



**UNIVERSIDADE TÉCNICA DE LISBOA**

**INSTITUTO SUPERIOR TÉCNICO**

**MODELIZAÇÃO E CONTROLO DE SISTEMAS  
ELECTROMECAÑNICOS POR INFERÊNCIA  
GRAMATICAL**

**João Francisco Alves Martins**

(Mestre)

**Dissertação para obtenção do grau de Doutor em  
Engenharia Electrotécnica e de Computadores**

**Orientador:** Doutor Joaquim António Fraga Gonçalves Dente  
**Co-Orientadores:** Doutor Rui Luís Vilela Lima Mendes  
Doutor Armando José Pinheiro Marques Pires

**Presidente:** Reitor da Universidade Técnica de Lisboa  
**Vogais:** Doutor Joaquim António Fraga Gonçalves Dente  
Doutor Armando José Pinheiro Marques Pires  
Doutor José António Tenreiro Machado  
Doutor Rui Luís Vilela Lima Mendes  
Doutor Ana Luísa Nobre Fred  
Doutor Paulo José da Costa Branco

**Lisboa, Julho 2002**



*À minha querida Família*



# RESUMO

A utilização da máquina assíncrona em accionamentos electromecânicos encontra-se cada vez mais generalizada em vários processos industriais, onde maiores exigências, quer económicas quer de desempenho, lhe são colocadas. O controlo destes sistemas, factor essencial para um bom desempenho, apresenta-se complexo fazendo uso de complicados procedimentos teóricos e numéricos.

Os desenvolvimentos tecnológicos alcançados nos últimos anos têm potenciado a aplicação de novas metodologias de comando e controlo, ditas de *soft computing*, na realização de sistemas de accionamento electromecânico “inteligentes”.

As técnicas baseadas em linguagens formais têm sido utilizadas para estudar sistemas dinâmicos autónomos. Contudo, para sistemas dinâmicos comandados são necessárias novas especificações para distinguir entre a informação gerada pela dinâmica interna do sistema e pela que depende da entrada de comando.

Nesta perspectiva, estuda-se neste trabalho a utilização de algoritmos de aprendizagem através de exemplos baseados em elementos da linguagem formal. Considera-se um sistema dinâmico como uma entidade capaz de gerar uma linguagