximo	3,02
Assimetria	1,77
Curtose	4,31

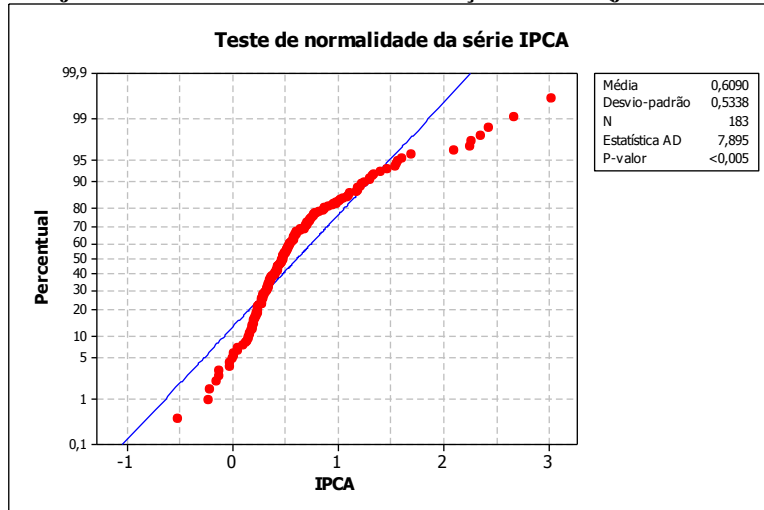
Observa-se a partir da Tabela 7 que o valor da inflação no período de jan/1995 a mar/2010 foi em média igual a 0,61%, com uma dispersão de 0,53% em torno desta. Além disso, nota-se que 50% dos valores observados foram inferiores a 0,47% e que eles estão variando entre -0,51% e 3,02%. Os dados também apresentam uma assimetria positiva (1,77) e uma curva de distribuição do tipo leptocúrtica, já que o valor da curtose foi maior que três (4,31), como pode ser visualizado na Figura 8.

**Figura 8: Histograma da série IPCA (jan/1995 – mar/2010)**

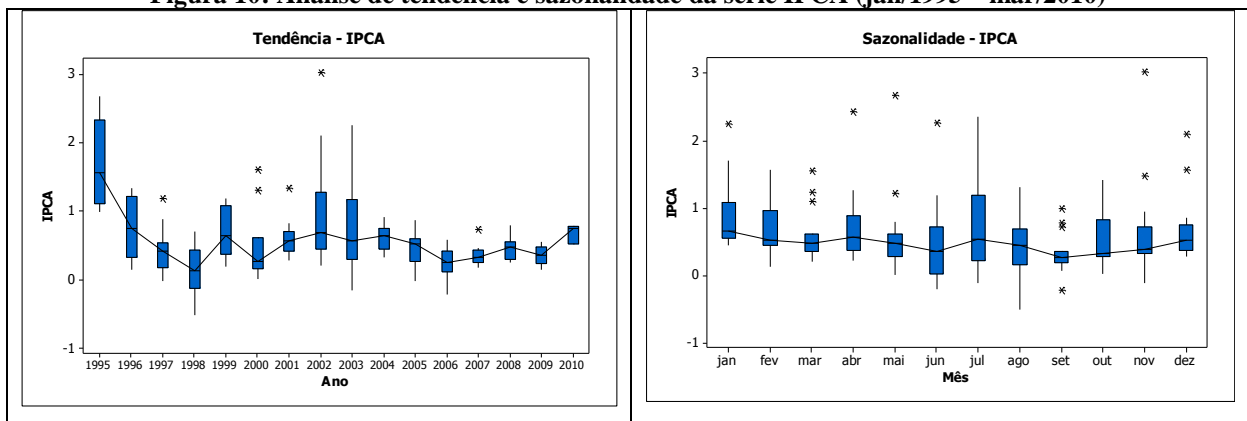


A não normalidade dos dados pode ser confirmada pela Figura 9, que mostra o ajuste dos dados quando são aproximados pela distribuição Normal padrão e o valor do teste Anderson-Darling. Nota-se que a hipótese de normalidade foi rejeitada, uma vez que o p-valor calculado foi menor que o nível de significância adotado (0,05) para o teste.

**Figura 9: Ajuste da série IPCA à uma distribuição Normal (jan/1995 – mar/2010)**



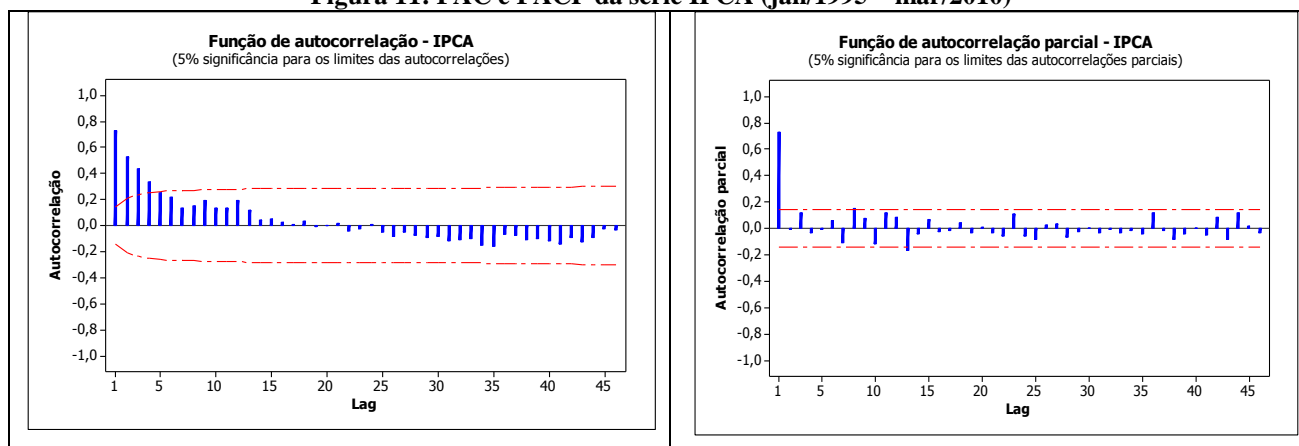
**Figura 10: Análise de tendência e sazonalidade da série IPCA (jan/1995 – mar/2010)**



A partir da Figura 10 observa-se que a série IPCA não apresenta uma tendência definida em todo seu período de estudo e também não apresenta sazonalidade, uma vez que as medianas dos boxplot por mês estão quase alinhadas. No entanto, são observados vários valores discrepantes.



**Figura 11: FAC e FACP da série IPCA (jan/1995 – mar/2010)**



Por meio da Figura 11 é possível perceber que o FAC apresenta um decaimento rápido, o que confirma a estacionariedade da série e pela FACP confirma-se que a série não apresenta sazonalidade já que o lag 12 não foi significativo.

A partir da análise exploratória da série IPCA fica evidente a não existência dos componentes de tendência e sazonalidade na série, e sim, um componente de nível acrescido de um erro sendo necessário, portanto, utilizar o modelo MNL. Após executar o programa foram obtidas as estimativas dos hiperparâmetros da série em estudo e os seguintes intervalos de previsão para os valores previstos, os quais são apresentados nas Tabelas 8 e 9.

**Tabela 8: Estimativas pontuais para os hiperparâmetros da série do IPCA**

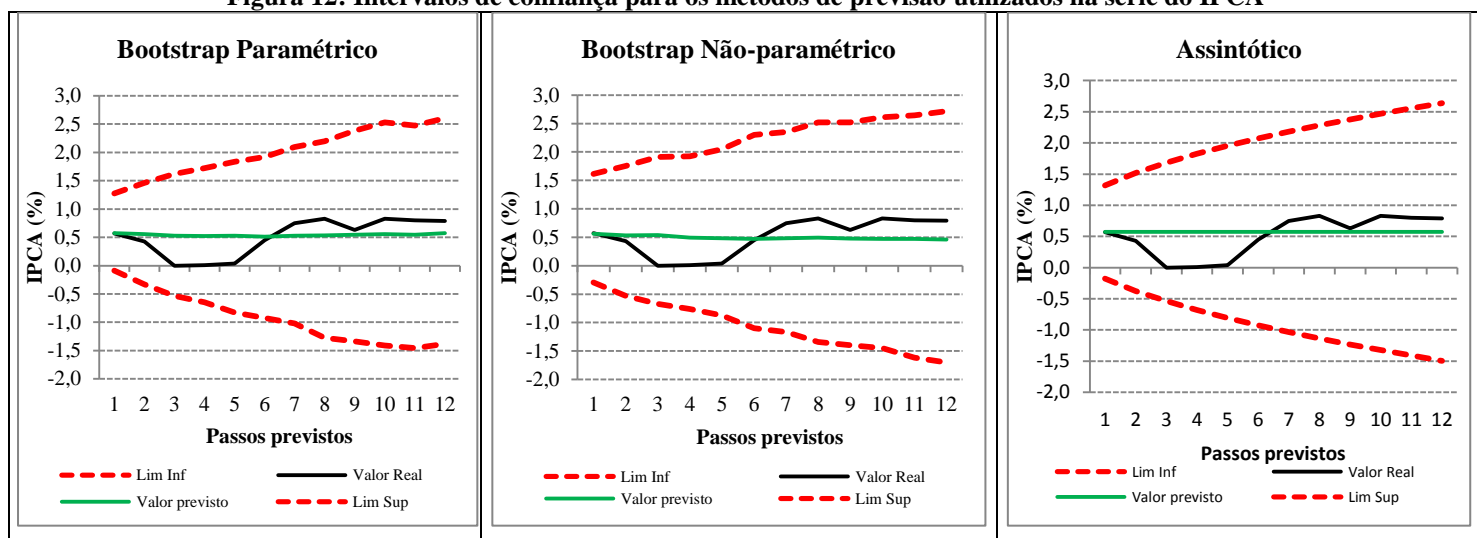
Tamanho da série	Hiperparâmetros	Valor Estimado		
		Boot Par.	Boot NP	EMV
<b>182</b>	$\sigma_{\eta}^2$	0,08817	0,08898	0,08787
	$\sigma_{\varepsilon}^2$	0,03219	0,03237	0,03244

Com os resultados da Tabela 8 observa-se que os valores obtidos pelos métodos *bootstrap* estão muito próximos dos valores obtidos pelo estimador de máxima verossimilhança, sendo que a diferença fica na terceira casa decimal para o  $\sigma_{\eta}^2$  e na quarta casa decimal para o  $\sigma_{\varepsilon}^2$ .

**Tabela 9: Intervalos de previsão para a série do IPCA (período previsto: abril/2010 – março/2011)**

K	Valor Real	Boot Par			Boot NP			Assintótico		
		Valor previsto	Lim Inf	Lim Sup	Valor previsto	Lim Inf	Lim Sup	Valor previsto	Lim Inf	Lim Sup
1	0,57	0,57	-0,09	1,27	0,56	-0,30	1,61	0,57	-0,18	1,32
2	0,43	0,56	-0,33	1,46	0,54	-0,53	1,76	0,57	-0,37	1,52
3	0,00	0,53	-0,53	1,62	0,54	-0,67	1,91	0,57	-0,54	1,68
4	0,01	0,52	-0,64	1,72	0,50	-0,76	1,93	0,57	-0,68	1,83
5	0,04	0,53	-0,83	1,83	0,48	-0,87	2,05	0,57	-0,81	1,95
6	0,45	0,51	-0,92	1,92	0,47	-1,10	2,30	0,57	-0,93	2,07
7	0,75	0,53	-1,02	2,10	0,49	-1,17	2,35	0,57	-1,04	2,18
8	0,83	0,54	-1,27	2,20	0,49	-1,34	2,52	0,57	-1,14	2,28
9	0,63	0,55	-1,34	2,38	0,48	-1,40	2,52	0,57	-1,23	2,38
10	0,83	0,56	-1,41	2,53	0,47	-1,45	2,61	0,57	-1,32	2,47
11	0,80	0,54	-1,46	2,47	0,47	-1,62	2,65	0,57	-1,41	2,56
12	0,79	0,57	-1,38	2,60	0,46	-1,70	2,72	0,57	-1,49	2,64

**Figura 12: Intervalos de confiança para os métodos de previsão utilizados na série do IPCA**



A partir da Tabela 9 e da Figura 12 observa-se que os métodos de previsão apresentaram o mesmo comportamento observado nas simulações do MNL, isto é, com o aumento dos passos previstos as amplitudes aumentaram, sendo as maiores amplitudes observadas nos intervalos Boot NP e as menores nos intervalos Boot Par. Adicionalmente, vale ressaltar a eficiência dos intervalos ao compreender todos os valores reais e destaca-se o acompanhamento dos limites inferior e superior dos intervalos Boot NP à trajetória dos valores reais, sendo que o mesmo não ocorreu nos demais intervalos.

Com relação aos valores previstos, como se esperava, eles foram mais próximos dos valores reais nos primeiros passos, sendo o melhor desempenho observado nas previsões do Boot Par.

## 7. Considerações Finais

Ao longo da realização desta dissertação, tornou-se possível conhecer melhor os modelos estruturais, as suas técnicas de estimação e previsão, e, por meio deles, foi desenvolvido um algoritmo em linguagem Ox capaz de implementar todas essas técnicas para os principais modelos.

No Capítulo 3, a técnica de reamostragem *bootstrap* foi abordada, apresentando-se a sua aplicabilidade a fim de fazer inferência sobre os hiperparâmetros e as previsões dos modelos estruturais. Mediante os resultados apresentados no Capítulo 5, notou-se a eficiência do processo de estimação pelas implementações em Ox, utilizando-se tanto o método de máxima verossimilhança quanto a técnica de reamostragem *bootstrap* (paramétrico e não-paramétrico).

Em geral, as estimativas obtidas dos valores esperados dos hiperparâmetros ficaram bem próximas dos seus valores reais. Foi interessante verificar que, adotando-se o maior tamanho da série, as estimativas dos hiperparâmetros se aproximavam mais dos valores pré-definidos, podendo-se concluir que os valores *bootstrap* e os estimadores de máxima verossimilhança são consistentes.

A respeito das previsões, percebeu-se que, com o aumento do tamanho das séries simuladas, as amplitudes intervalares decresceram. No entanto, as coberturas dos intervalos de previsão não se mantiveram próximas do nível de confiança assumido para as diferentes técnicas, sendo verificada uma performance melhor nos intervalos *bootstrap* não-paramétrico quando  $q=0,1$  no MNL e de forma geral no MEB. Já os intervalos assintóticos foram melhores quando  $q=1$  e  $q=2$  no MNL e de forma geral no MTL. Por sua vez, os intervalos *bootstrap* paramétrico não foram eficientes em nenhum modelo.

No[ Capítulo 6 foi apresentada a aplicação da metodologia estudada em uma série temporal real, o IPCA no período de jan/2005 até mar/2011. Tomou-se uma série cujo comportamento caracteriza-se como sendo um passeio aleatório acrescido de erro. Ajustou-se o MNL com a intenção de verificar se os intervalos de previsão construídos a partir desta modelagem continham os valores reais da série. Como resultado, foi possível verificar que tanto os intervalos assintóticos quanto os intervalos *bootstrap*, apresentaram um bom comportamento por compreenderem todos os valores reais, no entanto os intervalos do Boot NP acompanharam um pouco melhor a trajetória dos valores reais. Adicionalmente, assim como nas simulações, verificou-se que os intervalos Boot NP apresentaram as maiores

amplitudes e que os intervalos Boot Par apresentaram as menores. Com relação ao valores previstos, como se esperava, eles foram mais próximos dos valores reais nos primeiros passos de previsão, sendo o método Boot Par o que apresentou os melhores resultados.

Diante do estudo apresentado, concluiu-se que a implementação da estimação dos hiperparâmetros dos modelos estruturais e do cálculo dos valores previstos e seus intervalos utilizando a linguagem Ox mostrou-se satisfatória, tanto em dados simulados, quanto nos dados da série temporal real. No entanto, os resultados obtidos não foram como o esperado, sendo o método de previsão assintótico apontado como o mais adequado.

Tópicos para pesquisas futuras na área incluem a ampliação dos estudos de simulação para modelos incompletos, a análise de outras séries reais, bem como a implementação de testes de hipóteses para testar a significância dos parâmetros e hiperparâmetros em modelos estruturais.

## Referências Bibliográficas

- Doornik, J.A. (1999). *Ox: An Object-Oriented Matrix Language*. 3. ed. London: Timberlake Consultants Press.
- Efron, B. (1979). Bootstrap methods: another look at the jackknife. *The Annals of Statistics*, 7, 1-26.
- Efron, B. & Tibshirani, R. (1993). *An Introduction to the Bootstrap*. London: Chapman and Hall.
- Franco, G.C. (1998). *Bootstrap em Modelos Estruturais: Construção de intervalos de confiança e testes de hipóteses*. Tese (Doutorado em Engenharia Elétrica) – PUC, Rio de Janeiro.
- Franco, G. C., Gamerman, D. & Santos, T. R. (2009) *Modelos de Espaço de Estados: Abordagens Clássica e Bayesiana*. Livro do minicurso apresentado na 13ª Escola de Séries Temporais e Econometria - SP.
- Franco, G. C., Santos, T. R., Ribeiro, J. A. & Cruz, F. R. (2008) Confidence intervals for hyperparameters in structural models. *Communications in Statistics: Simulation and Computation*, 37 (3), 486-497.
- Harrison, P. J. & Stevens, C. F. (1971). A Bayesian approach to short-term forecasting. *Operational Research Quarterly*, 22, 341-362.
- Harrison, P. J. & Stevens, C. F. (1976). Bayesian forecasting. *Journal of the Royal Statistical Society, Series B*, 38, 205-247.
- Harvey, A. C. (1989). *Forecasting, Structural Time Series Models and the Kalman Fitter*. Cambridge: University Press.
- Harvey, A. C. (1993). *Time Series Models*. Maylands: Prentice Hall.
- Harvey, A. C. (2001). Testing unobserved component models. *Journal of Forecasting*. **20**, 1-19.
- Holt, C. C. (1957) *Forecasting seasonals and trends by exponentially weighted moving averages*. ONR Research Memorandum 52, Carnegie Institute of Technology, Pittsburgh, Pennsylvania.
- Kalman, R.E. (1960). A new approach to linear filtering and prediction problems. *Trans. ASME J. Basic Eng.*, 82, 35-45.
- Koopman, S. J., Shephard, N. & Doornik, J. (1999). Statistical algorithms for models in state space using SsfPack 2.2. *Econometrics Journal*, 2, 113-166.

Morettin, P.A. e Tolo, C.M.C. (2004). *Análise de Séries Temporais*. Edgard Blücher, São Paulo.

Muth, J. F. (1960). Optimal properties of exponentially weighted forecasts. *Journal of the American Statistical Association*, 55, 299-305.

Nervole, M. & Wage, S. (1964). On the optimality of adaptive forecasting. *Management Science*, 10, 207-229.

Press, W. H., Flannery, B. P., Teukolsky, S. A. & Vetterling, W. T. (1988). *Numerical Recipes in C*. New York: Cambridge University Press.

Rodriguez, A. & Ruiz, E. (2009) Bootstrap prediction intervals in state-space models. *Journal of Time Series Analysis* 30, 167-178.

Stoffer, D. S. & Wall, K. D. (1991) Bootstrapping State-Space Models: Gaussian Maximum Likelihood Estimation and the Kalman Filter. *American Statistical Association* Vol.86, No. 416, 1024-1033.

Theil, H. & Wage, S. (1964). Some observations on adaptive forecasting. *Management Science*, 10, 198-206.

Wall, K. D. & Stoffer, D. S. (2002) A State-Space approach to bootstrapping conditional forecasts in ARMA models. *Journal of Time Series Analysis* Vol. 23, No. 6, 733-751.

West, M & Harrison, J. (1997). *Bayesian Forecasting and Dynamic Models*. New York: Springer.

Winters, P. R. (1960) Forecasting sales by exponentially weighted moving averages. *Management Science*, 6, 324-342.

## Apêndice 1 – Matriz de Covariância

Cálculo da matriz de covariância para o modelo MNL:

$$a_t = \mu_t$$

$$P_t = Cov(a_t) = Cov(\mu_t, \mu_t) = Var(\mu_t) = P_t^{\mu}$$

Cálculo da matriz  $P_t$  para o modelo MTL:

$$a_t = \begin{bmatrix} \mu_t \\ \beta_t \end{bmatrix}$$

$$P_t = Cov(a_t) = Cov \begin{bmatrix} \mu_t \\ \beta_t \end{bmatrix} = \begin{bmatrix} Cov(\mu_t, \mu_t) & Cov(\mu_t, \beta_t) \\ Cov(\beta_t, \mu_t) & Cov(\beta_t, \beta_t) \end{bmatrix} = \begin{bmatrix} Var(\mu_t) & \rho \\ \rho & Var(\beta_t) \end{bmatrix} = \begin{bmatrix} P_t^{\mu t} & \rho \\ \rho & P_t^{\beta t} \end{bmatrix}$$

sendo:  $\rho = Cov(\mu_t, \beta_t)$  substituindo  $\mu_t$  de acordo com a equação 2.5 do relatório, temos:

$$\rho = Cov(\mu_{t-1} + \beta_{t-1} + \eta_t, \beta_{t-1} + \xi_t)$$

$$\rho = Cov(\mu_{t-1}, \beta_{t-1}) + Cov(\mu_{t-1}, \xi_t) + Cov(\beta_{t-1}, \beta_{t-1}) + Cov(\beta_{t-1}, \xi_t) + Cov(\eta_t, \beta_{t-1}) + Cov(\eta_t, \xi_t)$$

$$\rho = Var(\beta_{t-1})$$

Logo,

$$P_t = \begin{bmatrix} P_t^{\mu t} & Var(\beta_{t-1}) \\ Var(\beta_{t-1}) & P_t^{\beta t} \end{bmatrix} = \begin{bmatrix} P_t^{\mu t} & P_t^{\beta_{t-1}} \\ P_t^{\beta_{t-1}} & P_t^{\beta t} \end{bmatrix}$$

Por sua vez, as matrizes do MEB, supondo  $s=3$ , ficam com as seguintes quantidades:

$$a_t = \begin{bmatrix} \mu_t \\ \beta_t \\ \gamma_t \\ \gamma_{t-1} \end{bmatrix}$$

$$P_t = Cov(a_t) = Cov \begin{bmatrix} \mu_t \\ \beta_t \\ \gamma_t \\ \gamma_{t-1} \end{bmatrix} = \begin{bmatrix} Cov(\mu_t, \mu_t) & Cov(\mu_t, \beta_t) & Cov(\mu_t, \gamma_t) & Cov(\mu_t, \gamma_{t-1}) \\ Cov(\beta_t, \mu_t) & Cov(\beta_t, \beta_t) & Cov(\beta_t, \gamma_t) & Cov(\beta_t, \gamma_{t-1}) \\ Cov(\gamma_t, \mu_t) & Cov(\gamma_t, \beta_t) & Cov(\gamma_t, \gamma_t) & Cov(\gamma_t, \gamma_{t-1}) \\ Cov(\gamma_{t-1}, \mu_t) & Cov(\gamma_{t-1}, \beta_t) & Cov(\gamma_{t-1}, \gamma_t) & Cov(\gamma_{t-1}, \gamma_{t-1}) \end{bmatrix}$$

$$= \begin{bmatrix} Var(\mu_t) & \rho & 0 & 0 \\ \rho & Var(\beta_t) & 0 & 0 \\ 0 & 0 & Var(\gamma_t) & w \\ 0 & 0 & w & Var(\gamma_{t-1}) \end{bmatrix} = \begin{bmatrix} P_t^{\mu t} & \rho & 0 & 0 \\ \rho & P_t^{\beta t} & 0 & 0 \\ 0 & 0 & P_t^{\gamma t} & w \\ 0 & 0 & w & P_t^{\gamma_{t-1}} \end{bmatrix}$$

sendo:  $w = Cov(\gamma_t, \gamma_{t-1})$  substituindo  $\gamma_t$  de acordo com a equação 2.10 do relatório, temos:

$$\begin{aligned}
w &= Cov(-\gamma_{t-1} - \gamma_{t-2} + \omega_t, \gamma_{t-1}) \\
w &= -Cov(\gamma_{t-1}, \gamma_{t-1}) - Cov(\gamma_{t-2}, \gamma_{t-1}) + Cov(\omega_t, \gamma_{t-1}) \\
w &= -Var(\gamma_{t-1}) - Cov(\gamma_{t-2}, -\gamma_{t-2} - \gamma_{t-3} + \omega_{t-1}) \\
w &= -Var(\gamma_{t-1}) + Cov(\gamma_{t-2}, \gamma_{t-2}) + Cov(\gamma_{t-2}, \gamma_{t-3}) - Cov(\gamma_{t-2}, \omega_{t-1}) \\
w &= -Var(\gamma_{t-1}) + Var(\gamma_{t-2}) + Cov(-\gamma_{t-3} - \gamma_{t-4} + \omega_{t-2}, \gamma_{t-3}) \\
w &= -Var(\gamma_{t-1}) + Var(\gamma_{t-2}) - Cov(\gamma_{t-3}, \gamma_{t-3}) - Cov(-\gamma_{t-4}, \gamma_{t-3}) + Cov(\omega_{t-2}, \gamma_{t-3}) \\
w &= -Var(\gamma_{t-1}) + Var(\gamma_{t-2}) - Var(\gamma_{t-3}) - Cov(-\gamma_{t-4}, -\gamma_{t-4} - \gamma_{t-5} + \omega_{t-3}) \\
&\vdots \\
w &= \sum_{i=1}^{n-1} (-1)^i Var(\gamma_{t-i})
\end{aligned}$$

Logo,

$$P_t = \begin{bmatrix} P_t^{\mu_t} & P_t^{\beta_{t-1}} & & & 0 & 0 \\ P_t^{\beta_{t-1}} & P_t^{\beta_t} & & & 0 & 0 \\ & 0 & 0 & & & \\ & 0 & 0 & P_t^{\gamma_t} & & \\ & & & \sum_{i=1}^{n-1} (-1)^i Var(\gamma_{t-i}) & & \\ & & & & P_t^{\gamma_{t-1}} & \end{bmatrix} = \begin{bmatrix} P_t^{\mu_t} & P_t^{\beta_{t-1}} & 0 & 0 \\ P_t^{\beta_{t-1}} & P_t^{\beta_t} & 0 & 0 \\ 0 & 0 & P_t^{\gamma_t} & P_t^w \\ 0 & 0 & P_t^w & P_t^{\gamma_{t-1}} \end{bmatrix}$$



## Apêndice 2 – Equações de Previsão *Bootstrap* Paramétrico

$$\hat{y}_{T+k|T} = \hat{\mu}_{T+k|T} + \hat{\gamma}_{T+k|T} + \hat{\delta}_{T+k|T} + \hat{\varepsilon}_{T+k|T}$$

### EQUAÇÕES DE PREVISÃO PARA O MNL

$$y_t = \mu_t + \varepsilon_t, \quad \varepsilon_t \sim N(0, \sigma_\varepsilon^2)$$
$$\mu_t = \mu_{t-1} + \eta_t, \quad \eta_t \sim N(0, \sigma_\eta^2)$$

**Suponha k=1, temos:**

$$\hat{\mu}_{T+1}^* = \hat{\mu}_T^* + \hat{\eta}_{T+1}^*$$

$$\hat{y}_{T+1}^* = \hat{\mu}_{T+1}^* + \hat{\varepsilon}_{T+1}^*$$

$$\hat{y}_{T+1}^* = \hat{\mu}_T^* + \hat{\eta}_{T+1}^* + \hat{\varepsilon}_{T+1}^*$$

**Suponha k=2, temos:**

$$\hat{\mu}_{T+2}^* = \hat{\mu}_{T+1}^* + \hat{\eta}_{T+2}^*$$

$$\hat{y}_{T+2}^* = \hat{\mu}_{T+2}^* + \hat{\varepsilon}_{T+2}^*$$

$$\hat{y}_{T+2}^* = \hat{\mu}_{T+1}^* + \hat{\eta}_{T+2}^* + \hat{\varepsilon}_{T+2}^*$$

$$\hat{y}_{T+2}^* = \hat{\mu}_T^* + \hat{\eta}_{T+1}^* + \hat{\eta}_{T+2}^* + \hat{\varepsilon}_{T+2}^*$$

**Suponha k=3, temos:**

$$\hat{\mu}_{T+3}^* = \hat{\mu}_{T+2}^* + \hat{\eta}_{T+3}^*$$

$$\hat{y}_{T+3}^* = \hat{\mu}_{T+3}^* + \hat{\varepsilon}_{T+3}^*$$

$$\hat{y}_{T+3}^* = \hat{\mu}_{T+2}^* + \hat{\eta}_{T+3}^* + \hat{\varepsilon}_{T+3}^*$$

$$\hat{y}_{T+3}^* = \hat{\mu}_{T+1}^* + \hat{\eta}_{T+2}^* + \hat{\eta}_{T+3}^* + \hat{\varepsilon}_{T+3}^*$$

$$\hat{y}_{T+3}^* = \hat{\mu}_T^* + \hat{\eta}_{T+1}^* + \hat{\eta}_{T+2}^* + \hat{\eta}_{T+3}^* + \hat{\varepsilon}_{T+3}^*$$

**Fórmula Geral:**

$$\hat{y}_{T+k}^* = \hat{\mu}_T^* + \sum_{j=1}^k \hat{\eta}_{T+j}^* + \hat{\varepsilon}_{T+k}^*$$

## EQUAÇÕES DE PREVISÃO PARA O MTL

$$\begin{aligned}y_t &= \mu_t + \varepsilon_t, & \varepsilon_t &\sim N(0, \sigma_\varepsilon^2) \\ \mu_t &= \mu_{t-1} + \beta_{t-1} + \eta_t, & \eta_t &\sim N(0, \sigma_\eta^2) \\ \beta_t &= \beta_{t-1} + \xi_t, & \xi_t &\sim N(0, \sigma_\xi^2)\end{aligned}$$

**Suponha k=1, temos:**

$$\hat{\mu}_{T+1}^* = \hat{\mu}_T^* + \hat{\beta}_T^* + \hat{\eta}_{T+1}^*$$

$$\hat{\beta}_{T+1}^* = \hat{\beta}_T^* + \hat{\xi}_{T+1}^*$$

$$\hat{y}_{T+1}^* = \hat{\mu}_{T+1}^* + \hat{\varepsilon}_{T+1}^*$$

$$\hat{y}_{T+1}^* = \hat{\mu}_T^* + \hat{\beta}_T^* + \hat{\eta}_{T+1}^* + \hat{\varepsilon}_{T+1}^*$$

**Suponha k=2, temos:**

$$\hat{\mu}_{T+2}^* = \hat{\mu}_{T+1}^* + \hat{\beta}_{T+1}^* + \hat{\eta}_{T+2}^*$$

$$\hat{\beta}_{T+2}^* = \hat{\beta}_{T+1}^* + \hat{\xi}_{T+2}^*$$

$$\hat{y}_{T+2}^* = \hat{\mu}_{T+2}^* + \hat{\varepsilon}_{T+2}^*$$

$$\hat{y}_{T+2}^* = \hat{\mu}_{T+1}^* + \hat{\beta}_{T+1}^* + \hat{\eta}_{T+2}^* + \hat{\varepsilon}_{T+2}^*$$

$$\hat{y}_{T+2}^* = \hat{\mu}_T^* + \hat{\beta}_T^* + \hat{\eta}_{T+1}^* + \hat{\beta}_T^* + \hat{\xi}_{T+1}^* + \hat{\eta}_{T+2}^* + \hat{\varepsilon}_{T+2}^*$$

$$\hat{y}_{T+2}^* = \hat{\mu}_T^* + 2\hat{\beta}_T^* + \hat{\eta}_{T+1}^* + \hat{\eta}_{T+2}^* + \hat{\xi}_{T+1}^* + \hat{\varepsilon}_{T+2}^*$$

**Suponha k=3, temos:**

$$\hat{\mu}_{T+3}^* = \hat{\mu}_{T+2}^* + \hat{\beta}_{T+2}^* + \hat{\eta}_{T+3}^*$$

$$\hat{\beta}_{T+3}^* = \hat{\beta}_{T+2}^* + \hat{\xi}_{T+3}^*$$

$$\hat{y}_{T+3}^* = \hat{\mu}_{T+3}^* + \hat{\varepsilon}_{T+3}^*$$

$$\hat{y}_{T+3}^* = \hat{\mu}_{T+2}^* + \hat{\beta}_{T+2}^* + \hat{\eta}_{T+3}^* + \hat{\varepsilon}_{T+3}^*$$

$$\hat{y}_{T+3}^* = \hat{\mu}_{T+1}^* + \hat{\beta}_{T+1}^* + \hat{\eta}_{T+2}^* + \hat{\beta}_{T+1}^* + \hat{\xi}_{T+2}^* + \hat{\eta}_{T+3}^* + \hat{\varepsilon}_{T+3}^*$$

$$\hat{y}_{T+3}^* = \hat{\mu}_T^* + \hat{\beta}_T^* + \hat{\eta}_{T+1}^* + \hat{\beta}_T^* + \hat{\xi}_{T+1}^* + \hat{\eta}_{T+2}^* + \hat{\beta}_T^* + \hat{\xi}_{T+1}^* + \hat{\xi}_{T+2}^* + \hat{\eta}_{T+3}^* + \hat{\varepsilon}_{T+3}^*$$

$$\hat{y}_{T+3}^* = \hat{\mu}_T^* + 3\hat{\beta}_T^* + \hat{\eta}_{T+1}^* + \hat{\eta}_{T+2}^* + \hat{\eta}_{T+3}^* + 2\hat{\xi}_{T+1}^* + \hat{\xi}_{T+2}^* + \hat{\varepsilon}_{T+3}^*$$

**Fórmula Geral:**

$$\hat{y}_{T+k}^* = \hat{\mu}_T^* + k\hat{\beta}_T^* + \sum_{j=1}^k [\hat{\eta}_{T+j}^* + (k-j)\hat{\xi}_{T+j}^*] + \hat{\varepsilon}_{T+k}^*$$

### EQUAÇÕES DE PREVISÃO PARA O MEB

$$\begin{aligned}
 y_t &= \mu_t + \gamma_t + \varepsilon_t, & \varepsilon_t &\sim N(0, \sigma_\varepsilon^2) \\
 \mu_t &= \mu_{t-1} + \beta_{t-1} + \eta_t, & \eta_t &\sim N(0, \sigma_\eta^2) \\
 \beta_t &= \beta_{t-1} + \xi_t, & \xi_t &\sim N(0, \sigma_\xi^2) \\
 \gamma_t &= -\gamma_{t-1} - \dots - \gamma_{t-s+1} + \omega_t, & \omega_t &\sim N(0, \sigma_\omega^2)
 \end{aligned}$$

\*Neste estudo s=3.

**Suponha k=1, temos:**

$$\begin{aligned}
 \hat{\mu}_{T+1}^* &= \hat{\mu}_T^* + \hat{\beta}_T^* + \hat{\eta}_{T+1}^* \\
 \hat{\beta}_{T+1}^* &= \hat{\beta}_T^* + \hat{\xi}_{T+1}^* \\
 \hat{\gamma}_{T+1}^* &= -\hat{\gamma}_T^* - \hat{\gamma}_{T-1}^* + \hat{\omega}_{T+1}^* \\
 \hat{y}_{T+1}^* &= \hat{\mu}_{T+1}^* + \hat{\gamma}_{T+1}^* + \hat{\varepsilon}_{T+1}^* \\
 \hat{y}_{T+1}^* &= \hat{\mu}_T^* + \hat{\beta}_T^* + \hat{\eta}_{T+1}^* - \hat{\gamma}_T^* - \hat{\gamma}_{T-1}^* + \hat{\omega}_{T+1}^* + \hat{\varepsilon}_{T+1}^*
 \end{aligned}$$

**Suponha k=2, temos:**

$$\begin{aligned}
 \hat{\mu}_{T+2}^* &= \hat{\mu}_{T+1}^* + \hat{\beta}_{T+1}^* + \hat{\eta}_{T+2}^* \\
 \hat{\beta}_{T+2}^* &= \hat{\beta}_{T+1}^* + \hat{\xi}_{T+2}^* \\
 \hat{\gamma}_{T+2}^* &= -\hat{\gamma}_{T+1}^* - \hat{\gamma}_T^* + \hat{\omega}_{T+2}^* \\
 \hat{y}_{T+2}^* &= \hat{\mu}_{T+2}^* + \hat{\gamma}_{T+2}^* + \hat{\varepsilon}_{T+2}^* \\
 \hat{y}_{T+2}^* &= \hat{\mu}_{T+1}^* + \hat{\beta}_{T+1}^* + \hat{\eta}_{T+2}^* - \hat{\gamma}_{T+1}^* - \hat{\gamma}_T^* + \hat{\omega}_{T+2}^* + \hat{\varepsilon}_{T+2}^* \\
 \hat{y}_{T+2}^* &= \hat{\mu}_T^* + \hat{\beta}_T^* + \hat{\eta}_{T+1}^* + \hat{\beta}_{T+1}^* + \hat{\xi}_{T+1}^* + \hat{\eta}_{T+2}^* - \hat{\gamma}_{T+1}^* - \hat{\gamma}_T^* + \hat{\omega}_{T+2}^* + \hat{\varepsilon}_{T+2}^* \\
 \hat{y}_{T+2}^* &= \hat{\mu}_T^* + 2\hat{\beta}_T^* + \hat{\eta}_{T+1}^* + \hat{\eta}_{T+2}^* + \hat{\xi}_{T+1}^* - \hat{\gamma}_{T+1}^* - \hat{\gamma}_T^* + \hat{\omega}_{T+2}^* + \hat{\varepsilon}_{T+2}^* \\
 \hat{y}_{T+2}^* &= \hat{\mu}_T^* + 2\hat{\beta}_T^* + \hat{\eta}_{T+1}^* + \hat{\eta}_{T+2}^* + \hat{\xi}_{T+1}^* - \hat{\gamma}_{T+1}^* - \hat{\gamma}_T^* + \hat{\omega}_{T+2}^* + \hat{\varepsilon}_{T+2}^*
 \end{aligned}$$

**Suponha k=3, temos:**

$$\begin{aligned}
 \hat{\mu}_{T+3}^* &= \hat{\mu}_{T+2}^* + \hat{\beta}_{T+2}^* + \hat{\eta}_{T+3}^* \\
 \hat{\beta}_{T+3}^* &= \hat{\beta}_{T+2}^* + \hat{\xi}_{T+3}^* \\
 \hat{\gamma}_{T+3}^* &= -\hat{\gamma}_{T+2}^* - \hat{\gamma}_{T+1}^* + \hat{\omega}_{T+3}^* \\
 \hat{y}_{T+3}^* &= \hat{\mu}_{T+3}^* + \hat{\gamma}_{T+3}^* + \hat{\varepsilon}_{T+3}^* \\
 \hat{y}_{T+3}^* &= \hat{\mu}_{T+2}^* + \hat{\beta}_{T+2}^* + \hat{\eta}_{T+3}^* - \hat{\gamma}_{T+2}^* - \hat{\gamma}_{T+1}^* + \hat{\omega}_{T+3}^* + \hat{\varepsilon}_{T+3}^* \\
 \hat{y}_{T+3}^* &= \hat{\mu}_{T+1}^* + \hat{\beta}_{T+1}^* + \hat{\eta}_{T+2}^* + \hat{\beta}_{T+2}^* + \hat{\xi}_{T+2}^* + \hat{\eta}_{T+3}^* - \hat{\gamma}_{T+2}^* - \hat{\gamma}_{T+1}^* + \hat{\omega}_{T+3}^* + \hat{\varepsilon}_{T+3}^* \\
 \hat{y}_{T+3}^* &= \hat{\mu}_T^* + \hat{\beta}_T^* + \hat{\eta}_{T+1}^* + \hat{\beta}_{T+1}^* + \hat{\xi}_{T+1}^* + \hat{\eta}_{T+2}^* + \hat{\beta}_{T+2}^* + \hat{\xi}_{T+2}^* + \hat{\eta}_{T+3}^* - \hat{\gamma}_{T+2}^* - \hat{\gamma}_{T+1}^* + \hat{\omega}_{T+3}^* + \hat{\varepsilon}_{T+3}^* \\
 \hat{y}_{T+3}^* &= \hat{\mu}_T^* + 3\hat{\beta}_T^* + \hat{\eta}_{T+1}^* + \hat{\eta}_{T+2}^* + \hat{\eta}_{T+3}^* + 2\hat{\xi}_{T+1}^* + \hat{\xi}_{T+2}^* - \hat{\gamma}_{T+2}^* - \hat{\gamma}_{T+1}^* + \hat{\omega}_{T+3}^* + \hat{\varepsilon}_{T+3}^*
 \end{aligned}$$

**Fórmula Geral para grau de sazonalidade = 3:**

$$\hat{y}_{T+k}^* = \hat{\mu}_T^* + k\hat{\beta}_T^* + \sum_{j=1}^k [\hat{\eta}_{T+j}^* + (k-j)\hat{\xi}_{T+j}^*] - \hat{\gamma}_{T+k-1}^* - \hat{\gamma}_{T+k-2}^* + \hat{\omega}_{T+k}^* + \hat{\varepsilon}_{T+k}^*$$

### Apêndice 3 – Equações de Previsão *Bootstrap* Não-Paramétrico

$$\hat{a}_{T+k|T}^* = \hat{T}^{*k} \hat{a}_{T|T-1}^* + \sum_{j=0}^{k-1} \hat{T}^{*k-1-j} c + \sum_{j=0}^{k-1} \hat{T}^{*k-1-j} \hat{K}_{T+j}^* \hat{F}_{T+j}^{*-1} \hat{v}_{T+j}^*$$

$$\hat{y}_{T+k|T}^* = \hat{Z}^* \hat{T}^{*k} \hat{a}_{T|T-1}^* + \hat{Z}^* \sum_{j=0}^{k-1} \hat{T}^{*k-1-j} c + d + \hat{Z}^* \sum_{j=0}^{k-1} \hat{T}^{*k-1-j} \hat{K}_{T+j}^* \hat{F}_{T+j}^{*-1} \hat{v}_{T+j}^* + \hat{v}_{T+k}^*$$

Obs.: no nosso algoritmo o  $K_t$  já está sendo dividido pelo  $F_t$ ,  $c=0$  e  $d=0$ .

#### EQUAÇÕES DE PREVISÃO PARA O MNL

$$a_t = \mu_t \quad T_t = 1 \quad \text{e} \quad Z_t = 1$$

**Suponha  $k=1$ , temos:**

$$\hat{a}_{T+1|T}^* = \hat{T}^* \hat{a}_{T|T-1}^* + \hat{T}^{*0} \hat{K}_T^* \hat{v}_T^*$$

$$\hat{\mu}_{T+1|T}^* = \hat{\mu}_{T|T-1}^* + \hat{K}_T^* \hat{v}_T^*$$

$$\hat{y}_{T+1|T}^* = \hat{Z}^* \hat{T}^* \hat{a}_{T|T-1}^* + \hat{Z}^* (\hat{T}^{*0} \hat{K}_T^* \hat{v}_T^*) + \hat{v}_{T+1}^*$$

$$\hat{y}_{T+1|T}^* = \hat{\mu}_{T|T-1}^* + \hat{K}_T^* \hat{v}_T^* + \hat{v}_{T+1}^*$$

**Suponha  $k=2$ , temos:**

$$\hat{a}_{T+2|T}^* = \hat{T}^{*2} \hat{a}_{T|T-1}^* + \hat{T}^* \hat{K}_T^* \hat{v}_T^* + \hat{K}_{T+1}^* \hat{v}_{T+1}^*$$

$$\hat{\mu}_{T+2|T}^* = \hat{\mu}_{T|T-1}^* + \hat{K}_T^* \hat{v}_T^* + \hat{K}_{T+1}^* \hat{v}_{T+1}^*$$

$$\hat{y}_{T+2|T}^* = \hat{Z}^* \hat{T}^{*2} \hat{a}_{T|T-1}^* + \hat{Z}^* (\hat{T}^* \hat{K}_T^* \hat{v}_T^* + \hat{K}_{T+1}^* \hat{v}_{T+1}^*) + \hat{v}_{T+2}^*$$

$$\hat{y}_{T+2|T}^* = \hat{\mu}_{T|T-1}^* + \hat{K}_T^* \hat{v}_T^* + \hat{K}_{T+1}^* \hat{v}_{T+1}^* + \hat{v}_{T+2}^*$$

**Suponha  $k=3$ , temos:**

$$\hat{a}_{T+3|T}^* = \hat{T}^{*3} \hat{a}_{T|T-1}^* + \hat{T}^{*2} \hat{K}_T^* \hat{v}_T^* + \hat{T}^* \hat{K}_{T+1}^* \hat{v}_{T+1}^* + \hat{K}_{T+2}^* \hat{v}_{T+2}^*$$

$$\hat{\mu}_{T+3|T}^* = \hat{\mu}_{T|T-1}^* + \hat{K}_T^* \hat{v}_T^* + \hat{K}_{T+1}^* \hat{v}_{T+1}^* + \hat{K}_{T+2}^* \hat{v}_{T+2}^*$$

$$\hat{y}_{T+3|T}^* = \hat{Z}^* \hat{T}^{*3} \hat{a}_{T|T-1}^* + \hat{Z}^* (\hat{T}^{*2} \hat{K}_T^* \hat{v}_T^* + \hat{T}^* \hat{K}_{T+1}^* \hat{v}_{T+1}^* + \hat{K}_{T+2}^* \hat{v}_{T+2}^*) + \hat{v}_{T+3}^*$$

$$\hat{y}_{T+3|T}^* = \hat{\mu}_{T|T-1}^* + \hat{K}_T^* \hat{v}_T^* + \hat{K}_{T+1}^* \hat{v}_{T+1}^* + \hat{K}_{T+2}^* \hat{v}_{T+2}^* + \hat{v}_{T+3}^*$$

**Fórmula Geral:**

$$\hat{\mu}_{T+k|T}^* = \hat{\mu}_{T|T-1}^* + \sum_{j=0}^{k-1} \hat{K}_{T+j}^* \hat{v}_{T+j}^*$$

$$\hat{y}_{T+k|T}^* = \hat{\mu}_{T+k|T}^* + \hat{v}_{T+k}^*$$

Obs.: no algoritmo o  $\hat{K}_{T+j}^*$  foi aproximado pelo  $\hat{K}_T^*$  assim como o  $\hat{\mu}_{T+k}^*$ .

### EQUAÇÕES DE PREVISÃO PARA O MTL

$$a_t = \begin{bmatrix} \mu_t \\ \beta_t \end{bmatrix} \quad T_t = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \quad T_t^k = \begin{bmatrix} 1 & k \\ 0 & 1 \end{bmatrix} \quad K_t = \begin{bmatrix} K_t^{\mu} \\ K_t^{\beta} \end{bmatrix} \quad \text{e } Z_t = [1 \quad 0]$$

**Suponha k=1, temos:**

$$\hat{a}_{T+1|T}^* = \hat{T}^* \hat{a}_{T|T-1}^* + \hat{T}^{*0} \hat{K}_T^* \hat{v}_T^*$$

$$\begin{bmatrix} \hat{\mu}_{T+1|T}^* \\ \hat{\beta}_{T+1|T}^* \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{\mu}_{T|T-1}^* \\ \hat{\beta}_{T|T-1}^* \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{K}_T^{*\mu} \\ \hat{K}_T^{*\beta} \end{bmatrix} \hat{v}_T^*$$

$$\begin{bmatrix} \hat{\mu}_{T+1|T}^* \\ \hat{\beta}_{T+1|T}^* \end{bmatrix} = \begin{bmatrix} \hat{\mu}_{T|T-1}^* + \hat{\beta}_{T|T-1}^* \\ \hat{\beta}_{T|T-1}^* \end{bmatrix} + \begin{bmatrix} \hat{K}_T^{*\mu} \\ \hat{K}_T^{*\beta} \end{bmatrix} \hat{v}_T^*$$

$$\hat{\mu}_{T+1|T}^* = \hat{\mu}_{T|T-1}^* + \hat{\beta}_{T|T-1}^* + \hat{K}_T^{*\mu} \hat{v}_T^*$$

$$\hat{\beta}_{T+1|T}^* = \hat{\beta}_{T|T-1}^* + \hat{K}_T^{*\beta} \hat{v}_T^*$$

$$\hat{y}_{T+1|T}^* = \hat{Z}^* \hat{T}^* \hat{a}_{T|T-1}^* + \hat{Z}^* (\hat{T}^{*0} \hat{K}_T^* \hat{v}_T^*) + \hat{v}_{T+1}^*$$

$$\hat{y}_{T+1|T}^* = [1 \quad 0] \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{\mu}_{T|T-1}^* \\ \hat{\beta}_{T|T-1}^* \end{bmatrix} + [1 \quad 0] \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{K}_T^{*\mu} \\ \hat{K}_T^{*\beta} \end{bmatrix} \hat{v}_T^* + \hat{v}_{T+1}^*$$

$$\hat{y}_{T+1|T}^* = [1 \quad 0] \begin{bmatrix} \hat{\mu}_{T|T-1}^* + \hat{\beta}_{T|T-1}^* \\ \hat{\beta}_{T|T-1}^* \end{bmatrix} + [1 \quad 0] \begin{bmatrix} \hat{K}_T^{*\mu} \\ \hat{K}_T^{*\beta} \end{bmatrix} \hat{v}_T^* + \hat{v}_{T+1}^*$$

$$\hat{y}_{T+1|T}^* = \hat{\mu}_{T|T-1}^* + \hat{\beta}_{T|T-1}^* + \hat{K}_T^{*\mu} \hat{v}_T^* + \hat{v}_{T+1}^*$$

**Suponha k=2, temos:**

$$\hat{a}_{T+2|T}^* = \hat{T}^{*2} \hat{a}_{T|T-1}^* + \hat{T}^* \hat{K}_T^* \hat{v}_T^* + \hat{K}_{T+1}^* \hat{v}_{T+1}^*$$

$$\begin{bmatrix} \hat{\mu}_{T+2|T}^* \\ \hat{\beta}_{T+2|T}^* \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{\mu}_{T|T-1}^* \\ \hat{\beta}_{T|T-1}^* \end{bmatrix} + \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{K}_T^{*\mu} \\ \hat{K}_T^{*\beta} \end{bmatrix} \hat{v}_T^* + \begin{bmatrix} \hat{K}_{T+1}^{*\mu} \\ \hat{K}_{T+1}^{*\beta} \end{bmatrix} \hat{v}_{T+1}^*$$

$$\begin{bmatrix} \hat{\mu}_{T+2|T}^* \\ \hat{\beta}_{T+2|T}^* \end{bmatrix} = \begin{bmatrix} \hat{\mu}_{T|T-1}^* + 2\hat{\beta}_{T|T-1}^* \\ \hat{\beta}_{T|T-1}^* \end{bmatrix} + \begin{bmatrix} \hat{K}_T^{*\mu} + \hat{K}_T^{*\beta} \\ \hat{K}_T^{*\beta} \end{bmatrix} \hat{v}_T^* + \begin{bmatrix} \hat{K}_{T+1}^{*\mu} \\ \hat{K}_{T+1}^{*\beta} \end{bmatrix} \hat{v}_{T+1}^*$$

$$\hat{\mu}_{T+2|T}^* = \hat{\mu}_{T|T-1}^* + 2\hat{\beta}_{T|T-1}^* + (\hat{K}_T^{*\mu} + \hat{K}_T^{*\beta}) \hat{v}_T^* + \hat{K}_{T+1}^{*\mu} \hat{v}_{T+1}^*$$

$$\hat{\beta}_{T+2|T}^* = \hat{\beta}_{T|T-1}^* + \hat{K}_T^{*\beta} \hat{v}_T^* + \hat{K}_{T+1}^{*\beta} \hat{v}_{T+1}^*$$

$$\hat{y}_{T+2|T}^* = \hat{Z}^* \hat{T}^{*2} \hat{a}_{T|T-1}^* + \hat{Z}^* (\hat{T}^* \hat{K}_T^* \hat{v}_T^* + \hat{K}_{T+1}^* \hat{v}_{T+1}^*) + \hat{v}_{T+2}^*$$

$$\hat{y}_{T+2|T}^* = [1 \quad 0] \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{\mu}_{T|T-1}^* \\ \hat{\beta}_{T|T-1}^* \end{bmatrix} + [1 \quad 0] \left( \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{K}_T^{*\mu} \\ \hat{K}_T^{*\beta} \end{bmatrix} \hat{v}_T^* + \begin{bmatrix} \hat{K}_{T+1}^{*\mu} \\ \hat{K}_{T+1}^{*\beta} \end{bmatrix} \hat{v}_{T+1}^* \right) + \hat{v}_{T+2}^*$$

$$\hat{y}_{T+2|T}^* = [1 \quad 0] \begin{bmatrix} \hat{\mu}_{T|T-1}^* + 2\hat{\beta}_{T|T-1}^* \\ \hat{\beta}_{T|T-1}^* \end{bmatrix} + [1 \quad 0] \left( \begin{bmatrix} \hat{K}_T^{*\mu} + \hat{K}_T^{*\beta} \\ \hat{K}_T^{*\beta} \end{bmatrix} \hat{v}_T^* + \begin{bmatrix} \hat{K}_{T+1}^{*\mu} \\ \hat{K}_{T+1}^{*\beta} \end{bmatrix} \hat{v}_{T+1}^* \right) + \hat{v}_{T+2}^*$$

$$\hat{y}_{T+2|T}^* = \hat{\mu}_{T|T-1}^* + 2\hat{\beta}_{T|T-1}^* + (\hat{K}_T^{*\mu} + \hat{K}_T^{*\beta})\hat{v}_T^* + \hat{K}_{T+1}^{*\mu}\hat{v}_{T+1}^* + \hat{v}_{T+2}^*$$

**Suponha k=3, temos:**

$$\hat{a}_{T+3|T}^* = \hat{T}^{*3}\hat{a}_{T|T-1}^* + \hat{T}^{*2}\hat{K}_T^*\hat{v}_T^* + \hat{T}^*\hat{K}_{T+1}^*\hat{v}_{T+1}^* + \hat{K}_{T+2}^*\hat{v}_{T+2}^*$$

$$\begin{bmatrix} \hat{\mu}_{T+3|T}^* \\ \hat{\beta}_{T+3|T}^* \end{bmatrix} = \begin{bmatrix} 1 & 3 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{\mu}_{T|T-1}^* \\ \hat{\beta}_{T|T-1}^* \end{bmatrix} + \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{K}_T^{*\mu} \\ \hat{K}_T^{*\beta} \end{bmatrix} \hat{v}_T^* + \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{K}_{T+1}^{*\mu} \\ \hat{K}_{T+1}^{*\beta} \end{bmatrix} \hat{v}_{T+1}^* + \begin{bmatrix} \hat{K}_{T+2}^{*\mu} \\ \hat{K}_{T+2}^{*\beta} \end{bmatrix} \hat{v}_{T+2}^*$$

$$\begin{bmatrix} \hat{\mu}_{T+3|T}^* \\ \hat{\beta}_{T+3|T}^* \end{bmatrix} = \begin{bmatrix} \hat{\mu}_{T|T-1}^* + 3\hat{\beta}_{T|T-1}^* \\ \hat{\beta}_{T|T-1}^* \end{bmatrix} + \begin{bmatrix} \hat{K}_T^{*\mu} + 2\hat{K}_T^{*\beta} \\ \hat{K}_T^{*\beta} \end{bmatrix} \hat{v}_T^* + \begin{bmatrix} \hat{K}_{T+1}^{*\mu} + \hat{K}_{T+1}^{*\beta} \\ \hat{K}_{T+1}^{*\beta} \end{bmatrix} \hat{v}_{T+1}^* + \begin{bmatrix} \hat{K}_{T+2}^{*\mu} \\ \hat{K}_{T+2}^{*\beta} \end{bmatrix} \hat{v}_{T+2}^*$$

$$\hat{\mu}_{T+3|T}^* = \hat{\mu}_{T|T-1}^* + 3\hat{\beta}_{T|T-1}^* + (\hat{K}_T^{*\mu} + 2\hat{K}_T^{*\beta})\hat{v}_T^* + (\hat{K}_{T+1}^{*\mu} + \hat{K}_{T+1}^{*\beta})\hat{v}_{T+1}^* + \hat{K}_{T+2}^{*\mu}\hat{v}_{T+2}^*$$

$$\hat{\beta}_{T+3|T}^* = \hat{\beta}_{T|T-1}^* + \hat{K}_T^{*\beta}\hat{v}_T^* + \hat{K}_{T+1}^{*\beta}\hat{v}_{T+1}^* + \hat{K}_{T+2}^{*\beta}\hat{v}_{T+2}^*$$

$$\hat{y}_{T+3|T}^* = \hat{Z}^*\hat{T}^{*3}\hat{a}_{T|T-1}^* + \hat{Z}(\hat{T}^{*2}\hat{K}_T^*\hat{v}_T^* + \hat{T}^*\hat{K}_{T+1}^*\hat{v}_{T+1}^* + \hat{K}_{T+2}^*\hat{v}_{T+2}^*) + \hat{v}_{T+3}^*$$

$$\hat{y}_{T+3|T}^* = [1 \quad 0] \begin{bmatrix} 1 & 3 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{\mu}_{T|T-1}^* \\ \hat{\beta}_{T|T-1}^* \end{bmatrix} + [1 \quad 0] \left( \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{K}_T^{*\mu} \\ \hat{K}_T^{*\beta} \end{bmatrix} \hat{v}_T^* + \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{K}_{T+1}^{*\mu} \\ \hat{K}_{T+1}^{*\beta} \end{bmatrix} \hat{v}_{T+1}^* + \begin{bmatrix} \hat{K}_{T+2}^{*\mu} \\ \hat{K}_{T+2}^{*\beta} \end{bmatrix} \hat{v}_{T+2}^* \right) + \hat{v}_{T+3}^*$$

$$\hat{y}_{T+3|T}^* = [1 \quad 0] \begin{bmatrix} \hat{\mu}_{T|T-1}^* + 3\hat{\beta}_{T|T-1}^* \\ \hat{\beta}_{T|T-1}^* \end{bmatrix} + [1 \quad 0] \left( \begin{bmatrix} \hat{K}_T^{*\mu} + 2\hat{K}_T^{*\beta} \\ \hat{K}_T^{*\beta} \end{bmatrix} \hat{v}_T^* + \begin{bmatrix} \hat{K}_{T+1}^{*\mu} + \hat{K}_{T+1}^{*\beta} \\ \hat{K}_{T+1}^{*\beta} \end{bmatrix} \hat{v}_{T+1}^* + \begin{bmatrix} \hat{K}_{T+2}^{*\mu} \\ \hat{K}_{T+2}^{*\beta} \end{bmatrix} \hat{v}_{T+2}^* \right) + \hat{v}_{T+3}^*$$

$$\hat{y}_{T+3|T}^* = \hat{\mu}_{T|T-1}^* + 3\hat{\beta}_{T|T-1}^* + (\hat{K}_T^{*\mu} + 2\hat{K}_T^{*\beta})\hat{v}_T^* + (\hat{K}_{T+1}^{*\mu} + \hat{K}_{T+1}^{*\beta})\hat{v}_{T+1}^* + \hat{K}_{T+2}^{*\mu}\hat{v}_{T+2}^* + \hat{v}_{T+3}^*$$

**Fórmula Geral:**

$$\hat{\mu}_{T+k|T}^* = \hat{\mu}_{T|T-1}^* + k\hat{\beta}_{T|T-1}^* + \sum_{j=0}^{k-1} (\hat{K}_{T+j}^{*\mu} + (k-1-j)\hat{K}_{T+j}^{*\beta})\hat{v}_{T+j}^*$$

$$\hat{\beta}_{T+k|T}^* = \hat{\beta}_{T|T-1}^* + \sum_{j=0}^{k-1} \hat{K}_{T+j}^{*\beta}\hat{v}_{T+j}^*$$

$$\hat{y}_{T+k|T}^* = \hat{\mu}_{T+k|T}^* + \hat{v}_{T+k}^*$$

## EQUAÇÕES DE PREVISÃO PARA O MEB

**Suponha s=3, temos:**

$$a_t = \begin{bmatrix} \mu_t \\ \beta_t \\ \gamma_t \\ \gamma_{t-1} \end{bmatrix} \quad T_t = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad T_t^2 = \begin{bmatrix} 1 & 2 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & -1 \end{bmatrix} \quad T_t^3 = \begin{bmatrix} 1 & 3 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_t^4 = \begin{bmatrix} 1 & 4 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad K_t = \begin{bmatrix} K_t^\mu \\ K_t^\beta \\ K_t^{\gamma_t} \\ K_t^{\gamma_{t-1}} \end{bmatrix} \quad R_t = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad Z_t = [1 \quad 0 \quad 1 \quad 0]$$

**Suponha k=1, temos:**

$$\hat{a}_{T+1|T} = \hat{T}^* \hat{a}_{T|T-1} + \hat{T}^{*0} \hat{K}_T^* \hat{v}_T^*$$

$$\begin{bmatrix} \hat{\mu}_{T+1|T}^* \\ \hat{\beta}_{T+1|T}^* \\ \hat{\gamma}_{T+1|T}^* \\ \hat{\gamma}_{T|T}^* \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \hat{\mu}_{T|T-1}^* \\ \hat{\beta}_{T|T-1}^* \\ \hat{\gamma}_{T|T-1}^* \\ \hat{\gamma}_{T-1|T-1}^* \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} K_t^\mu \\ K_t^\beta \\ K_t^{\gamma_t} \\ K_t^{\gamma_{t-1}} \end{bmatrix} \hat{v}_T^*$$

$$\begin{bmatrix} \hat{\mu}_{T+1|T}^* \\ \hat{\beta}_{T+1|T}^* \\ \hat{\gamma}_{T+1|T}^* \\ \hat{\gamma}_{T|T}^* \end{bmatrix} = \begin{bmatrix} \hat{\mu}_{T|T-1}^* + \hat{\beta}_{T|T-1}^* \\ \hat{\beta}_{T|T-1}^* \\ -\hat{\gamma}_{T|T-1}^* - \hat{\gamma}_{T-1|T-1}^* \\ \hat{\gamma}_{T|T-1}^* \end{bmatrix} + \begin{bmatrix} K_t^\mu \\ K_t^\beta \\ K_t^{\gamma_t} \\ K_t^{\gamma_{t-1}} \end{bmatrix} \hat{v}_T^*$$

$$\hat{\mu}_{T+1|T}^* = \hat{\mu}_{T|T-1}^* + \hat{\beta}_{T|T-1}^* + K_T^\mu \hat{v}_T^*$$

$$\hat{\beta}_{T+1|T}^* = \hat{\beta}_{T|T-1}^* + K_T^\beta \hat{v}_T^*$$

$$\hat{\gamma}_{T+1|T}^* = -\hat{\gamma}_{T|T-1}^* - \hat{\gamma}_{T-1|T-1}^* + K_t^{\gamma_t} \hat{v}_T^*$$

$$\hat{\gamma}_{T|T}^* = \hat{\gamma}_{T|T-1}^* + K_t^{\gamma_{t-1}} \hat{v}_T^*$$

$$\hat{y}_{T+1|T}^* = \hat{Z}^* \hat{T}^* \hat{a}_{T|T-1}^* + \hat{Z}^* (\hat{T}^{*0} \hat{K}_T^* \hat{v}_T^*) + \hat{v}_{T+1}^*$$

$$\hat{y}_{T+1|T}^* = [1 \quad 0 \quad 1 \quad 0] \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \hat{\mu}_{T|T-1}^* \\ \hat{\beta}_{T|T-1}^* \\ \hat{\gamma}_{T|T-1}^* \\ \hat{\gamma}_{T-1|T-1}^* \end{bmatrix} + [1 \quad 0 \quad 1 \quad 0] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} K_t^\mu \\ K_t^\beta \\ K_t^{\gamma_t} \\ K_t^{\gamma_{t-1}} \end{bmatrix} \hat{v}_T^* + \hat{v}_{T+1}^*$$

$$\hat{y}_{T+1|T}^* = [1 \quad 1 \quad -1 \quad -1] \begin{bmatrix} \hat{\mu}_{T|T-1}^* \\ \hat{\beta}_{T|T-1}^* \\ \hat{\gamma}_{T|T-1}^* \\ \hat{\gamma}_{T-1|T-1}^* \end{bmatrix} + [1 \quad 0 \quad 1 \quad 0] \begin{bmatrix} K_t^\mu \\ K_t^\beta \\ K_t^{\gamma_t} \\ K_t^{\gamma_{t-1}} \end{bmatrix} \hat{v}_T^* + \hat{v}_{T+1}^*$$

$$\hat{y}_{T+1|T}^* = \hat{\mu}_{T|T-1}^* + \hat{\beta}_{T|T-1}^* - \hat{\gamma}_{T|T-1}^* - \hat{\gamma}_{T-1|T-1}^* + (K_T^\mu + K_t^{\gamma_t}) \hat{v}_T^* + \hat{v}_{T+1}^*$$

**Suponha k=2, temos:**

$$\hat{a}_{T+2|T} = \hat{T}^{*2} \hat{a}_{T|T-1} + \hat{T}^* \hat{K}_T^* \hat{v}_T^* + \hat{K}_{T+1}^* \hat{v}_{T+1}^*$$

$$\begin{bmatrix} \hat{\mu}_{T+2|T}^* \\ \hat{\beta}_{T+2|T}^* \\ \hat{\gamma}_{T+2|T}^* \\ \hat{\gamma}_{T+1|T}^* \end{bmatrix} = \begin{bmatrix} 1 & 2 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & -1 \end{bmatrix} \begin{bmatrix} \hat{\mu}_{T|T-1}^* \\ \hat{\beta}_{T|T-1}^* \\ \hat{\gamma}_{T|T-1}^* \\ \hat{\gamma}_{T-1|T-1}^* \end{bmatrix} + \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} K_t^\mu \\ K_t^\beta \\ K_t^{\gamma_t} \\ K_t^{\gamma_{t-1}} \end{bmatrix} \hat{v}_T^* + \begin{bmatrix} K_{t+1}^\mu \\ K_{t+1}^\beta \\ K_{t+1}^{\gamma_t} \\ K_{t+1}^{\gamma_{t-1}} \end{bmatrix} \hat{v}_{T+1}^*$$

$$\begin{bmatrix} \hat{\mu}_{T+2|T}^* \\ \hat{\beta}_{T+2|T}^* \\ \hat{\gamma}_{T+2|T}^* \\ \hat{\gamma}_{T+1|T}^* \end{bmatrix} = \begin{bmatrix} \hat{\mu}_{T|T-1}^* + 2\hat{\beta}_{T|T-1}^* \\ \hat{\beta}_{T|T-1}^* \\ \hat{\gamma}_{T-1|T-1}^* \\ -\hat{\gamma}_{T|T-1}^* - \hat{\gamma}_{T-1|T-1}^* \end{bmatrix} + \begin{bmatrix} K_t^\mu + K_t^\beta \\ K_t^\beta \\ -K_t^{\gamma_t} - K_t^{\gamma_{t-1}} \\ K_t^{\gamma_t} \end{bmatrix} \hat{v}_T^* + \begin{bmatrix} K_{t+1}^\mu \\ K_{t+1}^\beta \\ K_{t+1}^{\gamma_t} \\ K_{t+1}^{\gamma_{t-1}} \end{bmatrix} \hat{v}_{T+1}^*$$

$$\hat{\mu}_{T+2|T}^* = \hat{\mu}_{T|T-1}^* + 2\hat{\beta}_{T|T-1}^* + (K_T^\mu + K_T^\beta) \hat{v}_T^* + K_{T+1}^\mu \hat{v}_{T+1}^*$$

$$\hat{\beta}_{T+2|T}^* = \hat{\beta}_{T|T-1}^* + K_T^\beta \hat{v}_T^* + K_{T+1}^\beta \hat{v}_{T+1}^*$$

$$\hat{\gamma}_{T+2|T}^* = \hat{\gamma}_{T-1|T-1}^* + (-K_t^{\gamma_t} - K_t^{\gamma_{t-1}}) \hat{v}_T^* + K_{t+1}^{\gamma_t} \hat{v}_{T+1}^*$$

$$\hat{\gamma}_{T+1|T}^* = -\hat{\gamma}_{T|T-1}^* - \hat{\gamma}_{T-1|T-1}^* + K_t^{\gamma_t} \hat{v}_T^* + K_{t+1}^{\gamma_{t-1}} \hat{v}_{T+1}^*$$

$$\hat{y}_{T+2|T}^* = \hat{Z}^* \hat{T}^{*2} \hat{a}_{T|T-1}^* + \hat{Z}^* (\hat{T}^* \hat{K}_T^* \hat{v}_T^* + \hat{K}_{T+1}^* \hat{v}_{T+1}^*) + \hat{v}_{T+2}^*$$

$$\begin{aligned} \hat{y}_{T+2|T}^* &= [1 \ 0 \ 1 \ 0] \begin{bmatrix} 1 & 2 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & -1 \end{bmatrix} \begin{bmatrix} \hat{\mu}_{T|T-1}^* \\ \hat{\beta}_{T|T-1}^* \\ \hat{\gamma}_{T|T-1}^* \\ \hat{\gamma}_{T-1|T-1}^* \end{bmatrix} + [1 \ 0 \ 1 \ 0] \left( \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} K_t^\mu \\ K_t^\beta \\ K_t^{\gamma_t} \\ K_t^{\gamma_{t-1}} \end{bmatrix} \hat{v}_T^* + \begin{bmatrix} K_{t+1}^\mu \\ K_{t+1}^\beta \\ K_{t+1}^{\gamma_t} \\ K_{t+1}^{\gamma_{t-1}} \end{bmatrix} \hat{v}_{T+1}^* \right) \\ &\quad + \hat{v}_{T+2}^* \\ \hat{y}_{T+2|T}^* &= [1 \ 2 \ 0 \ 1] \begin{bmatrix} \hat{\mu}_{T|T-1}^* \\ \hat{\beta}_{T|T-1}^* \\ \hat{\gamma}_{T|T-1}^* \\ \hat{\gamma}_{T-1|T-1}^* \end{bmatrix} + [1 \ 0 \ 1 \ 0] \left( \begin{bmatrix} K_t^\mu + K_t^\beta \\ K_t^\beta \\ -K_t^{\gamma_t} - K_t^{\gamma_{t-1}} \\ K_t^{\gamma_t} \end{bmatrix} \hat{v}_T^* + \begin{bmatrix} K_{t+1}^\mu \\ K_{t+1}^\beta \\ K_{t+1}^{\gamma_t} \\ K_{t+1}^{\gamma_{t-1}} \end{bmatrix} \hat{v}_{T+1}^* \right) + \hat{v}_{T+2}^* \end{aligned}$$

$$\hat{y}_{T+2|T}^* = \hat{\mu}_{T|T-1}^* + 2\hat{\beta}_{T|T-1}^* + \hat{\gamma}_{T-1|T-1}^* + (K_T^\mu + K_T^\beta - K_t^{\gamma_t} - K_t^{\gamma_{t-1}}) \hat{v}_T^* + (K_{T+1}^\mu + K_{t+1}^{\gamma_t}) \hat{v}_{T+1}^* + \hat{v}_{T+2}^*$$

**Suponha k=3, temos:**

$$\hat{a}_{T+3|T} = \hat{T}^{*3} \hat{a}_{T|T-1} + \hat{T}^{*2} \hat{K}_T^* \hat{v}_T^* + \hat{T}^* \hat{K}_{T+1}^* \hat{v}_{T+1}^* + \hat{K}_{T+2}^* \hat{v}_{T+2}^*$$

$$\begin{bmatrix} \hat{\mu}_{T+3|T}^* \\ \hat{\beta}_{T+3|T}^* \\ \hat{\gamma}_{T+3|T}^* \\ \hat{\gamma}_{T+2|T}^* \end{bmatrix} = \begin{bmatrix} 1 & 3 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{\mu}_{T|T-1}^* \\ \hat{\beta}_{T|T-1}^* \\ \hat{\gamma}_{T|T-1}^* \\ \hat{\gamma}_{T-1|T-1}^* \end{bmatrix} + \begin{bmatrix} 1 & 2 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & -1 \end{bmatrix} \begin{bmatrix} K_t^\mu \\ K_t^\beta \\ K_t^{\gamma_t} \\ K_t^{\gamma_{t-1}} \end{bmatrix} \hat{v}_T^* + \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} K_{t+1}^\mu \\ K_{t+1}^\beta \\ K_{t+1}^{\gamma_t} \\ K_{t+1}^{\gamma_{t-1}} \end{bmatrix} \hat{v}_{T+1}^* + \begin{bmatrix} K_{t+2}^\mu \\ K_{t+2}^\beta \\ K_{t+2}^{\gamma_t} \\ K_{t+2}^{\gamma_{t-1}} \end{bmatrix} \hat{v}_{T+2}^*$$

$$\begin{bmatrix} \hat{\mu}_{T+3|T}^* \\ \hat{\beta}_{T+3|T}^* \\ \hat{\gamma}_{T+3|T}^* \\ \hat{\gamma}_{T+2|T}^* \end{bmatrix} = \begin{bmatrix} \hat{\mu}_{T|T-1}^* + 3\hat{\beta}_{T|T-1}^* \\ \hat{\beta}_{T|T-1}^* \\ \hat{\gamma}_{T|T-1}^* \\ \hat{\gamma}_{T-1|T-1}^* \end{bmatrix} + \begin{bmatrix} K_t^\mu + 2K_t^\beta \\ K_t^\beta \\ K_t^{\gamma_{t-1}} \\ -K_t^{\gamma_t} - K_t^{\gamma_{t-1}} \end{bmatrix} \hat{v}_T^* + \begin{bmatrix} K_{t+1}^\mu + K_{t+1}^\beta \\ K_{t+1}^\beta \\ -K_{t+1}^{\gamma_t} - K_{t+1}^{\gamma_{t-1}} \\ K_{t+1}^{\gamma_t} \end{bmatrix} \hat{v}_{T+1}^* + \begin{bmatrix} K_{t+2}^\mu \\ K_{t+2}^\beta \\ K_{t+2}^{\gamma_t} \\ K_{t+2}^{\gamma_{t-1}} \end{bmatrix} \hat{v}_{T+2}^*$$

$$\hat{\mu}_{T+3|T}^* = \hat{\mu}_{T|T-1}^* + 3\hat{\beta}_{T|T-1}^* + (K_T^\mu + 2K_T^\beta) \hat{v}_T^* + (K_{T+1}^\mu + K_{T+1}^\beta) \hat{v}_{T+1}^* + K_{T+2}^\mu \hat{v}_{T+2}^*$$

$$\hat{\beta}_{T+3|T}^* = \hat{\beta}_{T|T-1}^* + K_T^\beta \hat{v}_T^* + K_{T+1}^\beta \hat{v}_{T+1}^* + K_{T+2}^\beta \hat{v}_{T+2}^*$$



$$\hat{Y}_{T+3|T}^* = \hat{Y}_{T|T-1}^* + K_t^{\gamma_{t-1}} \hat{v}_T^* + (-K_{t+1}^{\gamma_t} - K_{t+1}^{\gamma_{t-1}}) \hat{v}_{T+1}^* + K_{t+2}^{\gamma_t} \hat{v}_{T+2}^*$$

$$\hat{Y}_{T+2|T}^* = \hat{Y}_{T-1|T-1}^* + (-K_t^{\gamma_t} - K_t^{\gamma_{t-1}}) \hat{v}_T^* + K_{t+1}^{\gamma_t} \hat{v}_{T+1}^* + K_{t+2}^{\gamma_{t-1}} \hat{v}_{T+2}^*$$

$$\hat{Y}_{T+3|T}^* = \hat{Z}^* \hat{T}^{*3} \hat{a}_{T|T-1}^* + \hat{Z}^* (\hat{T}^{*2} \hat{K}_T^* \hat{v}_T^* + \hat{T}^* \hat{K}_{T+1}^* \hat{v}_{T+1}^* + \hat{K}_{T+2}^* \hat{v}_{T+2}^*) + \hat{v}_{T+3}^*$$

$$\begin{aligned} \hat{Y}_{T+3|T}^* &= [1 \quad 0 \quad 1 \quad 0] \begin{bmatrix} 1 & 3 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{\mu}_{T|T-1}^* \\ \hat{\beta}_{T|T-1}^* \\ \hat{\gamma}_{T|T-1}^* \\ \hat{\gamma}_{T-1|T-1}^* \end{bmatrix} \\ &+ [1 \quad 0 \quad 1 \quad 0] \left( \begin{bmatrix} 1 & 2 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & -1 \end{bmatrix} \begin{bmatrix} K_t^\mu \\ K_t^\beta \\ K_t^{\gamma_t} \\ K_t^{\gamma_{t-1}} \end{bmatrix} \hat{v}_T^* + \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} K_{t+1}^\mu \\ K_{t+1}^\beta \\ K_{t+1}^{\gamma_t} \\ K_{t+1}^{\gamma_{t-1}} \end{bmatrix} \hat{v}_{T+1}^* + \begin{bmatrix} K_{t+2}^\mu \\ K_{t+2}^\beta \\ K_{t+2}^{\gamma_t} \\ K_{t+2}^{\gamma_{t-1}} \end{bmatrix} \hat{v}_{T+2}^* \right) \\ &+ \hat{v}_{T+3}^* \\ \hat{Y}_{T+3|T}^* &= [1 \quad 3 \quad 1 \quad 0] \begin{bmatrix} \hat{\mu}_{T|T-1}^* \\ \hat{\beta}_{T|T-1}^* \\ \hat{\gamma}_{T|T-1}^* \\ \hat{\gamma}_{T-1|T-1}^* \end{bmatrix} + [1 \quad 0 \quad 1 \quad 0] \left( \begin{bmatrix} K_t^\mu + 2K_t^\beta \\ K_t^\beta \\ K_t^{\gamma_t} \\ -K_t^{\gamma_t} - K_t^{\gamma_{t-1}} \end{bmatrix} \hat{v}_T^* + \begin{bmatrix} K_{t+1}^\mu + K_{t+1}^\beta \\ K_{t+1}^\beta \\ -K_{t+1}^{\gamma_t} - K_{t+1}^{\gamma_{t-1}} \\ K_{t+1}^{\gamma_t} \end{bmatrix} \hat{v}_{T+1}^* + \begin{bmatrix} K_{t+2}^\mu \\ K_{t+2}^\beta \\ K_{t+2}^{\gamma_t} \\ K_{t+2}^{\gamma_{t-1}} \end{bmatrix} \hat{v}_{T+2}^* \right) + \hat{v}_{T+3}^* \end{aligned}$$

$$\begin{aligned} \hat{Y}_{T+3|T}^* &= \hat{\mu}_{T|T-1}^* + 3\hat{\beta}_{T|T-1}^* + \hat{\gamma}_{T|T-1}^* + (K_T^\mu + 2K_T^\beta + K_t^{\gamma_{t-1}}) \hat{v}_T^* + (K_{T+1}^\mu + K_{T+1}^\beta - K_{t+1}^{\gamma_t} - K_{t+1}^{\gamma_{t-1}}) \hat{v}_{T+1}^* + (K_{T+2}^\mu + K_{t+2}^{\gamma_t}) \hat{v}_{T+2}^* \\ &+ \hat{v}_{T+3}^* \end{aligned}$$

**Suponha k=4, temos:**

$$\hat{a}_{T+4|T}^* = \hat{T}^{*4} \hat{a}_{T|T-1}^* + \hat{T}^{*3} \hat{K}_T^* \hat{v}_T^* + \hat{T}^{*2} \hat{K}_{T+1}^* \hat{v}_{T+1}^* + \hat{T}^* \hat{K}_{T+2}^* \hat{v}_{T+2}^* + \hat{K}_{T+3}^* \hat{v}_{T+3}^*$$

$$\begin{bmatrix} \hat{\mu}_{T+4|T}^* \\ \hat{\beta}_{T+4|T}^* \\ \hat{\gamma}_{T+4|T}^* \\ \hat{\gamma}_{T+3|T}^* \end{bmatrix} = \begin{bmatrix} 1 & 4 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \hat{\mu}_{T|T-1}^* \\ \hat{\beta}_{T|T-1}^* \\ \hat{\gamma}_{T|T-1}^* \\ \hat{\gamma}_{T-1|T-1}^* \end{bmatrix} + \begin{bmatrix} 1 & 3 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} K_t^\mu \\ K_t^\beta \\ K_t^{\gamma_t} \\ K_t^{\gamma_{t-1}} \end{bmatrix} \hat{v}_T^* + \begin{bmatrix} 1 & 2 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & -1 \end{bmatrix} \begin{bmatrix} K_{t+1}^\mu \\ K_{t+1}^\beta \\ K_{t+1}^{\gamma_t} \\ K_{t+1}^{\gamma_{t-1}} \end{bmatrix} \hat{v}_{T+1}^*$$

$$+ \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} K_{t+2}^\mu \\ K_{t+2}^\beta \\ K_{t+2}^{\gamma_t} \\ K_{t+2}^{\gamma_{t-1}} \end{bmatrix} \hat{v}_{T+2}^* + \begin{bmatrix} K_{t+3}^\mu \\ K_{t+3}^\beta \\ K_{t+3}^{\gamma_t} \\ K_{t+3}^{\gamma_{t-1}} \end{bmatrix} \hat{v}_{T+3}^*$$

$$\begin{bmatrix} \hat{\mu}_{T+4|T}^* \\ \hat{\beta}_{T+4|T}^* \\ \hat{\gamma}_{T+4|T}^* \\ \hat{\gamma}_{T+3|T}^* \end{bmatrix} = \begin{bmatrix} \hat{\mu}_{T|T-1}^* + 4\hat{\beta}_{T|T-1}^* \\ \hat{\beta}_{T|T-1}^* \\ -\hat{\gamma}_{T|T-1}^* - \hat{\gamma}_{T-1|T-1}^* \\ \hat{\gamma}_{T|T-1}^* \end{bmatrix} + \begin{bmatrix} K_t^\mu + 3K_t^\beta \\ K_t^\beta \\ K_t^{\gamma_t} \\ K_t^{\gamma_{t-1}} \end{bmatrix} \hat{v}_T^* + \begin{bmatrix} K_{t+1}^\mu + 2K_{t+1}^\beta \\ K_{t+1}^\beta \\ K_{t+1}^{\gamma_{t-1}} \\ -K_{t+1}^{\gamma_t} - K_{t+1}^{\gamma_{t-1}} \end{bmatrix} \hat{v}_{T+1}^* + \begin{bmatrix} K_{t+2}^\mu + K_{t+2}^\beta \\ K_{t+2}^\beta \\ -K_{t+2}^{\gamma_t} - K_{t+2}^{\gamma_{t-1}} \\ K_{t+2}^{\gamma_t} \end{bmatrix} \hat{v}_{T+2}^* + \begin{bmatrix} K_{t+3}^\mu \\ K_{t+3}^\beta \\ K_{t+3}^{\gamma_t} \\ K_{t+3}^{\gamma_{t-1}} \end{bmatrix} \hat{v}_{T+3}^*$$

$$\hat{\mu}_{T+4|T}^* = \hat{\mu}_{T|T-1}^* + 4\hat{\beta}_{T|T-1}^* + (K_T^\mu + 3K_T^\beta) \hat{v}_T^* + (K_{T+1}^\mu + 2K_{T+1}^\beta) \hat{v}_{T+1}^* + (K_{T+2}^\mu + K_{T+2}^\beta) \hat{v}_{T+2}^* + K_{T+3}^\mu \hat{v}_{T+3}^*$$

$$\hat{\beta}_{T+4|T}^* = \hat{\beta}_{T|T-1}^* + K_T^\beta \hat{v}_T^* + K_{T+1}^\beta \hat{v}_{T+1}^* + K_{T+2}^\beta \hat{v}_{T+2}^* + K_{T+3}^\beta \hat{v}_{T+3}^*$$

$$\hat{\gamma}_{T+4|T}^* = -\hat{\gamma}_{T|T-1}^* - \hat{\gamma}_{T-1|T-1}^* + K_t^{\gamma_t} \hat{v}_T^* + K_{t+1}^{\gamma_{t-1}} \hat{v}_{T+1}^* + (-K_{t+2}^{\gamma_t} - K_{t+2}^{\gamma_{t-1}}) \hat{v}_{T+2}^* + K_{t+3}^{\gamma_t} \hat{v}_{T+3}^*$$

$$\hat{\gamma}_{T+3|T}^* = \hat{\gamma}_{T|T-1}^* + K_t^{\gamma_{t-1}} \hat{v}_T^* + (-K_{t+1}^{\gamma_t} - K_{t+1}^{\gamma_{t-1}}) \hat{v}_{T+1}^* + K_{t+2}^{\gamma_t} \hat{v}_{T+2}^* + K_{t+3}^{\gamma_{t-1}} \hat{v}_{T+3}^*$$

$$\hat{y}_{T+4|T}^* = Z^* \hat{T}^{*4} \hat{a}_{T|T-1}^* + Z(\hat{T}^{*3} \hat{R}_T^* \hat{v}_T^* + \hat{T}^{*2} \hat{R}_{T+1}^* \hat{v}_{T+1}^* + \hat{T}^* \hat{R}_{T+2}^* \hat{v}_{T+2}^* + \hat{R}_{T+3}^* \hat{v}_{T+3}^*) + \hat{v}_{T+4}^*$$

$$\begin{aligned} \hat{y}_{T+4|T}^* &= [1 \ 0 \ 1 \ 0] \begin{bmatrix} 1 & 4 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \hat{\mu}_{T|T-1}^* \\ \hat{\beta}_{T|T-1}^* \\ \hat{\gamma}_{T|T-1}^* \\ \hat{\gamma}_{T-1|T-1}^* \end{bmatrix} \\ &+ [1 \ 0 \ 1 \ 0] \left( \begin{bmatrix} 1 & 3 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} K_t^\mu \\ K_t^\beta \\ K_t^{\gamma_t} \\ K_t^{\gamma_{t-1}} \end{bmatrix} \hat{v}_T^* + \begin{bmatrix} 1 & 2 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & -1 \end{bmatrix} \begin{bmatrix} K_{t+1}^\mu \\ K_{t+1}^\beta \\ K_{t+1}^{\gamma_t} \\ K_{t+1}^{\gamma_{t-1}} \end{bmatrix} \hat{v}_{T+1}^* \right. \\ &\left. + \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} K_{t+2}^\mu \\ K_{t+2}^\beta \\ K_{t+2}^{\gamma_t} \\ K_{t+2}^{\gamma_{t-1}} \end{bmatrix} \hat{v}_{T+2}^* + \begin{bmatrix} K_{t+3}^\mu \\ K_{t+3}^\beta \\ K_{t+3}^{\gamma_t} \\ K_{t+3}^{\gamma_{t-1}} \end{bmatrix} \hat{v}_{T+3}^* \right) + \hat{v}_{T+4}^* \\ \hat{y}_{T+4|T}^* &= [1 \ 4 \ -1 \ -1] \begin{bmatrix} \hat{\mu}_{T|T-1}^* \\ \hat{\beta}_{T|T-1}^* \\ \hat{\gamma}_{T|T-1}^* \\ \hat{\gamma}_{T-1|T-1}^* \end{bmatrix} \\ &+ [1 \ 0 \ 1 \ 0] \left( \begin{bmatrix} K_t^\mu + 3K_t^\beta \\ K_t^\beta \\ K_t^{\gamma_t} \\ K_t^{\gamma_{t-1}} \end{bmatrix} \hat{v}_T^* + \begin{bmatrix} K_{t+1}^\mu + 2K_{t+1}^\beta \\ K_{t+1}^\beta \\ K_{t+1}^{\gamma_t} \\ -K_{t+1}^{\gamma_t} - K_{t+1}^{\gamma_{t-1}} \end{bmatrix} \hat{v}_{T+1}^* + \begin{bmatrix} K_{t+2}^\mu + K_{t+2}^\beta \\ K_{t+2}^\beta \\ -K_{t+2}^{\gamma_t} - K_{t+2}^{\gamma_{t-1}} \\ K_{t+2}^{\gamma_t} \end{bmatrix} \hat{v}_{T+2}^* + \begin{bmatrix} K_{t+3}^\mu \\ K_{t+3}^\beta \\ K_{t+3}^{\gamma_t} \\ K_{t+3}^{\gamma_{t-1}} \end{bmatrix} \hat{v}_{T+3}^* \right) \\ &+ \hat{v}_{T+4}^* \end{aligned}$$

$$\begin{aligned} \hat{y}_{T+4|T}^* &= \hat{\mu}_{T|T-1}^* + 4\hat{\beta}_{T|T-1}^* - \hat{\gamma}_{T|T-1}^* - \hat{\gamma}_{T-1|T-1}^* + (K_T^\mu + 3K_T^\beta + K_T^{\gamma_t})\hat{v}_T^* + (K_{T+1}^\mu + 2K_{T+1}^\beta + K_{T+1}^{\gamma_{t-1}})\hat{v}_{T+1}^* \\ &+ (K_{T+2}^\mu + K_{T+2}^\beta - K_{T+2}^{\gamma_t} - K_{T+2}^{\gamma_{t-1}})\hat{v}_{T+2}^* + (K_{T+3}^\mu + K_{T+3}^{\gamma_t})\hat{v}_{T+3}^* + \hat{v}_{T+4}^* \end{aligned}$$

**Fórmula Geral para grau de sazonalidade = 3:**

$$\hat{\mu}_{T+k|T}^* = \hat{\mu}_{T|T-1}^* + k\hat{\beta}_{T|T-1}^* + \sum_{j=0}^{k-1} (\hat{R}_{T+j}^{*\mu} + (k-1-j)\hat{R}_{T+j}^{*\beta})\hat{v}_{T+j}^*$$

$$\hat{\beta}_{T+k|T}^* = \hat{\beta}_{T|T-1}^* + \sum_{j=0}^{k-1} \hat{R}_{T+j}^{*\beta} \hat{v}_{T+j}^*$$

Inicializa-se o gamma e depois aplica-se a fórmula geral para T+k:

$$\hat{\gamma}_{T+1|T}^* = -\hat{\gamma}_{T|T-1}^* - \hat{\gamma}_{T-1|T-1}^* + K_t^{\gamma_t} \hat{v}_T^*$$

$$\hat{\gamma}_{T|T}^* = \hat{\gamma}_{T|T-1}^* + K_t^{\gamma_{t-1}} \hat{v}_T^*$$

$$\hat{\gamma}_{T+k|T}^* = -\hat{\gamma}_{T+k-1|T}^* - \hat{\gamma}_{T+k-2|T}^* + K_{T+k-1}^{\gamma_t} \hat{v}_{T+k-1}^*$$

$$\hat{\gamma}_{T+k|T}^* = \hat{\mu}_{T+k|T}^* + \hat{\gamma}_{T+k|T}^* + \hat{v}_{T+k}^*$$

## Apêndice 4 – Equações de Previsão Assintótica

$F_{T+k T} = \hat{Z}\hat{T}^k\hat{P}_T(\hat{T}^k)' \hat{Z}' + \hat{Z} \sum_{j=0}^{k-1} \hat{T}^j RQR'(\hat{T}^j)' \hat{Z}' + H$
$\tilde{y}_{T+k T} = \hat{Z}\hat{T}^k\hat{a}_T + \hat{Z} \sum_{j=0}^{k-1} \hat{T}^j c + d$

Obs.: no nosso algoritmo o  $K_t$  já está sendo dividido pelo  $F_t$ ,  $c=0$  e  $d=0$ .

### EQUAÇÕES DE PREVISÃO PARA O MNL

$$a_t = \mu_t \quad T_t = 1 \quad Z_t = 1 \quad R_t = 1 \quad H_t = \sigma_\varepsilon^2 \quad Q_t = \sigma_\eta^2 \quad P_t = P_t^{\mu_t}$$

**Suponha  $k=1$ , temos:**

$$F_{T+1|T} = \hat{Z}\hat{T}\hat{P}_T\hat{T}'\hat{Z}' + \hat{Z}(RQR'\hat{Z}') + H$$

$$F_{T+1|T} = \hat{P}_T^{\mu_t} + \sigma_\eta^2 + \sigma_\varepsilon^2$$

$$\tilde{y}_{T+1|T} = \hat{Z}\hat{T}\hat{a}_T$$

$$\tilde{y}_{T+1|T} = \hat{\mu}_T$$

**Suponha  $k=2$ , temos:**

$$F_{T+2|T} = \hat{Z}\hat{T}^2\hat{P}_T(\hat{T}^2)' \hat{Z}' + \hat{Z}(RQR'\hat{Z}' + \hat{T}^1RQR'(\hat{T}^1)' \hat{Z}') + H$$

$$F_{T+2|T} = \hat{P}_T^{\mu_t} + (\sigma_\eta^2 + \sigma_\eta^2) + \sigma_\varepsilon^2$$

$$F_{T+2|T} = \hat{P}_T^{\mu_t} + 2\sigma_\eta^2 + \sigma_\varepsilon^2$$

$$\tilde{y}_{T+2|T} = \hat{Z}\hat{T}^2\hat{a}_T$$

$$\tilde{y}_{T+2|T} = \hat{\mu}_T$$

**Suponha  $k=3$ , temos:**

$$F_{T+3|T} = \hat{Z}\hat{T}^3\hat{P}_T(\hat{T}^3)' \hat{Z}' + \hat{Z}(\hat{T}^0RQR'(\hat{T}^0)' \hat{Z}' + \hat{T}^1RQR'(\hat{T}^1)' \hat{Z}' + \hat{T}^2RQR'(\hat{T}^2)' \hat{Z}') + H$$

$$F_{T+3|T} = \hat{P}_T^{\mu_t} + (\sigma_\eta^2 + \sigma_\eta^2 + \sigma_\eta^2) + \sigma_\varepsilon^2$$

$$F_{T+3|T} = \hat{P}_T^{\mu_t} + 3\sigma_\eta^2 + \sigma_\varepsilon^2$$

$$\tilde{y}_{T+3|T} = \hat{Z}\hat{T}^3\hat{a}_T$$

$$\tilde{y}_{T+3|T} = \hat{\mu}_T$$

**Fórmula geral:**

$$F_{T+k|T} = \hat{P}_T^{\mu_t} + k\sigma_\eta^2 + \sigma_\varepsilon^2$$

$$\tilde{y}_{T+k|T} = \hat{\mu}_T$$

**EQUAÇÕES DE PREVISÃO PARA O MTL**

$$a_t = \begin{bmatrix} \mu_t \\ \beta_t \end{bmatrix} \quad T_t = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \quad T_t^k = \begin{bmatrix} 1 & k \\ 0 & 1 \end{bmatrix} \quad K_t = \begin{bmatrix} K_t^\mu \\ K_t^\beta \end{bmatrix} \quad Z_t = [1 \quad 0] \quad R_t = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad H_t = \sigma_\varepsilon^2 \quad Q_t = \begin{bmatrix} \sigma_\eta^2 & 0 \\ 0 & \sigma_\xi^2 \end{bmatrix}$$

$$P_t = \begin{bmatrix} P_t^{\mu_t} & P_t^{\beta_{t-1}} \\ P_t^{\beta_{t-1}} & P_t^{\beta_t} \end{bmatrix}$$

**Suponha k=1, temos:**

$$F_{T+1|T} = \hat{Z}\hat{T}\hat{P}_T\hat{T}'\hat{Z}' + \hat{Z}(RQR'\hat{Z}') + H$$

$$F_{T+1|T} = [1 \quad 0] \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} P_T^{\mu_t} & P_T^{\beta_{t-1}} \\ P_T^{\beta_{t-1}} & P_T^{\beta_t} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + [1 \quad 0] \left( \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \sigma_\eta^2 & 0 \\ 0 & \sigma_\xi^2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right) + \sigma_\varepsilon^2$$

$$F_{T+1|T} = [1 \quad 1] \begin{bmatrix} P_T^{\mu_t} & P_T^{\beta_{t-1}} \\ P_T^{\beta_{t-1}} & P_T^{\beta_t} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} + [1 \quad 0] \left( \begin{bmatrix} \sigma_\eta^2 & 0 \\ 0 & \sigma_\xi^2 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right) + \sigma_\varepsilon^2$$

$$F_{T+1|T} = [P_T^{\mu_t} + P_T^{\beta_{t-1}} \quad P_T^{\beta_{t-1}} + P_T^{\beta_t}] \begin{bmatrix} 1 \\ 1 \end{bmatrix} + [1 \quad 0] \left( \begin{bmatrix} \sigma_\eta^2 \\ 0 \end{bmatrix} \right) + \sigma_\varepsilon^2$$

$$F_{T+1|T} = P_T^{\mu_t} + P_T^{\beta_t} + 2P_T^{\beta_{t-1}} + \sigma_\eta^2 + \sigma_\varepsilon^2$$

$$\tilde{y}_{T+1|T} = \hat{Z}\hat{T}\hat{a}_T$$

$$\tilde{y}_{T+1|T} = [1 \quad 0] \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{\mu}_T \\ \hat{\beta}_T \end{bmatrix}$$

$$\tilde{y}_{T+1|T} = [1 \quad 1] \begin{bmatrix} \hat{\mu}_T \\ \hat{\beta}_T \end{bmatrix}$$

$$\tilde{y}_{T+1|T} = \hat{\mu}_T + \hat{\beta}_T$$

**Suponha k=2, temos:**

$$F_{T+2|T} = \hat{Z}\hat{T}^2\hat{P}_T(\hat{T}^2)'\hat{Z}' + \hat{Z}(RQR'\hat{Z}' + \hat{T}^1RQR'(\hat{T}^1)'\hat{Z}') + H$$

$$F_{T+2|T} = [1 \quad 0] \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} P_T^{\mu_t} & P_T^{\beta_{t-1}} \\ P_T^{\beta_{t-1}} & P_T^{\beta_t} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$+ [1 \quad 0] \left( \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \sigma_\eta^2 & 0 \\ 0 & \sigma_\xi^2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \sigma_\eta^2 & 0 \\ 0 & \sigma_\xi^2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right) + \sigma_\varepsilon^2$$

$$F_{T+2|T} = [1 \quad 2] \begin{bmatrix} P_T^{\mu_t} & P_T^{\beta_{t-1}} \\ P_T^{\beta_{t-1}} & P_T^{\beta_t} \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} + [1 \quad 0] \left( \begin{bmatrix} \sigma_\eta^2 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \sigma_\eta^2 & 0 \\ 0 & \sigma_\xi^2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right) + \sigma_\varepsilon^2$$

$$F_{T+2|T} = [1 \quad 2] \begin{bmatrix} P_T^{\mu t} + 2P_T^{\beta t-1} \\ P_T^{\beta t-1} + 2P_T^{\beta t} \end{bmatrix} + [1 \quad 0] \left( \begin{bmatrix} \sigma_\eta^2 \\ 0 \end{bmatrix} + \begin{bmatrix} \sigma_\eta^2 & \sigma_\xi^2 \\ 0 & \sigma_\xi^2 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right) + \sigma_\varepsilon^2$$

$$F_{T+2|T} = P_T^{\mu t} + 4P_T^{\beta t} + 4P_T^{\beta t-1} + [1 \quad 0] \left( \begin{bmatrix} \sigma_\eta^2 \\ 0 \end{bmatrix} + \begin{bmatrix} \sigma_\eta^2 + \sigma_\xi^2 \\ \sigma_\xi^2 \end{bmatrix} \right) + \sigma_\varepsilon^2$$

$$F_{T+2|T} = P_T^{\mu t} + 4P_T^{\beta t} + 4P_T^{\beta t-1} + 2\sigma_\eta^2 + \sigma_\xi^2 + \sigma_\varepsilon^2$$

$$\tilde{y}_{T+2|T} = \hat{Z}\hat{T}^2\hat{a}_T$$

$$\tilde{y}_{T+2|T} = [1 \quad 0] \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{\mu}_T \\ \hat{\beta}_T \end{bmatrix}$$

$$\tilde{y}_{T+2|T} = [1 \quad 2] \begin{bmatrix} \hat{\mu}_T \\ \hat{\beta}_T \end{bmatrix}$$

$$\tilde{y}_{T+2|T} = \hat{\mu}_T + 2\hat{\beta}_T$$

**Suponha k=3, temos:**

$$F_{T+3|T} = \hat{Z}\hat{T}^3\hat{P}_T(\hat{T}^3)' \hat{Z}' + \hat{Z}(\hat{T}^0 RQR'(\hat{T}^0)' \hat{Z}' + \hat{T}^1 RQR'(\hat{T}^1)' \hat{Z}' + \hat{T}^2 RQR'(\hat{T}^2)' \hat{Z}') + H$$

$$F_{T+3|T} = [1 \quad 0] \begin{bmatrix} 1 & 3 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} P_T^{\mu t} & P_T^{\beta t-1} \\ P_T^{\beta t-1} & P_T^{\beta t} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 3 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ + [1 \quad 0] \left( \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \sigma_\eta^2 & 0 \\ 0 & \sigma_\xi^2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \sigma_\eta^2 & 0 \\ 0 & \sigma_\xi^2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right) \\ + [1 \quad 2] \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \sigma_\eta^2 & 0 \\ 0 & \sigma_\xi^2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right) + \sigma_\varepsilon^2$$

$$F_{T+3|T} = [1 \quad 3] \begin{bmatrix} P_T^{\mu t} & P_T^{\beta t-1} \\ P_T^{\beta t-1} & P_T^{\beta t} \end{bmatrix} \begin{bmatrix} 1 \\ 3 \end{bmatrix} + [1 \quad 0] \left( \begin{bmatrix} \sigma_\eta^2 \\ 0 \end{bmatrix} + \begin{bmatrix} \sigma_\eta^2 + \sigma_\xi^2 \\ \sigma_\xi^2 \end{bmatrix} + \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \sigma_\eta^2 & 0 \\ 0 & \sigma_\xi^2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right) + \sigma_\varepsilon^2$$

$$F_{T+3|T} = [1 \quad 3] \begin{bmatrix} P_T^{\mu t} + 3P_T^{\beta t-1} \\ P_T^{\beta t-1} + 3P_T^{\beta t} \end{bmatrix} + [1 \quad 0] \left( \begin{bmatrix} \sigma_\eta^2 \\ 0 \end{bmatrix} + \begin{bmatrix} \sigma_\eta^2 + \sigma_\xi^2 \\ \sigma_\xi^2 \end{bmatrix} + \begin{bmatrix} \sigma_\eta^2 & 2\sigma_\xi^2 \\ 0 & \sigma_\xi^2 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} \right) + \sigma_\varepsilon^2$$

$$F_{T+3|T} = P_T^{\mu t} + 9P_T^{\beta t} + 6P_T^{\beta t-1} + [1 \quad 0] \left( \begin{bmatrix} \sigma_\eta^2 \\ 0 \end{bmatrix} + \begin{bmatrix} \sigma_\eta^2 + \sigma_\xi^2 \\ \sigma_\xi^2 \end{bmatrix} + \begin{bmatrix} \sigma_\eta^2 + 4\sigma_\xi^2 \\ 2\sigma_\xi^2 \end{bmatrix} \right) + \sigma_\varepsilon^2$$

$$F_{T+3|T} = P_T^{\mu t} + 9P_T^{\beta t} + 6P_T^{\beta t-1} + [1 \quad 0] \left( \begin{bmatrix} 3\sigma_\eta^2 + 5\sigma_\xi^2 \\ 3\sigma_\xi^2 \end{bmatrix} \right) + \sigma_\varepsilon^2$$

$$F_{T+3|T} = P_T^{\mu t} + 9P_T^{\beta t} + 6P_T^{\beta t-1} + 3\sigma_\eta^2 + 5\sigma_\xi^2 + \sigma_\varepsilon^2$$

$$\tilde{y}_{T+3|T} = \hat{Z}\hat{T}^3\hat{a}_T$$

$$\tilde{y}_{T+3|T} = [1 \quad 0] \begin{bmatrix} 1 & 3 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{\mu}_T \\ \hat{\beta}_T \end{bmatrix}$$

$$\tilde{y}_{T+3|T} = [1 \quad 3] \begin{bmatrix} \hat{\mu}_T \\ \hat{\beta}_T \end{bmatrix}$$

$$\tilde{y}_{T+3|T} = \hat{\mu}_T + 3\hat{\beta}_T$$

**Fórmula Geral:**

$$F_{T+k|T} = P_T^{\mu_t} + k^2 P_T^{\beta_t} + 2k P_T^{\beta_{t-1}} + k \sigma_\eta^2 + \sum_{i=1}^k (i-1)^2 \sigma_\xi^2 + \sigma_\varepsilon^2$$

$$\tilde{y}_{T+k|T} = \hat{\mu}_T + k \hat{\beta}_T$$

### EQUAÇÕES DE PREVISÃO PARA O MEB

**Suponha s=3, temos:**

$$a_t = \begin{bmatrix} \mu_t \\ \beta_t \\ \gamma_t \\ \gamma_{t-1} \end{bmatrix} \quad T_t = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad T_t^2 = \begin{bmatrix} 1 & 2 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & -1 \end{bmatrix} \quad T_t^3 = \begin{bmatrix} 1 & 3 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_t^4 = \begin{bmatrix} 1 & 4 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad K_t = \begin{bmatrix} K_t^\mu \\ K_t^\beta \\ K_t^{\gamma_1} \\ K_t^{\gamma_2} \end{bmatrix} \quad R_t = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad Z_t = [1 \quad 0 \quad 1 \quad 0] \quad H_t = \sigma_\varepsilon^2$$

$$Q_t = \begin{bmatrix} \sigma_\eta^2 & 0 & 0 & 0 \\ 0 & \sigma_\xi^2 & 0 & 0 \\ 0 & 0 & \sigma_\omega^2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad P_t = \begin{bmatrix} P_t^{\mu_t} & P_t^{\beta_{t-1}} & 0 & 0 \\ P_t^{\beta_{t-1}} & P_t^{\beta_t} & 0 & 0 \\ 0 & 0 & P_t^{\gamma_t} & P_t^w \\ 0 & 0 & P_t^w & P_t^{\gamma_{t-1}} \end{bmatrix}$$

**Suponha k=1, temos:**

$$F_{T+1|T} = \hat{Z}' \hat{P}_T \hat{T}' \hat{Z}' + \hat{Z}' (R Q R' \hat{Z}') + H$$

$$F_{T+1|T} = [1 \quad 0 \quad 1 \quad 0] \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} P_T^{\mu_t} & P_T^{\beta_{t-1}} & 0 & 0 \\ P_T^{\beta_{t-1}} & P_T^{\beta_t} & 0 & 0 \\ 0 & 0 & P_T^{\gamma_t} & P_T^w \\ 0 & 0 & P_T^w & P_T^{\gamma_{t-1}} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \\ + [1 \quad 0 \quad 1 \quad 0] \left( \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \sigma_\eta^2 & 0 & 0 & 0 \\ 0 & \sigma_\xi^2 & 0 & 0 \\ 0 & 0 & \sigma_\omega^2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \right) + \sigma_\varepsilon^2$$

$$F_{T+1|T} = [1 \quad 1 \quad -1 \quad -1] \begin{bmatrix} P_T^\mu & P_T^{\beta_{t-1}} & 0 & 0 \\ P_T^{\beta_{t-1}} & P_T^\beta & 0 & 0 \\ 0 & 0 & P_T^\gamma & P_T^w \\ 0 & 0 & P_T^w & P_T^{\gamma_{t-1}} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} + [1 \quad 0 \quad 1 \quad 0] \begin{pmatrix} \begin{bmatrix} \sigma_\eta^2 & 0 & 0 & 0 \\ 0 & \sigma_\xi^2 & 0 & 0 \\ 0 & 0 & \sigma_\omega^2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \end{pmatrix} + \sigma_\varepsilon^2$$

$$F_{T+1|T} = [P_T^\mu + P_T^{\beta_{t-1}} \quad P_T^{\beta_{t-1}} + P_T^\beta \quad -P_T^\gamma - P_T^w \quad -P_T^w - P_T^{\gamma_{t-1}}] \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} + [1 \quad 0 \quad 1 \quad 0] \begin{pmatrix} \begin{bmatrix} \sigma_\eta^2 \\ 0 \\ \sigma_\omega^2 \\ 0 \end{bmatrix} \end{pmatrix} + \sigma_\varepsilon^2$$

$$F_{T+1|T} = P_T^\mu + P_T^\beta + 2P_T^{\beta_{t-1}} + P_T^\gamma + P_T^{\gamma_{t-1}} + 2P_T^w + \sigma_\eta^2 + \sigma_\omega^2 + \sigma_\varepsilon^2$$

$$\tilde{y}_{T+1|T} = \hat{Z}\hat{T}\hat{a}_T$$

$$\tilde{y}_{T+1|T} = [1 \quad 0 \quad 1 \quad 0] \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mu_t \\ \beta_t \\ \gamma_t \\ \gamma_{t-1} \end{bmatrix}$$

$$\tilde{y}_{T+1|T} = [1 \quad 1 \quad -1 \quad -1] \begin{bmatrix} \mu_t \\ \beta_t \\ \gamma_t \\ \gamma_{t-1} \end{bmatrix}$$

$$\tilde{y}_{T+1|T} = \hat{\mu}_t + \hat{\beta}_t - \hat{\gamma}_t - \hat{\gamma}_{t-1}$$

**Suponha k=2, temos:**

$$F_{T+2|T} = \hat{Z}\hat{T}^2\hat{P}_T(\hat{T}^2)'\hat{Z}' + \hat{Z}(RQR'\hat{Z}' + \hat{T}^1RQR'(\hat{T}^1)'\hat{Z}') + H$$

$$F_{T+2|T} = [1 \quad 0 \quad 1 \quad 0] \begin{bmatrix} 1 & 2 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & -1 \end{bmatrix} \begin{bmatrix} P_T^\mu & P_T^{\beta_{t-1}} & 0 & 0 \\ P_T^{\beta_{t-1}} & P_T^\beta & 0 & 0 \\ 0 & 0 & P_T^\gamma & P_T^w \\ 0 & 0 & P_T^w & P_T^{\gamma_{t-1}} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \\ + [1 \quad 0 \quad 1 \quad 0] \begin{pmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \sigma_\eta^2 & 0 & 0 & 0 \\ 0 & \sigma_\xi^2 & 0 & 0 \\ 0 & 0 & \sigma_\omega^2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \end{pmatrix} \\ + \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \sigma_\eta^2 & 0 & 0 & 0 \\ 0 & \sigma_\xi^2 & 0 & 0 \\ 0 & 0 & \sigma_\omega^2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \end{pmatrix} + \sigma_\varepsilon^2$$

$$\begin{aligned}
F_{T+2|T} &= [1 \ 2 \ 0 \ 1] \begin{bmatrix} P_T^{\mu_t} & P_T^{\beta_{t-1}} & 0 & 0 \\ P_T^{\beta_{t-1}} & P_T^{\beta_t} & 0 & 0 \\ 0 & 0 & P_T^{\gamma_t} & P_T^{\omega} \\ 0 & 0 & P_T^{\omega} & P_T^{\gamma_{t-1}} \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 0 \\ 1 \end{bmatrix} \\
&\quad + [1 \ 0 \ 1 \ 0] \left( \begin{bmatrix} \sigma_\eta^2 \\ 0 \\ \sigma_\omega^2 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \sigma_\eta^2 & 0 & 0 & 0 \\ 0 & \sigma_\xi^2 & 0 & 0 \\ 0 & 0 & \sigma_\omega^2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} \right) + \sigma_\varepsilon^2 \\
F_{T+2|T} &= [P_T^{\mu_t} + 2P_T^{\beta_{t-1}} \quad P_T^{\beta_{t-1}} + 2P_T^{\beta_t} \quad P_T^{\omega} \quad P_T^{\gamma_{t-1}}] \begin{bmatrix} 1 \\ 2 \\ 0 \\ 1 \end{bmatrix} + [1 \ 0 \ 1 \ 0] \left( \begin{bmatrix} \sigma_\eta^2 \\ 0 \\ \sigma_\omega^2 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \sigma_\eta^2 \\ \sigma_\xi^2 \\ -\sigma_\omega^2 \\ 0 \end{bmatrix} \right) + \sigma_\varepsilon^2 \\
F_{T+2|T} &= P_T^{\mu_t} + 2P_T^{\beta_{t-1}} + 2P_T^{\beta_{t-1}} + 4P_T^{\beta_t} + P_T^{\gamma_{t-1}} + [1 \ 0 \ 1 \ 0] \left( \begin{bmatrix} \sigma_\eta^2 \\ 0 \\ \sigma_\omega^2 \\ 0 \end{bmatrix} + \begin{bmatrix} \sigma_\eta^2 + \sigma_\xi^2 \\ \sigma_\xi^2 \\ \sigma_\omega^2 \\ -\sigma_\omega^2 \end{bmatrix} \right) + \sigma_\varepsilon^2 \\
F_{T+2|T} &= P_T^{\mu_t} + 4P_T^{\beta_t} + 4P_T^{\beta_{t-1}} + P_T^{\gamma_{t-1}} + [1 \ 0 \ 1 \ 0] \left( \begin{bmatrix} 2\sigma_\eta^2 + \sigma_\xi^2 \\ \sigma_\xi^2 \\ 2\sigma_\omega^2 \\ -\sigma_\omega^2 \end{bmatrix} \right) + \sigma_\varepsilon^2
\end{aligned}$$

$$F_{T+2|T} = P_T^{\mu_t} + 4P_T^{\beta_t} + 4P_T^{\beta_{t-1}} + P_T^{\gamma_{t-1}} + 2\sigma_\eta^2 + \sigma_\xi^2 + 2\sigma_\omega^2 + \sigma_\varepsilon^2$$

$$\tilde{y}_{T+2|T} = \hat{Z}\hat{T}^2\hat{a}_T$$

$$\tilde{y}_{T+2|T} = [1 \ 0 \ 1 \ 0] \begin{bmatrix} 1 & 2 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & -1 \end{bmatrix} \begin{bmatrix} \mu_t \\ \beta_t \\ \gamma_t \\ \gamma_{t-1} \end{bmatrix}$$

$$\tilde{y}_{T+2|T} = [1 \ 2 \ 0 \ 1] \begin{bmatrix} \mu_t \\ \beta_t \\ \gamma_t \\ \gamma_{t-1} \end{bmatrix}$$

$$\tilde{y}_{T+2|T} = \hat{\mu}_t + 2\hat{\beta}_t + \hat{\gamma}_{t-1}$$

**Suponha k=3, temos:**

$$F_{T+3|T} = \hat{Z}\hat{T}^3\hat{P}_T(\hat{T}^3)' \hat{Z}' + \hat{Z}(\hat{T}^0 RQR'(\hat{T}^0)' \hat{Z}' + \hat{T}^1 RQR'(\hat{T}^1)' \hat{Z}' + \hat{T}^2 RQR'(\hat{T}^2)' \hat{Z}') + H$$





$$\tilde{y}_{T+3|T} = \hat{Z}\hat{T}^3\hat{a}_T$$

$$\tilde{y}_{T+3|T} = [1 \ 0 \ 1 \ 0] \begin{bmatrix} 1 & 3 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mu_t \\ \beta_t \\ \gamma_t \\ \gamma_{t-1} \end{bmatrix}$$

$$\tilde{y}_{T+3|T} = [1 \ 3 \ 1 \ 0] \begin{bmatrix} \mu_t \\ \beta_t \\ \gamma_t \\ \gamma_{t-1} \end{bmatrix}$$

$$\tilde{y}_{T+3|T} = \hat{\mu}_t + 3\hat{\beta}_t + \hat{\gamma}_t$$

**Suponha k=4, temos:**

$$F_{T+4|T} = \hat{Z}\hat{T}^4\hat{P}_T(\hat{T}^4)'\hat{Z}' + \hat{Z}(\hat{T}^0RQR'(\hat{T}^0)'\hat{Z}' + \hat{T}^1RQR'(\hat{T}^1)'\hat{Z}' + \hat{T}^2RQR'(\hat{T}^2)'\hat{Z}' + \hat{T}^3RQR'(\hat{T}^3)'\hat{Z}') + H$$

$$F_{T+4|T} = [1 \ 0 \ 1 \ 0] \begin{bmatrix} 1 & 4 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} P_T^{\mu_t} & P_T^{\beta_{t-1}} & 0 & 0 \\ P_T^{\beta_{t-1}} & P_T^{\beta_t} & 0 & 0 \\ 0 & 0 & P_T^{\gamma_t} & P_T^w \\ 0 & 0 & P_T^w & P_T^{\gamma_{t-1}} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 4 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$+ [1 \ 0 \ 1 \ 0] \left( \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \sigma_\eta^2 & 0 \\ 0 & \sigma_\xi^2 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \right)$$

$$+ \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \sigma_\eta^2 & 0 \\ 0 & \sigma_\xi^2 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$+ \begin{bmatrix} 1 & 2 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \sigma_\eta^2 & 0 \\ 0 & \sigma_\xi^2 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$+ \begin{bmatrix} 1 & 3 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \sigma_\eta^2 & 0 \\ 0 & \sigma_\xi^2 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 3 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} \Big) + \sigma_\varepsilon^2$$

$$F_{T+4|T} = [1 \ 4 \ -1 \ -1] \begin{bmatrix} P_T^{\mu_t} & P_T^{\beta_{t-1}} & 0 & 0 \\ P_T^{\beta_{t-1}} & P_T^{\beta_t} & 0 & 0 \\ 0 & 0 & P_T^{\gamma_t} & P_T^w \\ 0 & 0 & P_T^w & P_T^{\gamma_{t-1}} \end{bmatrix} \begin{bmatrix} 1 \\ 4 \\ -1 \\ -1 \end{bmatrix}$$

$$+ [1 \ 0 \ 1 \ 0] \left( \begin{bmatrix} \sigma_\eta^2 \\ 0 \\ \sigma_\omega^2 \\ 0 \end{bmatrix} + \begin{bmatrix} \sigma_\eta^2 + \sigma_\xi^2 \\ \sigma_\xi^2 \\ \sigma_\omega^2 \\ -\sigma_\omega^2 \end{bmatrix} + \begin{bmatrix} \sigma_\eta^2 + 4\sigma_\xi^2 \\ 2\sigma_\xi^2 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 3 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \sigma_\eta^2 & 0 \\ 0 & \sigma_\xi^2 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 3 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 3 \\ 1 \\ 0 \end{bmatrix} \right) + \sigma_\varepsilon^2$$

$$\begin{aligned}
F_{T+4|T} &= \begin{bmatrix} P_T^\mu + 4P_T^{\beta_{t-1}} & P_T^{\beta_{t-1}} + 4P_T^{\beta_t} & -P_T^\gamma - P_T^w & -P_T^w - P_T^{\gamma_{t-1}} \end{bmatrix} \begin{bmatrix} 1 \\ 4 \\ -1 \\ -1 \end{bmatrix} \\
&+ [1 \ 0 \ 1 \ 0] \left( \begin{bmatrix} \sigma_\eta^2 \\ 0 \\ \sigma_\omega^2 \\ 0 \end{bmatrix} + \begin{bmatrix} \sigma_\eta^2 + \sigma_\xi^2 \\ \sigma_\xi^2 \\ \sigma_\omega^2 \\ -\sigma_\omega^2 \end{bmatrix} + \begin{bmatrix} \sigma_\eta^2 + 4\sigma_\xi^2 \\ 2\sigma_\xi^2 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 3 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \sigma_\eta^2 \\ 3\sigma_\xi^2 \\ \sigma_\omega^2 \\ 0 \end{bmatrix} \right) + \sigma_\varepsilon^2 \\
F_{T+4|T} &= P_T^\mu + 4P_T^{\beta_{t-1}} + 4P_T^{\beta_{t-1}} + 16P_T^{\beta_t} + P_T^\gamma + P_T^w + P_T^w + P_T^{\gamma_{t-1}} \\
&+ [1 \ 0 \ 1 \ 0] \left( \begin{bmatrix} \sigma_\eta^2 \\ 0 \\ \sigma_\omega^2 \\ 0 \end{bmatrix} + \begin{bmatrix} \sigma_\eta^2 + \sigma_\xi^2 \\ \sigma_\xi^2 \\ \sigma_\omega^2 \\ -\sigma_\omega^2 \end{bmatrix} + \begin{bmatrix} \sigma_\eta^2 + 4\sigma_\xi^2 \\ 2\sigma_\xi^2 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} \sigma_\eta^2 + 9\sigma_\xi^2 \\ 3\sigma_\xi^2 \\ \sigma_\omega^2 \\ 0 \end{bmatrix} \right) + \sigma_\varepsilon^2 \\
F_{T+4|T} &= P_T^\mu + 16P_T^{\beta_t} + 8P_T^{\beta_{t-1}} + P_T^\gamma + P_T^{\gamma_{t-1}} + 2P_T^w + [1 \ 0 \ 1 \ 0] \begin{bmatrix} 4\sigma_\eta^2 + 14\sigma_\xi^2 \\ 6\sigma_\xi^2 \\ 3\sigma_\omega^2 \\ -\sigma_\omega^2 \end{bmatrix} + \sigma_\varepsilon^2
\end{aligned}$$

$$F_{T+4|T} = P_T^\mu + 16P_T^{\beta_t} + 8P_T^{\beta_{t-1}} + P_T^\gamma + P_T^{\gamma_{t-1}} + 2P_T^w + 4\sigma_\eta^2 + 14\sigma_\xi^2 + 3\sigma_\omega^2 + \sigma_\varepsilon^2$$

$$\tilde{y}_{T+4|T} = \hat{Z}^4 \hat{a}_T$$

$$\tilde{y}_{T+4|T} = [1 \ 0 \ 1 \ 0] \begin{bmatrix} 1 & 4 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mu_t \\ \beta_t \\ \gamma_t \\ \gamma_{t-1} \end{bmatrix}$$

$$\tilde{y}_{T+4|T} = [1 \ 4 \ -1 \ -1] \begin{bmatrix} \mu_t \\ \beta_t \\ \gamma_t \\ \gamma_{t-1} \end{bmatrix}$$

$$\tilde{y}_{T+4|T} = \hat{\mu}_t + 4\hat{\beta}_t - \hat{\gamma}_t - \hat{\gamma}_{t-1}$$

### Fórmula Geral para grau de sazonalidade = 3:

Assuma que  $x$  seja um valor retornado da seguinte condição:

Para  $i=1$  até  $k$ , se o resto da divisão entre  $i$  e o grau de sazonalidade for diferente de 0,  $x = x + 1$ .

### Se o resto da divisão entre $k$ e o grau de sazonalidade for = 0

$$F_{T+k|T} = P_T^\mu + k^2 P_T^{\beta_t} + 2k P_T^{\beta_{t-1}} + P_T^\gamma + k\sigma_\eta^2 + \sum_{i=1}^k [(i-1)^2 \sigma_\xi^2] + x\sigma_\omega^2 + \sigma_\varepsilon^2$$

$$\tilde{y}_{T+k|T} = \hat{\mu}_t + k\hat{\beta}_t + \hat{\gamma}_t$$

**Se o resto da divisão entre k e o grau de sazonalidade for = 1**

$$F_{T+k|T} = P_T^{\mu_t} + k^2 P_T^{\beta_t} + 2k P_T^{\beta_{t-1}} + P_T^{\gamma_t} + P_T^{\gamma_{t-1}} + 2P_T^w + k\sigma_\eta^2 + \sum_{i=1}^k [(i-1)^2 \sigma_\xi^2] + x\sigma_\omega^2 + \sigma_\varepsilon^2$$

$$\tilde{y}_{T+k|T} = \hat{\mu}_t + k\hat{\beta}_t - \hat{\gamma}_t - \hat{\gamma}_{t-1}$$

**Se o resto da divisão entre k e o grau de sazonalidade for = 2**

$$F_{T+k|T} = P_T^{\mu_t} + k^2 P_T^{\beta_t} + 2k P_T^{\beta_{t-1}} + P_T^{\gamma_{t-1}} + k\sigma_\eta^2 + \sum_{i=1}^k [(i-1)^2 \sigma_\xi^2] + x\sigma_\omega^2 + \sigma_\varepsilon^2$$

$$\tilde{y}_{T+k|T} = \hat{\mu}_t + k\hat{\beta}_t + \hat{\gamma}_{t-1}$$

Apêndice 5 – Resultados Rodriguez e Ruiz (2009)

TABLE I

MONTE CARLO AVERAGE COVERAGES, LENGTH AND PERCENTAGE OF OBSERVATIONS LEFT OUT ON THE RIGHT AND ON THE LEFT OF THE PREDICTION INTERVALS FOR  $y_{T+k}$  CONSTRUCTED USING ST, WALL AND STOFFER (WS) AND SSB WHEN  $\varepsilon_t$  IS  $\mathcal{N}(0, 1)$ ,  $\eta_t$  IS  $\mathcal{N}(0, q)$  AND THE NOMINAL COVERAGE IS 95%

	k	Mean coverage			Mean coverage in tails			Mean length		
		ST	WS	SSB	ST	WS	SSB	ST	WS	SSB
					Below/Above	Below/Above	Below/Above			
<i>T</i> = 50										
<i>q</i> = 0.1	1	0.927	0.935	0.936	0.036/0.037	0.030/0.035	0.031/0.033	4.530	4.597	4.774
	5	0.927	0.940	0.943	0.036/0.037	0.029/0.031	0.028/0.029	5.182	5.285	5.539
	15	0.915	0.928	0.940	0.042/0.042	0.035/0.037	0.030/0.031	6.460	6.633	7.052
<i>q</i> = 1	1	0.936	0.923	0.928	0.029/0.035	0.036/0.041	0.036/0.035	6.157	6.250	6.280
	5	0.927	0.921	0.938	0.035/0.039	0.037/0.042	0.032/0.031	9.722	9.718	10.274
	15	0.914	0.909	0.934	0.041/0.045	0.043/0.047	0.033/0.033	15.258	15.194	16.469
<i>q</i> = 2	1	0.938	0.930	0.930	0.032/0.029	0.036/0.034	0.036/0.034	7.424	7.56	7.433
	5	0.926	0.924	0.931	0.037/0.036	0.038/0.038	0.034/0.034	12.849	12.880	13.088
	15	0.918	0.915	0.930	0.041/0.041	0.042/0.042	0.035/0.035	20.889	20.830	21.632
<i>T</i> = 100										
<i>q</i> = 0.1	1	0.945	0.941	0.943	0.025/0.030	0.031/0.028	0.026/0.031	4.569	4.576	4.618
	5	0.945	0.942	0.948	0.025/0.030	0.030/0.028	0.024/0.029	5.206	5.238	5.334
	15	0.938	0.938	0.945	0.029/0.033	0.032/0.030	0.026/0.030	6.498	6.575	6.743
<i>q</i> = 1	1	0.944	0.940	0.939	0.028/0.028	0.030/0.029	0.030/0.031	6.271	6.314	6.278
	5	0.939	0.937	0.942	0.031/0.030	0.032/0.031	0.029/0.029	9.874	9.873	10.120
	15	0.934	0.932	0.940	0.033/0.033	0.034/0.034	0.030/0.030	15.547	15.521	16.165
<i>q</i> = 2	1	0.945	0.937	0.939	0.028/0.027	0.032/0.030	0.031/0.030	7.476	7.537	7.460
	5	0.939	0.938	0.939	0.030/0.030	0.031/0.031	0.031/0.031	13.137	13.155	13.210
	15	0.935	0.935	0.937	0.032/0.032	0.032/0.033	0.031/0.031	21.509	21.539	21.758
<i>T</i> = 500										
<i>q</i> = 0.1	1	0.946	0.948	0.945	0.027/0.027	0.025/0.028	0.028/0.027	4.592	4.577	4.582
	5	0.946	0.947	0.946	0.026/0.028	0.025/0.028	0.027/0.027	5.217	5.206	5.223
	15	0.946	0.945	0.945	0.026/0.029	0.026/0.029	0.027/0.028	6.515	6.477	6.511
<i>q</i> = 1	1	0.948	0.948	0.947	0.029/0.023	0.027/0.025	0.027/0.025	6.339	6.335	6.314
	5	0.948	0.947	0.947	0.027/0.025	0.027/0.026	0.028/0.025	10.075	10.049	10.073
	15	0.947	0.946	0.947	0.027/0.026	0.027/0.027	0.027/0.026	15.956	15.919	15.944
<i>q</i> = 2	1	0.947	0.945	0.947	0.027/0.026	0.029/0.026	0.027/0.026	7.563	7.546	7.540
	5	0.948	0.948	0.948	0.027/0.027	0.027/0.025	0.027/0.027	13.418	13.446	13.387
	15	0.947	0.948	0.947	0.027/0.026	0.026/0.026	0.026/0.027	22.066	22.112	22.051

## Apêndice 6 – Algoritmo

### Funções

```
#include <oxstd.h>
#include <oxdraw.h>
#import <maximize>
#include <oxprob.h>
#include <oxfloat.h>
#include </packages/ssfpack/ssfpack.h>
////////////////////////////////////
// Declarando as variáveis globais
////////////////////////////////////
decl g_dEps1=1.0E-04;
decl g_dEps2=5.0E-03;
decl g_STSiBFGS=50;
decl g_STSiBurnIn=100;
decl g_STSiMessages=FALSE;
static decl g_STSmPhi, g_STSmOmega, g_STSmSigma, g_STSmKF;
static decl g_STSVt, Phi, Omega, Sigma;
////////////////////////////////////
// Simulando as séries temporais de acordo com o modelo
////////////////////////////////////
// Modelo de Nível Local
////////////////////////////////////
STSSimulateLLM(const dSigma2Eta, const dSigma2Eps, const iYtSize){
    decl iBurnIn=g_STSiBurnIn;
    decl vEta=sqrt(dSigma2Eta)*rann(1,iYtSize+iBurnIn);
    decl vEps=sqrt(dSigma2Eps)*rann(1,iYtSize+iBurnIn);
    decl vMu=zeros(1,iYtSize+iBurnIn);
    decl vYtAux=zeros(1,iYtSize+iBurnIn);
    decl vYt=zeros(1,iYtSize);
    decl i;
    vMu[0]=vEta[0];
    vYtAux[0]=vMu[0]+vEps[0];
    for (i=1;i<iYtSize+iBurnIn; i++){
        vMu[i]=vMu[i-1]+vEta[i];
        vYtAux[i]=vMu[i]+vEps[i];
    }
    for (i=0;i<iYtSize;i++){
        vYt[i]=vYtAux[i+iBurnIn];
    }
    return vYt;
}

////////////////////////////////////
// Modelo de Tendência Linear
////////////////////////////////////
STSSimulateLTL(const dSigma2Eta, const dSigma2Csi, const dSigma2Eps, const iYtSize){
    decl iBurnIn=g_STSiBurnIn;
    decl vEta=sqrt(dSigma2Eta)*rann(1,iYtSize+iBurnIn);
    decl vCsi=sqrt(dSigma2Csi)*rann(1,iYtSize+iBurnIn);
    decl vEps=sqrt(dSigma2Eps)*rann(1,iYtSize+iBurnIn);
    decl vMu=zeros(1,iYtSize+iBurnIn);
    decl vBeta=zeros(1,iYtSize+iBurnIn);
    decl vYtAux=zeros(1,iYtSize+iBurnIn);
    decl vYt=zeros(1,iYtSize);
    decl i;
    vMu[0]=vEta[0];
    vBeta[0]=vCsi[0];
    vYtAux[0]=vMu[0]+vEps[0];
    for (i=1;i<iYtSize+iBurnIn;i++){
        vMu[i]=vMu[i-1]+vBeta[i-1]+vEta[i];
        vBeta[i]=vBeta[i-1]+vCsi[i];
        vYtAux[i]=vMu[i]+vEps[i];
    }
}
```

```

    for (i=0;i<iYtSize;i++){
        vYt[i]=vYtAux[i+iBurnIn];
    }
    return vYt;
}

////////////////////////////////////
// Modelo Estrutural Básico
////////////////////////////////////
STSSimulateSBM(const dSigma2Eta, const dSigma2Csi,const dSigma2Eps, const dSigma2Omg,const iSazo, const
iYtSize){
    decl iBurnIn=g_STSiBurnIn+iSazo;
    decl vEta=sqrt(dSigma2Eta)*rann(1,iYtSize+iBurnIn);
    decl vCsi=sqrt(dSigma2Csi)*rann(1,iYtSize+iBurnIn);
    decl vOmg=sqrt(dSigma2Omg)*rann(1,iYtSize+iBurnIn);
    decl vEps=sqrt(dSigma2Eps)*rann(1,iYtSize+iBurnIn);
    decl vMu=zeros(1,iYtSize+iBurnIn);
    decl vBeta=zeros(1,iYtSize+iBurnIn);
    decl vGamma=zeros(1,iYtSize+iBurnIn);
    decl vYtAux=zeros(1,iYtSize+iBurnIn);
    decl vYt=zeros(1,iYtSize);
    decl i, j;
    vMu[0]=vEta[0];
    vBeta[0]=vCsi[0];
    vGamma[0]=vOmg[0];
    for (i=1;i<iYtSize+iBurnIn;i++){
        vMu[i]=vMu[i-1]+vBeta[i-1]+vEta[i];
        vBeta[i]=vBeta[i-1]+vCsi[i];
        vGamma[i]=vOmg[i];
        for (j=i-1;(j>=i-(iSazo-1))&&(j>=0);j--){
            vGamma[i]=vGamma[i]-vGamma[j];
        }
        vYtAux[i]=vMu[i]+vGamma[i]+vEps[i];
    }
    for (i=0;i<iYtSize;i++) {
        vYt[i]=vYtAux[i+iBurnIn];
    }
    return vYt;
}

////////////////////////////////////
// Definindo a função de Log-verossimilhança
////////////////////////////////////
STSLoglike(const vP, vFunc, const vScore, const mHess) {
    decl ii,dVar;
    ii=rows(g_STSmOmega);
    if (ii==2){
        g_STSmOmega[0][0]=exp(2.0*vP[0]);
        g_STSmOmega[1][1]=exp(2.0*vP[1]);
    } else if (ii==3){
        g_STSmOmega[0][0]=exp(2.0*vP[0]);
        g_STSmOmega[1][1]=exp(2.0*vP[1]);
        g_STSmOmega[2][2]=exp(2.0*vP[2]);
    } else{
        g_STSmOmega[0][0]=exp(2.0*vP[0]);
        g_STSmOmega[1][1]=exp(2.0*vP[1]);
        g_STSmOmega[2][2]=exp(2.0*vP[2]);
        g_STSmOmega[ii-1][ii-1]=exp(2.0*vP[3]);
    }
    ii=SsfLik(vFunc,&dVar,g_STsvYt,g_STSmPhi,g_STSmOmega,g_STSmSigma);
    return(ii);
}

```

```

////////////////////////////////////
// Ajustando o modelo
////////////////////////////////////
STSAjust(const vYt, const mSTSMModel){
    decl vP, vFunc, dVar;
    decl ii;
//    get time series
    g_STSVt=vYt;
//    initialize independent variables
    GetSsfStsm(mSTSMModel, &g_STSmPhi, &g_STSmOmega,&g_STSmSigma);
    ii=rows(g_STSmOmega);
    if (ii==2){
        vP=zeros(2,1);
        vP[0]=0.5*log(g_STSmOmega[0][0]);
        vP[1]=0.5*log(g_STSmOmega[1][1]);
    } else if (ii==3){
        vP=zeros(3,1);
        vP[0]=0.5*log(g_STSmOmega[0][0]);
        vP[1]=0.5*log(g_STSmOmega[1][1]);
        vP[2]=0.5*log(g_STSmOmega[2][2]);
    } else{
        vP=zeros(4,1);
        vP[0]=0.5*log(g_STSmOmega[0][0]);
        vP[1]=0.5*log(g_STSmOmega[1][1]);
        vP[2]=0.5*log(g_STSmOmega[2][2]);
        vP[3]=0.5*log(g_STSmOmega[ii-1][ii-1]);
    }
//    log-likelihood maximization
    MaxControl(g_STSiBFGS,-1);

    ii=MaxBFGS(STSLoglike,&vP,&vFunc,0,TRUE);
    if ((ii!=FALSE)&&(g_STSiMessages==TRUE)){
        println("STSAjust:ii=",ii," ",MaxConvergenceMsg(ii));
    }
    return g_STSmOmega;
}
////////////////////////////////////
// Série Bootstrap
////////////////////////////////////
STSBboot(const vYt,k) {
    decl mKF, mPhi, mOmega;
    decl iYtSize, il, iSazo, i, j;
    decl vVt, vKt, vFt, vEt, vRandU;
    decl vAtMu, vAtBeta, vAtGamma;
    decl num, numfinal,resboot,resbootprev, vYtBoot,vVt2,media;
    iYtSize=columns(vYt);
    resboot=zeros(1,iYtSize+k);
    vYtBoot=zeros(1,iYtSize);
//    Kalman filter estimation
    g_STSmKF=KalmanFil(vYt, g_STSmPhi, g_STSmOmega);
//    Seleccionando o modelo
    il=rows(g_STSmOmega);
    if (il==2) {
//        get Vt, Kt, Ft, and vEt
        vVt=g_STSmKF[0][:];
        vKt=g_STSmKF[1][:];
        vFt=ones(1,iYtSize)./g_STSmKF[2][:];
        vEt =(vVt)./(sqrt(vFt));
//        Bootstrapping residues
        vRandU=idiv((iYtSize-1)*ranu(iYtSize+k,1),1)+1;
        for(i=0;i<iYtSize+k;i++){
            resboot[i]=vEt[vRandU[i]];
        }
//        Creating bootstrap time series
        vAtMu=zeros(1,iYtSize+1);
        vAtMu[0]=0;

```



```

        for (i=0;i<iYtSize;i++) {
            vYtBoot[i]=vAtMu[i]+(sqrt(vFt[i])*resboot[i]);
            vAtMu[i+1]=vAtMu[i]+(vKt[i]*sqrt(vFt[i])*resboot[i]);
        }
    } else if (il==3) {
//      Get Vt, Kt, Ft, and vEt
        vVt=g_STSmKF[0][:];
        vKt=g_STSmKF[1:2][:];
        vFt=ones(1,iYtSize)./g_STSmKF[3][:];
        vEt = (vVt)./(sqrt(vFt));
//      Bootstrapping residues
        vRandU=idiv((iYtSize-2)*ranu(iYtSize+k,1),1)+2;
        for(i=0;i<iYtSize+k;i++){
            resboot[i]=vEt[vRandU[i]];
        }
//      Creating bootstrap time series
        vAtMu=zeros(1, iYtSize+1);
        vAtBeta=zeros(1, iYtSize+1);
        vAtMu[0]=0;
        vAtBeta[0]=0;
        for (i=0; i<iYtSize; i++) {
            vYtBoot[i]=vAtMu[i]+(sqrt(vFt[i])*resboot[i]);
            vAtMu[i+1]=vAtMu[i]+vAtBeta[i]+(vKt[0][i]*sqrt(vFt[i])*resboot[i]);
            vAtBeta[i+1]=vAtBeta[i]+(vKt[1][i]*sqrt(vFt[i])*resboot[i]);
        }
    } else if (il>3) {
//      Get Vt, Kt, Ft, and vEt
        iSazo=il-3;
        vVt=g_STSmKF[0][:];
        vKt=g_STSmKF[1:iSazo][:];
        vFt=ones(1,iYtSize)./g_STSmKF[iSazo+1][:];
        vEt = (vVt)./(sqrt(vFt));
//      Bootstrapping residues
        vRandU=idiv((iYtSize-il)*ranu(iYtSize+k,1),1)+il;
        for(i=0;i<iYtSize+k;i++){
            resboot[i]=vEt[vRandU[i]];
        }
//      creating bootstrap time series
        vAtMu=zeros(1, iYtSize+1);
        vAtBeta=zeros(1, iYtSize+1);
        vAtGamma=zeros(iSazo, iYtSize+1);
        vAtMu[0]=0;
        vAtBeta[0]=0;
        for (j=0; j<iSazo; j++) {
            vAtGamma[j][0]=0;
        }
        for (i=0; i<iYtSize; i++) {
            vYtBoot[i]=vAtMu[i]+vAtGamma[0][i]+sqrt(vFt[i])*resboot[i];
            vAtMu[i+1]=vAtMu[i]+vAtBeta[i]+vKt[0][i]*sqrt(vFt[i])*resboot[i];
            vAtBeta[i+1]=vAtBeta[i]+vKt[1][i]*sqrt(vFt[i])*resboot[i];
            vAtGamma[0][i+1]=vKt[2][i]*sqrt(vFt[i])*resboot[i];
            for (j=0; j<iSazo; j++) {
                vAtGamma[0][i+1]=vAtGamma[0][i+1]-vAtGamma[j][i];
            }
            for (j=1; j<iSazo; j++) {
                vAtGamma[j][i+1]=vAtGamma[j-1][i]+vKt[j+2][i]*sqrt(vFt[i])*resboot[i];
            }
        }
    }
}
return (vYtBoot);
}

```

```

////////////////////////////////////
// Previsão - Bootstrap      Não-paramétrico
////////////////////////////////////
STSTForecast(const vYt, const k) {
    decl mKF, mPhi, mOmega;
    decl iYtSize, il, iSazo, i, j;
    decl vVt, vKt, vFt, vEt;
    decl vAtGamma;
    decl soma, vYtPrev, mstate, mPred;;
    decl vRandU, resbootprev;
    iYtSize=columns(vYt);
    vYtPrev=zeros(k,1);
    resbootprev=zeros(1,iYtSize+k);
//
// Kalman filter estimation
g_STSmKF=KalmanFil(vYt, g_STSmPhi, g_STSmOmega);
//
// Selecionando o modelo
il=rows(g_STSmOmega);
if (il==2) {
//
// get Vt, Kt, Ft, and vEt
vVt=g_STSmKF[0][:];
vKt=g_STSmKF[1][:];
vFt=ones(1,iYtSize)./g_STSmKF[2][:];
//
// Construindo o vetor At
mstate=SsfMomentEst(1, &mPred, vYt, g_STSmPhi, g_STSmOmega);
//
// Bootstrapping residues
vRandU=idiv((iYtSize-1)*ranu(iYtSize+k,1),1)+1;
for(i=0;i<iYtSize+k;i++){
    resbootprev[i]= vVt[vRandU[i]];
}
//
// Previsão
for(i=0; i<k; i++){
    soma=0.0;
    for(j=0;j<=i;j++){
        soma = soma+(vKt[iYtSize-1]*resbootprev[iYtSize-1+j]);
    }
    vYtPrev[i]=mPred[0][iYtSize-1]+soma+resbootprev[iYtSize+i];
}
} else if (il==3) {
//
// get Vt, Kt, Ft, and vEt
vVt=g_STSmKF[0][:];
vKt=g_STSmKF[1:2][:];
vFt=ones(1,iYtSize)./g_STSmKF[3][:];
//
// Construindo o vetor At
mstate=SsfMomentEst(1, &mPred, vYt, g_STSmPhi, g_STSmOmega);
//
// Bootstrapping residues
vRandU=idiv((iYtSize-2)*ranu(iYtSize+k,1),1)+2;
for(i=0;i<iYtSize+k;i++){
    resbootprev[i]= vVt[vRandU[i]];
}
//
// Previsão
for(i=0; i<k; i++){
    soma=0.0;
    for(j=0;j<=i;j++){
        soma += (vKt[0][iYtSize-1]+(i-j)*vKt[1][iYtSize-1])*resbootprev[iYtSize-1+j];
    }
    vYtPrev[i]=mPred[0][iYtSize-1]+(i+1)*mPred[1][iYtSize-1]+soma+resbootprev[iYtSize+i];
}
} else if (il>3) {
//
// get Vt, Kt, Ft, and vEt
vVt=g_STSmKF[0][:];
vKt=g_STSmKF[1:i-1][:];
vFt=ones(1,iYtSize)./g_STSmKF[i][:];
//
// println("vKt",vKt[2][iYtSize-1]);
//
// Construindo o vetor At
mstate=SsfMomentEst(1, &mPred, vYt, g_STSmPhi, g_STSmOmega);
//
// println("mPred\t\n",mPred);

```

```

//      Bootstraping redidues
vRandU=idiv((iYtSize-i)*ranu(iYtSize+k,1),1)+i1;
for(i=0;i<iYtSize+k;i++){
    resbootprev[i]= vVt[vRandU[i]];
}
//      Previsão
vAtGamma=zeros(1,iYtSize+k);
for(i=2; i<iYtSize; i++){
    vAtGamma[i]=-mPred[2][i-1]-mPred[2][i-2]+(vKt[2][i-1]*resbootprev[i-1]);
}
for(i=0; i<k; i++){
    soma=0.0;
    for(j=0;j<=i;j++){
        soma += (vKt[0][iYtSize-1]+(i-j)*vKt[1][iYtSize-1])*resbootprev[iYtSize-1+j];
    }
    vAtGamma[iYtSize+i]=-vAtGamma[iYtSize-1+i]-vAtGamma[iYtSize-2+i]+
        +(vKt[2][iYtSize-1]*resbootprev[iYtSize-1+i]);
    vYtPrev[i]=mPred[0][iYtSize-1]+(i+1)*mPred[1][iYtSize-1]+soma+vAtGamma[iYtSize+i]+
        +resbootprev[iYtSize+i];
}
}
return (vYtPrev);
}
//////////////////////////////////////////////////////////////////
// Série Bootstrap Paramétrico MNL
//////////////////////////////////////////////////////////////////
STSTBootParMNL(const vYt, sig2eta, sig2eps, k) {
    decl mKF, mPhi, mOmega;
    decl i, j;
    decl vEt, vEt2;
    decl vAtMu;
    decl vYtBoot,vEps2,vEta2, media, desvio, media2, desvio2;
    decl iYtSize=columns(vYt);
    decl vEta=sqrt(sig2eta)*rann(1,iYtSize+k);
    decl vEps=sqrt(sig2eps)*rann(1,iYtSize+k);
//      Kalman filter estimation
g_STSmKF=KalmanFil(vYt, g_STSmPhi, g_STSmOmega);
//      Creating bootstrap time series
vAtMu=zeros(1,iYtSize+1);
vYtBoot=zeros(1,iYtSize);
vAtMu[0]=0;
for (i=0;i<iYtSize;i++) {
    vYtBoot[i]=vAtMu[i]+vEps[i];
    vAtMu[i+1]=vAtMu[i]+vEta[i];
}
return {vYtBoot, vEta, vEps};
}
//////////////////////////////////////////////////////////////////
// Série Bootstrap Paramétrico MTL
//////////////////////////////////////////////////////////////////
STSTBootParMTL(const vYt, sig2eta, sig2csi, sig2eps, k) {
    decl mKF, mPhi, mOmega;
    decl i, j;
    decl vEt;
    decl vAtMu, vAtBeta;
    decl vYtBoot,vEps2, media, desvio;
    decl iYtSize=columns(vYt);
    decl vEta=sqrt(sig2eta)*rann(1,iYtSize+k);
    decl vEps=sqrt(sig2eps)*rann(1,iYtSize+k);
    decl vCsi=sqrt(sig2csi)*rann(1,iYtSize+k);
//      Kalman filter estimation
g_STSmKF=KalmanFil(vYt, g_STSmPhi, g_STSmOmega);
//      Creating bootstrap time series
vAtMu=zeros(1, iYtSize+1);
vAtBeta=zeros(1, iYtSize+1);
vYtBoot=zeros(1, iYtSize);

```

```

vAtMu[0]=0;
vAtBeta[0]=0;
for (i=0;i<iYtSize;i++){
    vYtBoot[i]=vAtMu[i]+vEps[i];
    vAtMu[i+1]=vAtMu[i]+vAtBeta[i]+vEta[i];
    vAtBeta[i+1]=vAtBeta[i]+vCsi[i];
}
return {vYtBoot, vEta, vCsi, vEps};
}
////////////////////////////////////
// Série Bootstrap Paramétrico MEB
////////////////////////////////////
STSTBootParMEB(const vYt, sig2eta, sig2eps, sig2csi, sig2omg, k) {
    decl mKF, mPhi, mOmega;
    decl il, iSazo, i, j;
    decl vAtMu, vAtBeta, vAtGamma;
    decl vYtBoot;
    decl iYtSize=columns(vYt);
    decl vEta=sqrt(sig2eta)*rann(1,iYtSize+k);
    decl vEps=sqrt(sig2eps)*rann(1,iYtSize+k);
    decl vCsi=sqrt(sig2csi)*rann(1,iYtSize+k);
    decl vOmg=sqrt(sig2omg)*rann(1,iYtSize+k);
    il=rows(g_STSmOmega);
    iSazo=il-3;
// Kalman filter estimation
g_STSmKF=KalmanFil(vYt, g_STSmPhi, g_STSmOmega);
// creating bootstrap time series
vAtMu=zeros(1, iYtSize+1);
vAtBeta=zeros(1, iYtSize+1);
vYtBoot=zeros(1, iYtSize);
vAtMu[0]=0;
vAtBeta[0]=0;
vAtGamma=zeros(1,iYtSize+2);
vAtGamma[0]=0.0;
vAtGamma[1]=0.0;
for (i=0; i<iYtSize; i++) {
    vYtBoot[i]=vAtMu[i]+vAtGamma[i]+vEps[i];
    vAtMu[i+1]=vAtMu[i]+vAtBeta[i]+vEta[i];
    vAtBeta[i+1]=vAtBeta[i]+vCsi[i];
    vAtGamma[i+2]=-vAtGamma[i+1]-vAtGamma[i]+vOmg[i+2];
}
return {vYtBoot, vEta, vEps, vCsi, vOmg};
}
////////////////////////////////////
// Previsão - Bootstrap Paramétrico
////////////////////////////////////
STSTForecastPar(const vYt, const vEps, const vEta, const vCsi, const vOmg, const k) {
    decl iBurnIn=g_STSiBurnIn;
    decl mKF, mPhi, mOmega;
    decl iYtSize=columns(vYt);
    decl il, iSazo, i, j;
    decl vVt, vKt, vFt, vEt;
    decl vAtGamma;
    decl soma,soma1, vYtPrev, mstate, mPred;;
    decl vRandU, resbootprev, resbootprev2, vtp, media;
    vYtPrev=zeros(k,1);
// Kalman filter estimation
g_STSmKF=KalmanFil(vYt, g_STSmPhi, g_STSmOmega);
// Selecionando o modelo
il=rows(g_STSmOmega);
if (il==2) {
// mstate=SsfMomentEst(1, &mPred, vYt, g_STSmPhi, g_STSmOmega);
// Previsão
for(i=0; i<k; i++){
    soma=0.0;
    for(j=0;j<=i;j++){

```

```

        soma += vEta[iYtSize-1+j];
    }
    vYtPrev[i]=mPred[0][iYtSize-1]+soma+vEps[iYtSize+i];
}
} else if (il==3) {
    vKt=g_STSmKF[1:2][:];
    mstate=SsfMomentEst(1, &mPred, vYt, g_STSmPhi, g_STSmOmega);
// Previsão
    for(i=0; i<k; i++){
        soma=0.0;
        for(j=0;j<=i;j++){
            soma += vEta[iYtSize+j]+(i-j)*vCsi[iYtSize+j];
        }
        vYtPrev[i]=mPred[0][iYtSize-1]+(i+1)*mPred[1][iYtSize-1]+soma+vEps[iYtSize+i];
    }
} else if (il>3) {
    mstate=SsfMomentEst(1, &mPred, vYt, g_STSmPhi, g_STSmOmega);
// Previsão
    vAtGamma=zeros(1,iYtSize+k);
    for(i=2; i<iYtSize; i++){
        vAtGamma[i]=-mPred[2][i-1]-mPred[2][i-2]+vOmg[i];
    }
    for(i=0; i<k; i++){
        soma=0.0;
        for(j=0;j<=i;j++){
            soma += vEta[iYtSize+j]+(i-j)*vCsi[iYtSize+j];
        }
        vAtGamma[iYtSize+i]=-vAtGamma[iYtSize-1+i]-vAtGamma[iYtSize-2+i]+
            +vOmg[iYtSize+i];
        vYtPrev[i]=mPred[0][iYtSize-1]+(i+1)*mPred[1][iYtSize-1]+soma+vAtGamma[iYtSize+i]+
            +vEps[iYtSize+i];
    }
}
return (vYtPrev);
}
}
// Previsão - Assintótica
// Previsão - Assintótica
STSTForecastAs(const vYt, const k, const z) {
    decl iBurnIn=g_STSiBurnIn;
    decl mKF, mPhi, mOmega;
    decl iYtSize, il, iSazo, i, j,y,w,x;
    decl vVt, vKt, vFt, vPt,ptdt1;
    decl vAt,vAtMu, vAtBeta, vAtGamma;
    decl vYtPrev, vFtPrev, LimInf, LimSup;
    decl soma, mstate, mPred;
// Inicializando as variáveis
    iYtSize=columns(vYt);
    vYtPrev=zeros(k,1);
    vFtPrev=zeros(k,1);
    LimInf=zeros(k,1);
    LimSup=zeros(k,1);
    soma=0.0;
// Seleccionando o modelo
    il=rows(g_STSmOmega);
    if (il==2) {
        mstate=SsfMomentEst(1, &mPred, vYt, g_STSmPhi, g_STSmOmega);
        for(i=0; i<k; i++){
            vYtPrev[i]=mPred[0][iYtSize-1];
            vFtPrev[i]=mPred[2][iYtSize-1]+(i+1)*g_STSmOmega[0][0]+g_STSmOmega[1][1];
            LimInf[i]=vYtPrev[i]-(z*sqrt(vFtPrev[i]));
            LimSup[i]=vYtPrev[i]+(z*sqrt(vFtPrev[i]));
        }
    } else if (il==3) {
        mstate=SsfMomentEst(1, &mPred, vYt, g_STSmPhi, g_STSmOmega);
        for(i=0; i<k; i++){

```

```

vYtPrev[i]=mPred[0][iYtSize-1]+(i+1)*mPred[1][iYtSize-1];
soma=0.0;
for(j=0;j<=i;j++){
    soma += (j^2)*g_STSmOmega[1][1];
}
vFtPrev[i]=mPred[3][iYtSize-1]+((i+1)^2)*mPred[4][iYtSize-1]+
+2*(i+1)*mPred[4][iYtSize-2]+(i+1)*g_STSmOmega[0][0]+soma+g_STSmOmega[i-1][i-1];
LimInf[i]=vYtPrev[i]-(z*sqrt(vFtPrev[i]));
LimSup[i]=vYtPrev[i]+(z*sqrt(vFtPrev[i]));
}
} else if (il>3) {
    iSazo=il-3; //isazo=2
    mstate=SsfMomentEst(1, &mPred, vYt, g_STSmPhi, g_STSmOmega);
    w=0;
    for(i=0; i<iYtSize-1; i++){
        w=w+((-1)^(i+1))*mPred[iSazo+5][iYtSize-2-i];
    }
    x=0;
    for(i=0; i<k; i++){
        y=i%3; // y = resto da divisão entre i e 3
        if (y!=2){
            x=x+1;
        }
        if(y==0){
            vYtPrev[i]=mPred[0][iYtSize-1]+(i+1)*mPred[1][iYtSize-1]-
            -mPred[2][iYtSize-1]-mPred[2][iYtSize-2];
            soma=0.0;
            for(j=0;j<=i;j++){
                soma +=(j^2)*g_STSmOmega[1][1];
            }
            vFtPrev[i]=mPred[iSazo+3][iYtSize-1]+
            +((i+1)^2)*mPred[iSazo+4][iYtSize-1]+2*(i+1)*mPred[iSazo+4][iYtSize-2]+
            +mPred[iSazo+5][iYtSize-1]+mPred[iSazo+5][iYtSize-2]+
            +2*w+(i+1)*g_STSmOmega[0][0]+soma+(x*g_STSmOmega[2][2])+
            +g_STSmOmega[i-1][i-1];
        } if(y==1) {
            vYtPrev[i]=mPred[0][iYtSize-1]+(i+1)*mPred[1][iYtSize-1]+mPred[2][iYtSize-2];
            soma=0.0;
            for(j=0;j<=i;j++){
                soma +=(j^2)*g_STSmOmega[1][1];
            }
            vFtPrev[i]=mPred[iSazo+3][iYtSize-1]+
            +((i+1)^2)*mPred[iSazo+4][iYtSize-1]+2*(i+1)*mPred[iSazo+4][iYtSize-2]+
            +mPred[iSazo+5][iYtSize-2]+(i+1)*g_STSmOmega[0][0]+soma+
            +(x*g_STSmOmega[2][2])+g_STSmOmega[i-1][i-1];
        } if(y==2) {
            vYtPrev[i]=mPred[0][iYtSize-1]+(i+1)*mPred[1][iYtSize-1]+mPred[2][iYtSize-1];
            soma=0.0;
            for(j=0;j<=i;j++){
                soma +=(j^2)*g_STSmOmega[1][1];
            }
            vFtPrev[i]=mPred[iSazo+3][iYtSize-1]+
            +((i+1)^2)*mPred[iSazo+4][iYtSize-1]+2*(i+1)*mPred[iSazo+4][iYtSize-2]+
            +mPred[iSazo+5][iYtSize-1]+(i+1)*g_STSmOmega[0][0]+soma+
            +(x*g_STSmOmega[2][2])+g_STSmOmega[i-1][i-1];
        }
        LimInf[i]=vYtPrev[i]-(z*sqrt(vFtPrev[i]));
        LimSup[i]=vYtPrev[i]+(z*sqrt(vFtPrev[i]));
    }
}
return{vYtPrev,LimInf,LimSup};
}

```

## Main MNL

```
//////////////////////////////////////////////////////////////////
//                               Programa Principal - MNL
//////////////////////////////////////////////////////////////////
#include "funcoes.ox"

main(){
    decl L, B, n, k, m, alfa, z, sig2eta, sig2eps;
    decl somaepsbfgs, somaetabfgs, somaesteps, somaesteta, somaestepsPar, somaestetaPar;
    decl Amplitude, ICinf, ICsup, mediareal, mediarealAs, somarealdentro, somarealforainf, somarealforasup;
    decl ICfinalinf, ICfinalsup, Mediafinal, Amplitudefinal, finaldentro, finalforainf, finalforasup;
    decl AmplitudePar, ICinfPar, ICsupPar, mediarealPar, somarealdentroPar, somarealforainfPar;
    decl somarealforasupPar;
    decl ICfinalinfPar, ICfinalsupPar, MediafinalPar, AmplitudefinalPar, finaldentroPar, finalforainfPar;
    decl finalforasupPar;
    decl AmplitudeAs, LimInfAs, LimSupAs, somarealdentroAs, somarealforainfAs, somarealforasupAs;
    decl ICfinalinfAs, ICfinalsupAs, MediafinalAs, AmplitudefinalAs, finaldentroAs, finalforainfAs, finalforasupAs;
    decl l, Ytsim, Yt, x, Real, sigma2eta, sigma2eps, somaeta, somaeps, sigma2etaPar, sigma2epsPar, somaetaPar;
    decl somaepsPar;
    decl previsao, previsaoPar, previsaoAs, j, Ytb, resb, sigma2etaboot, sigma2epsboot, YtbPar, sigma2etabootPar;
    decl sigma2epsbootPar;
    decl Yprev, YprevPar, YprevAs, LInf, LSup, t, esteta, esteps, LInfPar, LSupPar, estetaPar, estepsPar;
    decl IC, inf, sup, liminf, limsup, ICP, infPar, supPar, liminfPar, limsupPar;
    decl estetabfgs, estepsbfgs, estepsfinal, estetafinal, estepsfinalPar, estetafinalPar, Eps, Eta;
// Modelo de Nível Local
decl STSmModelLLM=<
    CMP_LEVEL,1.0,0,0;
    CMP_IRREG,1.0,0,0>;

    ranseed(49870); // Inicializando a semente aleatória
    L=1000; // Número de séries a serem simuladas
    B=1000; // Número de réplicas Bootstrap
    n=500; // Tamanho da série a ser analisada
    k=12; // Número de previsões
    m=n+k; // Tamanho da série simulada (contém os k valores reais)
    alfa=0.05;
    z=1.96;

    sig2eps=0.5; // Inicializando os parâmetros da série simulada
    sig2eta=1.0;

    somaepsbfgs=0.0;
    somaetabfgs=0.0;

    somaesteps=0.0;
    somaesteta=0.0;

    somaestepsPar=0.0;
    somaestetaPar=0.0;

    STSmModelLLM[1][1]=sqrt(sig2eps); // Inicializando os parâmetros da série simulada MNL
    STSmModelLLM[0][1]=sqrt(sig2eta);

    Amplitude=zeros(L,k); // Intervalo Bootstrap não-paramétrico
    ICinf=zeros(L,k);
    ICsup=zeros(L,k);
    somarealdentro=zeros(L,k);
    somarealforainf=zeros(L,k);
    somarealforasup=zeros(L,k);
    ICfinalinf=zeros(1,k);
    ICfinalsup=zeros(1,k);
    mediareal=zeros(L,k);
    Mediafinal=zeros(1,k);
    Amplitudefinal=zeros(1,k);
    finaldentro=zeros(1,k);
}
```

```

finalforainf=zeros(1,k);
finalforasup=zeros(1,k);

AmplitudePar=zeros(L,k); // Intervalo Bootstrap paramétrico
ICinfPar=zeros(L,k);
ICsupPar=zeros(L,k);
somarealdentroPar=zeros(L,k);
somarealforainfPar=zeros(L,k);
somarealforasupPar=zeros(L,k);
ICfinalinfPar=zeros(1,k);
ICfinalsupPar=zeros(1,k);
mediarealPar=zeros(L,k);
MediafinalPar=zeros(1,k);
AmplitudefinalPar=zeros(1,k);
finaldentroPar=zeros(1,k);
finalforainfPar=zeros(1,k);
finalforasupPar=zeros(1,k);

previsaoAs=zeros(L,k); // Intervalo Assintótico
LimInfAs=zeros(L,k);
LimSupAs=zeros(L,k);
AmplitudeAs=zeros(L,k);
somarealdentroAs=zeros(L,k);
somarealforainfAs=zeros(L,k);
somarealforasupAs=zeros(L,k);
ICfinalinfAs=zeros(1,k);
ICfinalsupAs=zeros(1,k);
MediafinalAs=zeros(1,k);
AmplitudefinalAs=zeros(1,k);
finaldentroAs=zeros(1,k);
finalforainfAs=zeros(1,k);
finalforasupAs=zeros(1,k);

for(l=0;l<L;l++){

    println("\nPasso da simulação \t",l,"\n");
    Chamando a função que simula a série MNL
    Ytsim=STSSimulateLLM(sig2eta,sig2eps,m);
    Yt=zeros(1,n);
    for(x=0;x<n;x++){
        Yt[0,x]=Ytsim[0,x];
    }
    Real=zeros(1,k);
    for(x=0;x<k;x++){
        Real[0,x]=Ytsim[0,n+x];
    }
    g_STSmOmega=STSAdjust(Yt,STSmModelLLM);
    Estimando os hiperparâmetros do MNL
    sigma2eta=g_STSmOmega[0][0];
    sigma2eps=g_STSmOmega[1][1];
    somaetabfgs=sigma2eta+somaetabfgs;
    somaepsbfgs=sigma2eps+somaepsbfgs;
    Previsão Assintótica
    [YprevAs,LIInf,LSup]= STSForecastAs(Yt,k,z);
    for(t=0;t<k;t++){
        previsaoAs[l][t]=YprevAs[0,t];
        LimInfAs[l][t]=LIInf[0,t];
        LimSupAs[l][t]=LSup[0,t];
        AmplitudeAs[l][t]=LimSupAs[l][t]-LimInfAs[l][t];
    }
    somaeta=0.0;
    somaeps=0.0;
    somaetaPar=0.0;
    somaepsPar=0.0;
}

```



```

//      Previsão Bootstrap
      previsao=zeros(B,k);
      previsaoPar=zeros(B,k);
      for(j=0;j<B;j++){
          g_STSmOmega[0][0]=sigma2eta;
          g_STSmOmega[1][1]=sigma2eps;
//      Gerando a série Yt bootstrap não-paramétrico
          [Ytb]=STSBoot(Yt,k);
//      Estimando o sigma2epsboot e o sigma2etaboot pelo método BFGS para Ytboot não-paramétrico
          g_STSmOmega=STSAjust(Ytb,STSmModelLLM);
          sigma2etaboot=g_STSmOmega[0][0];
          sigma2epsboot=g_STSmOmega[1][1];
          somaeta=somaeta+sigma2etaboot;
          somaeps=somaeps+sigma2epsboot;
//      Chamando a funcao que calcula as previsões boot com a série Yt e os hiperparâmetros boot
//      não-paramétrico
          Yprev=STSForecast(Yt,k);
          for(t=0;t<k;t++){
              previsao[j][t]=Yprev[0,t];
          }
          g_STSmOmega[0][0]=sigma2eta;
          g_STSmOmega[1][1]=sigma2eps;
//      Gerando a série Yt bootstrap paramétrico
          [YtbPar,Eta,Eps]=STSBootParMNL(Yt,sigma2eta,sigma2eps,k);
//      Estimando o sigma2epsboot e o sigma2etaboot pelo método BFGS para Ytboot paramétrico
          g_STSmOmega=STSAjust(YtbPar,STSmModelLLM);
          sigma2etabootPar=g_STSmOmega[0][0];
          sigma2epsbootPar=g_STSmOmega[1][1];
          somaetaPar=somaetaPar+sigma2etabootPar;
          somaepsPar=somaepsPar+sigma2epsbootPar;
//      Chamando a funcao que calcula as previsões boot paramétricas com a série Yt e os
//      hiperparâmetros boot paramétrico
          YprevPar=STSForecastPar(Yt,Eps,Eta,0,0,k);
          for(t=0;t<k;t++){
              previsaoPar[j][t]=YprevPar[0,t];
          }
      }
      esteta=somaeta/B;
      esteps=somaeps/B;
      somaesteta=esteta+somaesteta;
      somaesteps=esteps+somaesteps;
      estetaPar=somaetaPar/B;
      estepsPar=somaepsPar/B;
      somaestetaPar=estetaPar+somaestetaPar;
      somaestepsPar=estepsPar+somaestepsPar;
      IC=zeros(B,k);
      IC=sortc(previsao);
      ICPar=zeros(B,k);
      ICPar=sortc(previsaoPar);
      inf=(alfa/2)*B;
      sup=B-((alfa/2)*B);
      liminf=round(inf)-1;
      limsup=round(sup)-1;
      for(t=0;t<k;t++){
          ICinf[1][t]=IC[liminf][t];
          ICsup[1][t]=IC[limsup][t];
          Amplitude[1][t]=ICsup[1][t]-ICinf[1][t];
          mediareal[1][t]=sumc(previsao[][t])/B;
          ICinfPar[1][t]=ICPar[liminf][t];
          ICsupPar[1][t]=ICPar[limsup][t];
          AmplitudePar[1][t]=ICsupPar[1][t]-ICinfPar[1][t];
          mediarealPar[1][t]=sumc(previsaoPar[][t])/B;
      }
      for(t=0;t<k;t++){
          if ((Real[0][t]>=ICinf[1][t]) && (Real[0][t]<=ICsup[1][t])){
              somarealdentro[1][t]=1;
          }
      }

```

```

    }
    if (Real[0][t]<ICinf[1][t]){
        somarealforainf[1][t]=1;
    }
    if (Real[0][t]>ICsup[1][t]){
        somarealforasup[1][t]=1;
    }
}
for(t=0;t<k;t++){
    if ((Real[0][t]>=ICinfPar[1][t]) && (Real[0][t]<=ICsupPar[1][t])){
        somarealdentroPar[1][t]=1;
    }
    if (Real[0][t]<ICinfPar[1][t]){
        somarealforainfPar[1][t]=1;
    }
    if (Real[0][t]>ICsupPar[1][t]){
        somarealforasupPar[1][t]=1;
    }
}
for(t=0;t<k;t++){
    if ((Real[0][t]>=LimInfAs[1][t]) && (Real[0][t]<=LimSupAs[1][t])){
        somarealdentroAs[1][t]=1;
    }
    if (Real[0][t]<LimInfAs[1][t]){
        somarealforainfAs[1][t]=1;
    }
    if (Real[0][t]>LimSupAs[1][t]){
        somarealforasupAs[1][t]=1;
    }
}
}
estetabfgs=somaetabfgs/L;
estepsbfgs=somaepsbfgs/L;
estetafinal=somaesteta/L;
estepsfinal=somaesteps/L;
estetafinalPar=somaestetaPar/L;
estepsfinalPar=somaestepsPar/L;

ICfinalinf=sumc(ICinf)/L;
ICfinalsup=sumc(ICsup)/L;
Mediafinal=sumc(mediareal)/L;
Amplitudefinal=sumc(Amplitude)/L;
finaldentro=sumc(somarealdentro);
finalforainf=sumc(somarealforainf);
finalforasup=sumc(somarealforasup);

ICfinalinfPar=sumc(ICinfPar)/L;
ICfinalsupPar=sumc(ICsupPar)/L;
MediafinalPar=sumc(mediarealPar)/L;
AmplitudefinalPar=sumc(AmplitudePar)/L;
finaldentroPar=sumc(somarealdentroPar);
finalforainfPar=sumc(somarealforainfPar);
finalforasupPar=sumc(somarealforasupPar);

ICfinalinfAs=sumc(LimInfAs)/L;
ICfinalsupAs=sumc(LimSupAs)/L;
MediafinalAs=sumc(previsaoAs)/L;
AmplitudefinalAs=sumc(AmplitudeAs)/L;
finaldentroAs=sumc(somarealdentroAs);
finalforainfAs=sumc(somarealforainfAs);
finalforasupAs=sumc(somarealforasupAs);

```

```

println("\t\n***** Hiperparametros reais para as series simuladas *****\t\n");
println("Parametro\ Estimativa");
println("Sigma2Eps\t",sig2eps,"\t");
println("Sigma2Eta\t",sig2eta,"\t");
println("\t\n***** Hiperparametros estimados-BFGS para as series simuladas *****\t\n");
println("Parametro\ Estimativa");
println("Sigma2Eps\t",estepsbfgs,"\t");
println("Sigma2Eta\t",estetabfgs,"\t");
println("\t\n**** Hiperparametros bootstrap Não-paramétrico estimados para as series simuladas **\t\n");
println("Parametro\ Estimativa");
println("Sigma2EpsBoot\t",estepsfinal,"\t");
println("Sigma2EtaBoot\t",estetafinal,"\t");
println("\t\n*****\t\n");
println("\t\n***** Hiperparametros bootstrap Paramétrico estimados para as series simuladas ***\t\n");
println("Parametro\ Estimativa");
println("Sigma2EpsBoot\t",estepsfinalPar,"\t");
println("Sigma2EtaBoot\t",estetafinalPar,"\t");
println("\t\n*****\t\n");

println("\t\n***** Intervalo de confianca Bootstrap Não-paramétrico*****\t\n");
println(" Passo 1 Passo 2 Passo 3 Passo 4 Passo 5 Passo 6\t");
println("Valor medio das previsoese",Mediafinal,"\t");
println("Lim Inf",ICfinalinf,"\t");
println("Lim Sup",ICfinalsup,"\t");
println("Amplitude do intervalo",Amplitudefinal,"\t");
println("Numero de vezes que o valor real ficou dentro do IC ",finaldentro,"\t");
println("Numero de vezes que o valor real ficou abaixo do limite inferior",finalforainf,"\t");
println("Numero de vezes que o valor real ficou acima do limite superior",finalforasup,"\t");
println("\t\n*****\t\n");

println("\t\n***** Intervalo de confianca Bootstrap Paramétrico*****\t\n");
println(" Passo 1 Passo 2 Passo 3 Passo 4 Passo 5 Passo 6\t");
println("Valor medio das previsoese",MediafinalPar,"\t");
println("Lim Inf",ICfinalinfPar,"\t");
println("Lim Sup",ICfinalsupPar,"\t");
println("Amplitude do intervalo",AmplitudefinalPar,"\t");
println("Numero de vezes que o valor real ficou dentro do IC ",finaldentroPar,"\t");
println("Numero de vezes que o valor real ficou abaixo do limite inferior",finalforainfPar,"\t");
println("Numero de vezes que o valor real ficou acima do limite superior",finalforasupPar,"\t");
println("\t\n*****\t\n");

println("\t\n***** Intervalo de confianca Assintotico *****\t\n");
println(" Passo 1 Passo 2 Passo 3 Passo 4 Passo 5 Passo 6\t");
println("Valor medio das previsoese",MediafinalAs,"\t");
println("Lim Inf",ICfinalinfAs,"\t");
println("Lim Sup",ICfinalsupAs,"\t");
println("Amplitude do intervalo",AmplitudefinalAs,"\t");
println("Numero de vezes que o valor real ficou dentro do IC",finaldentroAs,"\t");
println("Numero de vezes que o valor real ficou abaixo do limite inferior",finalforainfAs,"\t");
println("Numero de vezes que o valor real ficou acima do limite superior",finalforasupAs,"\t");
println("\t\n*****\t\n");
}

```

## Main MTL

```
////////////////////////////////////
//                               Programa Principal - MTL
////////////////////////////////////
#include "funcoes.ox"

main(){
    decl L, B, n, k, m, alfa, z, sig2eta, sig2eps, sig2csi;
    decl somaepsbfgs, somaetabfgs, somacsibfgs, somaesteps, somaesteta, somaestcsi;
    decl Amplitude, ICinf, ICsup, somarealdentro, somarealforainf, somarealforasup;
    decl ICfinalinf, ICfinalsup, Mediafinal, Amplitudefinal, finaldentro, finalforainf, finalforasup;
    decl AmplitudeAs, LimInfAs, LimSupAs, somarealdentroAs, somarealforainfAs, somarealforasupAs;
    decl ICfinalinfAs, ICfinalsupAs, MediafinalAs, AmplitudefinalAs, finaldentroAs, finalforainfAs, finalforasupAs;
    decl l, Ytsim, Yt, x, Real, sigma2eta, sigma2csi, sigma2eps, somaeta, somacsi, somaeps;
    decl previsao, previsaoAs, j, Ytb, resb, sigma2etaboot, sigma2csiboot, sigma2epsboot;
    decl Yprev, YprevAs, LInf, LSup, t, esteta, estcsi, esteps;
    decl IC, inf, sup, liminf, limsup;
    decl estetabfgs, estcsibfgs, estepsbfgs, estepsfinal, estcsifinal, estetafinal;
    decl mediareal, mediarealAs;
    decl somaestepsPar, somaestetaPar, somaestcsiPar, AmplitudePar, ICinfPar, ICsupPar;
    decl somarealdentroPar, somarealforainfPar, somarealforasupPar, ICfinalinfPar, ICfinalsupPar, mediarealPar;
    decl MediafinalPar, AmplitudefinalPar, finaldentroPar, finalforainfPar, finalforasupPar, Eps, Eta, Csi;
    decl somaepsPar, somaetaPar, somacsiPar, sigma2etabootPar, sigma2csibootPar, sigma2epsbootPar;
    decl estetaPar, estcsiPar, estepsPar, ICPAr, YtbPar, YprevPar;
    decl previsaoPar, estetafinalPar, estcsifinalPar, estepsfinalPar;
// Modelo de Tendência Linear
    decl STSmModelLTL=<
        CMP_LEVEL,1.0,0,0;
        CMP_SLOPE,1.0,0,0;
        CMP_IRREG,1.0,0,0>;

    ranseed(29660); // Inicializando a semente aleatória
    L=1000; // Número de séries a serem simuladas
    B=1000; // Número de réplicas Bootstrap
    n=500; // Tamanho da série a ser analisada
    k=12; // Número de previsões
    m=n+k; // Tamanho da série simulada (contém os k valores reais)
    alfa=0.05;
    z=1.96;

    sig2eta=0.5; // Inicilizando os parâmetros da série simulada
    sig2csi=0.01;
    sig2eps=1.0;

    somaepsbfgs=0.0;
    somaetabfgs=0.0;
    somacsibfgs=0.0;

    somaesteps=0.0;
    somaesteta=0.0;
    somaestcsi=0.0;

    somaestepsPar=0.0;
    somaestetaPar=0.0;
    somaestcsiPar=0.0;

    STSmModelLTL[0][1]=sqrt(sig2eta); // Inicilizando os parâmetros da série simulada MTL
    STSmModelLTL[1][1]=sqrt(sig2csi);
    STSmModelLTL[2][1]=sqrt(sig2eps);

    Amplitude=zeros(L,k); // Intervalo Bootstrap Não-paramétrico
    ICinf=zeros(L,k);
    ICsup=zeros(L,k);
    somarealdentro=zeros(L,k);
    somarealforainf=zeros(L,k);
}
```

```

somarealforasup=zeros(L,k);
ICfinalinf=zeros(1,k);
ICfinalsup=zeros(1,k);
mediareal=zeros(L,k);
Mediafinal=zeros(1,k);
Amplitudefinal=zeros(1,k);
finaldentro=zeros(1,k);
finalforainf=zeros(1,k);
finalforasup=zeros(1,k);

AmplitudePar=zeros(L,k); // Intervalo Bootstrap paramétrico
ICinfPar=zeros(L,k);
ICsupPar=zeros(L,k);
somarealdentroPar=zeros(L,k);
somarealforainfPar=zeros(L,k);
somarealforasupPar=zeros(L,k);
ICfinalinfPar=zeros(1,k);
ICfinalsupPar=zeros(1,k);
mediarealPar=zeros(L,k);
MediafinalPar=zeros(1,k);
AmplitudefinalPar=zeros(1,k);
finaldentroPar=zeros(1,k);
finalforainfPar=zeros(1,k);
finalforasupPar=zeros(1,k);

previsaoAs=zeros(L,k); // Intervalo Assintótico
LimInfAs=zeros(L,k);
LimSupAs=zeros(L,k);
AmplitudeAs=zeros(L,k);
somarealdentroAs=zeros(L,k);
somarealforainfAs=zeros(L,k);
somarealforasupAs=zeros(L,k);
ICfinalinfAs=zeros(1,k);
ICfinalsupAs=zeros(1,k);
MediafinalAs=zeros(1,k);
AmplitudefinalAs=zeros(1,k);
finaldentroAs=zeros(1,k);
finalforainfAs=zeros(1,k);
finalforasupAs=zeros(1,k);

for(l=0;l<L;l++){
    println("\nPasso da simulação \t",l,"\n");
// Chamando a função que simula a série MTL
    Ytsim=STSSimulateLTL(sig2eta,sig2csi,sig2eps,m);
    Yt=zeros(1,n);
    for(x=0;x<n;x++){
        Yt[0,x]=Ytsim[0,x];
    }
    Real=zeros(1,k);
    for(x=0;x<k;x++){
        Real[0,x]=Ytsim[0,n+x];
    }
// Estimando os hiperparâmetros do MTL pelo método BFGS
    g_STSmOmega=STSAjust(Yt,STSmModelLTL);
    sigma2eta=g_STSmOmega[0][0];
    sigma2csi=g_STSmOmega[1][1];
    sigma2eps=g_STSmOmega[2][2];
    somaetabfgs=sigma2eta+somaetabfgs;
    somaepsbfgs=sigma2eps+somaepsbfgs;
    somacsibfgs=sigma2csi+somacsibfgs;

// Previsão Assintótica
    [YprevAs,LInf,LSup]= STSForecastAs(Yt,k,z);
    for(t=0;t<k;t++){
        previsaoAs[l][t]=YprevAs[0,t];
    }
}

```

```

        LimInfAs[1][t]=LInf[0,t];
        LimSupAs[1][t]=LSup[0,t];
        AmplitudeAs[1][t]=LimSupAs[1][t]-LimInfAs[1][t];
    }
    somaeps=0.0;
    somaeta=0.0;
    somacsi=0.0;

    somaepsPar=0.0;
    somaetaPar=0.0;
    somacsiPar=0.0;

//    Previsão Bootstrap
    previsao=zeros(B,k);
    previsaoPar=zeros(B,k);
    for(j=0;j<B;j++){
        g_STSmOmega[0][0]=sigma2eta;
        g_STSmOmega[1][1]=sigma2csi;
        g_STSmOmega[2][2]=sigma2eps;
//        Gerando a série Yt bootstrap
        [Ytb]=STSBoot(Yt,k);
//        Estimando o sigma2epsboot e o sigma2etaboot pelo método BFGS para Ytboot
        g_STSmOmega=STSAdjust(Ytb,STSModelLTL);
        sigma2etaboot=g_STSmOmega[0][0];
        sigma2csiboot=g_STSmOmega[1][1];
        sigma2epsboot=g_STSmOmega[2][2];
        somaeta=somaeta+sigma2etaboot;
        somacsi=somacsi+sigma2csiboot;
        somaeps=somaeps+sigma2epsboot;
//        Chamando a funcao que calcula as previsões boot com a série Yt e os hiperparâmetros boot
        Yprev=STSTForecast(Yt,k);
        for(t=0;t<k;t++){
            previsao[j][t]=Yprev[0,t];
        }
        g_STSmOmega[0][0]=sigma2eta;
        g_STSmOmega[1][1]=sigma2csi;
        g_STSmOmega[2][2]=sigma2eps;
//        Gerando a série Yt bootstrap paramétrico
        [YtbPar,Eta, Csi, Eps]=STSBootParMTL(Yt,sigma2eta,sigma2csi,sigma2eps,k);
//        Estimando o sigma2epsboot e o sigma2etaboot pelo método BFGS para Ytboot paramétrico
        g_STSmOmega=STSAdjust(YtbPar,STSModelLTL);
        sigma2etabootPar=g_STSmOmega[0][0];
        sigma2csibootPar=g_STSmOmega[1][1];
        sigma2epsbootPar=g_STSmOmega[2][2];
        somaetaPar=somaetaPar+sigma2etabootPar;
        somaepsPar=somaepsPar+sigma2epsbootPar;
        somacsiPar=somacsiPar+sigma2csibootPar;

//        Chamando a funcao que calcula as previsões bootstrap paramétricas com a série Yt e os
//        hiperparâmetros bootstrap paramétrico
        YprevPar=STSTForecastPar(Yt,Eps, Eta, Csi,0,k);
        for(t=0;t<k;t++){
            previsaoPar[j][t]=YprevPar[0,t];
        }
    }
    esteta=somaeta/B;
    estcsi=somacsi/B;
    esteps=somaeps/B;
    somaesteta=esteta+somaesteta;
    somaestcsi=estcsi+somaestcsi;
    somaesteps=esteps+somaesteps;

    estetaPar=somaetaPar/B;
    estcsiPar=somacsiPar/B;
    estepsPar=somaepsPar/B;
    somaestetaPar=estetaPar+somaestetaPar;

```

```

somaestcsiPar=estcsiPar+somaestcsiPar;
somaestepsPar=estepsPar+somaestepsPar;

IC=zeros(B,k);
IC=sortc(previsao);
ICPar=zeros(B,k);
ICPar=sortc(previsaoPar);
inf=(alfa/2)*B;
sup=B-((alfa/2)*B);
liminf=round(inf)-1;
limsup=round(sup)-1;
for(t=0;t<k;t++){
    ICinf[1][t]=IC[liminf][t];
    ICsup[1][t]=IC[limsup][t];
    Amplitude[1][t]=ICsup[1][t]-ICinf[1][t];
    mediareal[1][t]=sumc(previsao[1][t])/B;
    ICinfPar[1][t]=ICPar[liminf][t];
    ICsupPar[1][t]=ICPar[limsup][t];
    AmplitudePar[1][t]=ICsupPar[1][t]-ICinfPar[1][t];
    mediarealPar[1][t]=sumc(previsaoPar[1][t])/B;
}
for(t=0;t<k;t++){
    if ((Real[0][t]>=ICinf[1][t]) && (Real[0][t]<=ICsup[1][t])){
        somarealdentro[1][t]=1;
    }
    if (Real[0][t]<ICinf[1][t]){
        somarealforainf[1][t]=1;
    }
    if (Real[0][t]>ICsup[1][t]){
        somarealforasup[1][t]=1;
    }
}
for(t=0;t<k;t++){
    if ((Real[0][t]>=ICinfPar[1][t]) && (Real[0][t]<=ICsupPar[1][t])){
        somarealdentroPar[1][t]=1;
    }
    if (Real[0][t]<ICinfPar[1][t]){
        somarealforainfPar[1][t]=1;
    }
    if (Real[0][t]>ICsupPar[1][t]){
        somarealforasupPar[1][t]=1;
    }
}
for(t=0;t<k;t++){
    if ((Real[0][t]>=LimInfAs[1][t]) && (Real[0][t]<=LimSupAs[1][t])){
        somarealdentroAs[1][t]=1;
    }
    if (Real[0][t]<LimInfAs[1][t]){
        somarealforainfAs[1][t]=1;
    }
    if (Real[0][t]>LimSupAs[1][t]){
        somarealforasupAs[1][t]=1;
    }
}
}
estetabfgs=somaetabfgs/L;
estcsibfgs=somacsibfgs/L;
estepsbfgs=somaepsbfgs/L;
estetafinal=somaesteta/L;
estcsifinal=somaestcsi/L;
estepsfinal=somaesteps/L;
estetafinalPar=somaestetaPar/L;
estcsifinalPar=somaestcsiPar/L;
estepsfinalPar=somaestepsPar/L;

ICfinalinf=sumc(ICinf)/L;

```

```

ICfinalsup=sumc(ICsup)/L;
Mediafinal=sumc(mediareal)/L;
Amplitudefinal=sumc(Amplitude)/L;
finaldentro=sumc(somarealdentro);
finalforainf=sumc(somarealforainf);
finalforasup=sumc(somarealforasup);

ICfinalinfPar=sumc(ICinfPar)/L;
ICfinalsupPar=sumc(ICsupPar)/L;
MediafinalPar=sumc(mediarealPar)/L;
AmplitudefinalPar=sumc(AmplitudePar)/L;
finaldentroPar=sumc(somarealdentroPar);
finalforainfPar=sumc(somarealforainfPar);
finalforasupPar=sumc(somarealforasupPar);

ICfinalinfAs=sumc(LimInfAs)/L;
ICfinalsupAs=sumc(LimSupAs)/L;
MediafinalAs=sumc(previsaoAs)/L;
AmplitudefinalAs=sumc(AmplitudeAs)/L;
finaldentroAs=sumc(somarealdentroAs);
finalforainfAs=sumc(somarealforainfAs);
finalforasupAs=sumc(somarealforasupAs);

println("\t\n***** Hiperparametros reais para as series simuladas *****\t\n");
println("Parametro\ Estimativa");
println("Sigma2Eta\t",sig2eta,"\t");
println("Sigma2Csi\t",sig2csi,"\t");
println("Sigma2Eps\t",sig2eps,"\t");

println("\t\n***** Hiperparametros estimados-BFGS para as series simuladas *****\t\n");
println("Parametro\ Estimativa");
println("Sigma2Eta\t",estetabfgs,"\t");
println("Sigma2Csi\t",estcsibfgs,"\t");
println("Sigma2Eps\t",estepsbfgs,"\t");

println("\t\n**** Hiperparametros bootstrap Não-paramétrico estimados para as series simuladas ****\t\n");
println("Parametro\ Estimativa");
println("Sigma2EtaBoot\t",estetafinal,"\t");
println("Sigma2CsiBoot\t",estcsifinal,"\t");
println("Sigma2EpsBoot\t",estepsfinal,"\t");

println("\t\n***** Hiperparametros bootstrap Paramétrico estimados para as series simuladas *****\t\n");
println("Parametro\ Estimativa");
println("Sigma2EtaBoot\t",estetafinalPar,"\t");
println("Sigma2CsiBoot\t",estcsifinalPar,"\t");
println("Sigma2EpsBoot\t",estepsfinalPar,"\t");
println("\t\n*****\t\n");

println("\t\n***** Intervalo de confianca Boostrap Não-paramétrico*****\t\n");
println(" Passo 1 Passo 2 Passo 3 Passo 4 Passo 5 Passo 6\t");
println("Valor medio das previsoes",Mediafinal,"\t");
println("Lim Inf",ICfinalinf,"\t");
println("Lim Sup",ICfinalsup,"\t");
println("Amplitude do intervalo",Amplitudefinal,"\t");
println("Numero de vezes que o valor real ficou dentro do IC ",finaldentro,"\t");
println("Numero de vezes que o valor real ficou abaixo do limite inferior",finalforainf,"\t");
println("Numero de vezes que o valor real ficou acima do limite superior",finalforasup,"\t");
println("\t\n*****\t\n");

println("\t\n***** Intervalo de confianca Boostrap Paramétrico*****\t\n");
println(" Passo 1 Passo 2 Passo 3 Passo 4 Passo 5 Passo 6\t");
println("Valor medio das previsoes",MediafinalPar,"\t");
println("Lim Inf",ICfinalinfPar,"\t");
println("Lim Sup",ICfinalsupPar,"\t");
println("Amplitude do intervalo",AmplitudefinalPar,"\t");
println("Numero de vezes que o valor real ficou dentro do IC ",finaldentroPar,"\t");

```



```

println("Numero de vezes que o valor real ficou abaixo do limite inferior",finalforainfPar,"\t");
println("Numero de vezes que o valor real ficou acima do limite superior",finalforasupPar,"\t");
println("\t\n*****\t\n");

println("\t\n***** Intervalo de confianca Assintotico *****\t\n");
println("    Passo 1    Passo 2    Passo 3    Passo 4    Passo 5    Passo 6\t");
println("Valor medio das previsoes",MediafinalAs,"\t");
println("Lim Inf",ICfinalinfAs,"\t");
println("Lim Sup",ICfinalsupAs,"\t");
println("Amplitude do intervalo",AmplitudefinalAs,"\t");
println("Numero de vezes que o valor real ficou dentro do IC",finaldentroAs,"\t");
println("Numero de vezes que o valor real ficou abaixo do limite inferior",finalforainfAs,"\t");
println("Numero de vezes que o valor real ficou acima do limite superior",finalforasupAs,"\t");
println("\t\n*****\t\n");
}

```

## Main MEB

```
//////////////////////////////////////////////////////////////////
//                               Programa Principal - MEB
//////////////////////////////////////////////////////////////////
#include "funcoes.ox"

main(){

    decl L, B, n, k, m, alfa, z, sig2eta, sig2csi, sig2omg, sig2eps, sazo;
    decl somaepsbfgs, somaetabfgs, somacsibfgs, somaomgbfgs, somaesteps, somaesteta, somaestcsi, somaestomg;
    decl Amplitude, ICinf, ICsup, somarealdentro, somarealforainf, somarealforasup, mediareal, mediarealAs;
    decl ICfinalinf, ICfinalsup, Mediafinal, Amplitudefinal, finaldentro, finalforainf, finalforasup;
    decl AmplitudeAs, LimInfAs, LimSupAs, somarealdentroAs, somarealforainfAs, somarealforasupAs;
    decl ICfinalinfAs, ICfinalsupAs, MediafinalAs, AmplitudefinalAs, finaldentroAs, finalforainfAs, finalforasupAs;
    decl l, Ytsim, Yt, x, Real, ii, sigma2eta, sigma2csi, sigma2omg, sigma2eps, somaeta, somacsi, somaomg,
somaeps;
    decl previsao, previsaoAs, j, Ytb, resb, sigma2etaboot, sigma2csiboot, sigma2omgboot, sigma2epsboot;
    decl Yprev, YprevAs, LInf, LSup, t, esteta, estcsi, estomg, esteps;
    decl IC, inf, sup, liminf, limsup;
    decl estetabfgs, estcsibfgs, estomgbfgs, estepsbfgs, estepsfinal, estcsifinal, estomgfinal, estetafinal;
    decl somaestepsPar, somaestetaPar, somaestcsiPar, somaestomgPar, AmplitudePar, ICinfPar, ICsupPar;
    decl somarealdentroPar, somarealforainfPar, somarealforasupPar, ICfinalinfPar, ICfinalsupPar, mediarealPar;
    decl MediafinalPar, AmplitudefinalPar, finaldentroPar, finalforainfPar, finalforasupPar, Eps, Eta, Csi, Omg;
    decl somaepsPar, somaetaPar, somacsiPar, somaomgPar, sigma2etabootPar, sigma2csibootPar, sigma2epsbootPar;
    decl sigma2omgbootPar;
    decl estetaPar, estcsiPar, estepsPar, estomgPar, ICPAr, YtbPar, YprevPar, previsaoPar, estetafinalPar, estcsifinalPar;
    decl estepsfinalPar, estomgfinalPar;

    decl STSmModelSBM=<                               // Modelo Estrutural Básico
        CMP_LEVEL,1.0,0,0;
        CMP_SLOPE,1.0,0,0;
        CMP_SEAS_DUMMY,1.0,12,0;
        CMP_IRREG,1.0,0,0>;

    ranseed(49870);                                     // Inicializando a semente aleatória
    L=1000;                                             // Número de séries a serem simuladas
    B=1000;                                             // Número de réplicas Bootstrap
    n=500;                                             // Tamanho da série a ser analisada
    k=12;                                              // Número de previsões
    m=n+k;                                             // Tamanho da série simulada (contém os k valores

reais)
    alfa=0.05;
    z=1.96;

    sig2eps=1.0;                                       // Inicilizando os parâmetros da série simulada
    sig2eta=0.5;
    sig2csi=0.01;
    sig2omg=0.1;
    sazo=3;

    somaepsbfgs=0.0;
    somaetabfgs=0.0;
    somacsibfgs=0.0;
    somaomgbfgs=0.0;

    somaesteps=0.0;
    somaesteta=0.0;
    somaestcsi=0.0;
    somaestomg=0.0;

    somaestepsPar=0.0;
    somaestetaPar=0.0;
    somaestcsiPar=0.0;
    somaestomgPar=0.0;

    STSmModelSBM[0][1]=sqrt(sig2eta);                 // Inicilizando os parâmetros da série simulada MEB
```

```

STSmModelSBM[1][1]=sqrt(sig2csi);
STSmModelSBM[2][1]=sqrt(sig2omg);
STSmModelSBM[2][2]=sazo;
STSmModelSBM[3][1]=sqrt(sig2eps);

Amplitude=zeros(L,k); // Intervalo Bootstrap não-paramétrico
ICinf=zeros(L,k);
ICsup=zeros(L,k);
somarealdentro=zeros(L,k);
somarealforainf=zeros(L,k);
somarealforasup=zeros(L,k);
ICfinalinf=zeros(1,k);
ICfinalsup=zeros(1,k);
mediareal=zeros(L,k);
Mediafinal=zeros(1,k);
Amplitudefinal=zeros(1,k);
finaldentro=zeros(1,k);
finalforainf=zeros(1,k);
finalforasup=zeros(1,k);

AmplitudePar=zeros(L,k); // Intervalo Bootstrap paramétrico
ICinfPar=zeros(L,k);
ICsupPar=zeros(L,k);
somarealdentroPar=zeros(L,k);
somarealforainfPar=zeros(L,k);
somarealforasupPar=zeros(L,k);
ICfinalinfPar=zeros(1,k);
ICfinalsupPar=zeros(1,k);
mediarealPar=zeros(L,k);
MediafinalPar=zeros(1,k);
AmplitudefinalPar=zeros(1,k);
finaldentroPar=zeros(1,k);
finalforainfPar=zeros(1,k);
finalforasupPar=zeros(1,k);

previsaoAs=zeros(L,k); // Intervalo Assintótico
LimInfAs=zeros(L,k);
LimSupAs=zeros(L,k);
AmplitudeAs=zeros(L,k);
somarealdentroAs=zeros(L,k);
somarealforainfAs=zeros(L,k);
somarealforasupAs=zeros(L,k);
ICfinalinfAs=zeros(1,k);
ICfinalsupAs=zeros(1,k);
MediafinalAs=zeros(1,k);
AmplitudefinalAs=zeros(1,k);
finaldentroAs=zeros(1,k);
finalforainfAs=zeros(1,k);
finalforasupAs=zeros(1,k);

for(l=0;l<L;l++){
    println("\nPasso da simulação \t",l,"\n");
    Chamando a função que simula a série MEB
    Ytsim=STSSimulateSBM(sig2eta,sig2csi,sig2eps,sig2omg,sazo,m);
    Yt=zeros(1,n);
    for(x=0;x<n;x++){
        Yt[0,x]=Ytsim[0,x];
    }
    Real=zeros(1,k);
    for(x=0;x<k;x++){
        Real[0,x]=Ytsim[0,n+x];
    }
    // Estimando os hiperparâmetros do MEB pelo método BFGS
    g_STSmOmega=STSAadjust(Yt,STSmModelSBM);
    ii=rows(g_STSmOmega);
    sigma2eta=g_STSmOmega[0][0];

```

```

sigma2csi=g_STSmOmega[1][1];
sigma2omg=g_STSmOmega[2][2];
sigma2eps=g_STSmOmega[ii-1][ii-1];
somaetabfgs=sigma2eta+somaetabfgs;
somaacsibfgs=sigma2csi+somaacsibfgs;
somaomgbfgs=sigma2omg+somaomgbfgs;
somaepsbfgs=sigma2eps+somaepsbfgs;
// Chamando a funcao que calcula as previsões assintóticas com a série Yt e os hiperparâmetros estimados
[YprevAs,LInf,LSup]= STSForecastAs(Yt,k,z);
for(t=0;t<k;t++){
    previsaoAs[1][t]=YprevAs[0,t];
    LimInfAs[1][t]=LInf[0,t];
    LimSupAs[1][t]=LSup[0,t];
    AmplitudeAs[1][t]=LimSupAs[1][t]-LimInfAs[1][t];
}
somaeta=0.0;
somacsi=0.0;
somaomg=0.0;
somaeps=0.0;

somaetaPar=0.0;
somacsiPar=0.0;
somaomgPar=0.0;
somaepsPar=0.0;

// Previsão Bootstrap
previsao=zeros(B,k);
previsaoPar=zeros(B,k);
for(j=0;j<B;j++){
    g_STSmOmega[0][0]=sigma2eta;
    g_STSmOmega[1][1]=sigma2csi;
    g_STSmOmega[2][2]=sigma2omg;
    g_STSmOmega[ii-1][ii-1]=sigma2eps;
// Gerando a série Yt bootstrap
[Ytb]=STSBoot(Yt,k);
// Estimando o sigma2epsboot e o sigma2etaboot pelo método BFGS para Ytboot
g_STSmOmega=STSAdjust(Ytb,STSmModelSBM);
ii=rows(g_STSmOmega);
sigma2etaboot=g_STSmOmega[0][0];
sigma2csiboot=g_STSmOmega[1][1];
sigma2omgboot=g_STSmOmega[2][2];
sigma2epsboot=g_STSmOmega[ii-1][ii-1];
somaeta=somaeta+sigma2etaboot;
somacsi=somacsi+sigma2csiboot;
somaomg=somaomg+sigma2omgboot;
somaeps=somaeps+sigma2epsboot;
// Chamando a funcao que calcula as previsões boot com a série Yt e os hiperparâmetros boot
Yprev=STSForecast(Yt,k);
for(t=0;t<k;t++){
    previsao[j][t]=Yprev[0,t];
}

g_STSmOmega[0][0]=sigma2eta;
g_STSmOmega[1][1]=sigma2csi;
g_STSmOmega[2][2]=sigma2omg;
g_STSmOmega[ii-1][ii-1]=sigma2eps;
// Estimando o sigma2epsboot e o sigma2etaboot pelo método BFGS para Ytboot

[YtbPar,Eta,Eps,Csi,Omg]=STSBootParMEB(Yt,sigma2eta,sigma2eps,sigma2csi,sigma2omg,k);
g_STSmOmega=STSAdjust(YtbPar,STSmModelSBM);
ii=rows(g_STSmOmega);
sigma2etabootPar=g_STSmOmega[0][0];
sigma2csibootPar=g_STSmOmega[1][1];
sigma2omgbootPar=g_STSmOmega[2][2];
sigma2epsbootPar=g_STSmOmega[ii-1][ii-1];
somaetaPar=somaetaPar+sigma2etabootPar;

```

```

somacsiPar=somacsiPar+sigma2csibootPar;
somaomgPar=somaomgPar+sigma2omgbootPar;
somaepsPar=somaepsPar+sigma2epsbootPar;
// Chamando a funcao que calcula as previsões boot paramétricas com a série Yt e os
// hiperparâmetros boot paramétrico
YprevPar=STSForecastPar(Yt,Eps,Eta,Csi,Omg,k);
for(t=0;t<k;t++){
    previsaoPar[j][t]=YprevPar[0,t];
}
}
esteta=somaeta/B;
estcsi=somacsi/B;
estomg=somaomg/B;
esteps=somaeps/B;
somaesteta=esteta+somaesteta;
somaestcsi=estcsi+somaestcsi;
somaestomg=estomg+somaestomg;
somaesteps=esteps+somaesteps;

estetaPar=somaetaPar/B;
estcsiPar=somacsiPar/B;
estomgPar=somaomgPar/B;
estepsPar=somaepsPar/B;

somaestetaPar=estetaPar+somaestetaPar;
somaestcsiPar=estcsiPar+somaestcsiPar;
somaestomgPar=estomgPar+somaestomgPar;
somaestepsPar=estepsPar+somaestepsPar;

IC=zeros(B,k);
IC=sortc(previsao);
ICPar=zeros(B,k);
ICPar=sortc(previsaoPar);
inf=(alfa/2)*B;
sup=B-((alfa/2)*B);
liminf=round(inf)-1;
limsup=round(sup)-1;
for(t=0;t<k;t++){
    ICinf[1][t]=IC[liminf][t];
    ICsup[1][t]=IC[limsup][t];
    Amplitude[1][t]=ICsup[1][t]-ICinf[1][t];
    mediareal[1][t]=sumc(previsao[1][t])/B;
    ICinfPar[1][t]=ICPar[liminf][t];
    ICsupPar[1][t]=ICPar[limsup][t];
    AmplitudePar[1][t]=ICsupPar[1][t]-ICinfPar[1][t];
    mediarealPar[1][t]=sumc(previsaoPar[1][t])/B;
}
for(t=0;t<k;t++){
    if ((Real[0][t]>=ICinf[1][t]) && (Real[0][t]<=ICsup[1][t])){
        somarealdentro[1][t]=1;
    }
    if (Real[0][t]<ICinf[1][t]){
        somarealforainf[1][t]=1;
    }
    if (Real[0][t]>ICsup[1][t]){
        somarealforasup[1][t]=1;
    }
}
for(t=0;t<k;t++){
    if ((Real[0][t]>=ICinfPar[1][t]) && (Real[0][t]<=ICsupPar[1][t])){
        somarealdentroPar[1][t]=1;
    }
    if (Real[0][t]<ICinfPar[1][t]){
        somarealforainfPar[1][t]=1;
    }
    if (Real[0][t]>ICsupPar[1][t]){

```

```

        somarealforasupPar[l][t]=1;
    }
}
for(t=0;t<k;t++){
    if ((Real[0][t]>=LimInfAs[l][t]) && (Real[0][t]<=LimSupAs[l][t])){
        somarealdentroAs[l][t]=1;
    }
    if (Real[0][t]<LimInfAs[l][t]){
        somarealforainfAs[l][t]=1;
    }
    if (Real[0][t]>LimSupAs[l][t]){
        somarealforasupAs[l][t]=1;
    }
}
}
estetabfgs=somaetabfgs/L;
estcsibfgs=somacsibfgs/L;
estomgbfgs=somaomgbfgs/L;
estepsbfgs=somaepsbfgs/L;

estetafinal=somaesteta/L;
estcsifinal=somaestcsi/L;
estomgfinal=somaestomg/L;
estepsfinal=somaesteps/L;

estetafinalPar=somaestetaPar/L;
estcsifinalPar=somaestcsiPar/L;
estomgfinalPar=somaestomgPar/L;
estepsfinalPar=somaestepsPar/L;

ICfinalinf=sumc(ICinf)/L;
ICfinalsup=sumc(ICsup)/L;
Mediafinal=sumc(mediareal)/L;
Amplitudefinal=sumc(Amplitude)/L;
finaldentro=sumc(somarealdentro);
finalforainf=sumc(somarealforainf);
finalforasup=sumc(somarealforasup);

ICfinalinfPar=sumc(ICinfPar)/L;
ICfinalsupPar=sumc(ICsupPar)/L;
MediafinalPar=sumc(mediarealPar)/L;
AmplitudefinalPar=sumc(AmplitudePar)/L;
finaldentroPar=sumc(somarealdentroPar);
finalforainfPar=sumc(somarealforainfPar);
finalforasupPar=sumc(somarealforasupPar);

ICfinalinfAs=sumc(LimInfAs)/L;
ICfinalsupAs=sumc(LimSupAs)/L;
MediafinalAs=sumc(previsaoAs)/L;
AmplitudefinalAs=sumc(AmplitudeAs)/L;
finaldentroAs=sumc(somarealdentroAs);
finalforainfAs=sumc(somarealforainfAs);
finalforasupAs=sumc(somarealforasupAs);

println("\t\n***** Hiperparametros reais para as series simuladas *****\t\n");
println("Parametro\ Estimativa");
println("Sigma2Eta\t",sig2eta,"\t");
println("Sigma2Csi\t",sig2csi,"\t");
println("Sigma2Omg\t",sig2omg,"\t");
println("Sigma2Eps\t",sig2eps,"\t");

println("\t\n***** Hiperparametros estimados-BFGS para as series simuladas *****\t\n");
println("Parametro\ Estimativa");
println("Sigma2Eta\t",estetabfgs,"\t");
println("Sigma2Csi\t",estcsibfgs,"\t");
println("Sigma2Omg\t",estomgbfgs,"\t");

```

```

println("Sigma2Eps\t",estepsbfgs,"\t");

println("\t\n**** Hiperparametros bootstrap Não-paramétrico estimados para as series simuladas **\t\n");
println("Parametro\ Estimativa");
println("Sigma2EtaBoot\t",estetafinal,"\t");
println("Sigma2CsiBoot\t",estcsifinal,"\t");
println("Sigma2OmgBoot\t",estomgfinal,"\t");
println("Sigma2EpsBoot\t",estepsfinal,"\t");

println("\t\n***** Hiperparametros bootstrap Paramétrico estimados para as series simuladas ***\t\n");
println("Parametro\ Estimativa");
println("Sigma2EtaBoot\t",estetafinalPar,"\t");
println("Sigma2CsiBoot\t",estcsifinalPar,"\t");
println("Sigma2OmgBoot\t",estomgfinalPar,"\t");
println("Sigma2EpsBoot\t",estepsfinalPar,"\t");
println("\t\n*****\t\n");

println("\t\n***** Intervalo de confianca Bootstrap Não-paramétrico*****\t\n");
println(" Passo 1 Passo 2 Passo 3 Passo 4 Passo 5 Passo 6\t");
println("Valor medio das previsoos",Mediafinal,"\t");
println("Lim Inf",ICfinalinf,"\t");
println("Lim Sup",ICfinalsup,"\t");
println("Amplitude do intervalo",Amplitudefinal,"\t");
println("Numero de vezes que o valor real ficou dentro do IC ",finaldentro,"\t");
println("Numero de vezes que o valor real ficou abaixo do limite inferior",finalforainf,"\t");
println("Numero de vezes que o valor real ficou acima do limite superior",finalforasup,"\t");
println("\t\n*****\t\n");

println("\t\n***** Intervalo de confianca Boostrap Paramétrico*****\t\n");
println(" Passo 1 Passo 2 Passo 3 Passo 4 Passo 5 Passo 6\t");
println("Valor medio das previsoos",MediafinalPar,"\t");
println("Lim Inf",ICfinalinfPar,"\t");
println("Lim Sup",ICfinalsupPar,"\t");
println("Amplitude do intervalo",AmplitudefinalPar,"\t");
println("Numero de vezes que o valor real ficou dentro do IC ",finaldentroPar,"\t");
println("Numero de vezes que o valor real ficou abaixo do limite inferior",finalforainfPar,"\t");
println("Numero de vezes que o valor real ficou acima do limite superior",finalforasupPar,"\t");
println("\t\n*****\t\n");

println("\t\n***** Intervalo de confianca Assintotico *****\t\n");
println(" Passo 1 Passo 2 Passo 3 Passo 4 Passo 5 Passo 6\t");
println("Valor medio das previsoos",MediafinalAs,"\t");
println("Lim Inf",ICfinalinfAs,"\t");
println("Lim Sup",ICfinalsupAs,"\t");
println("Amplitude do intervalo",AmplitudefinalAs,"\t");
println("Numero de vezes que o valor real ficou dentro do IC",finaldentroAs,"\t");
println("Numero de vezes que o valor real ficou abaixo do limite inferior",finalforainfAs,"\t");
println("Numero de vezes que o valor real ficou acima do limite superior",finalforasupAs,"\t");
println("\t\n*****\t\n");
}

```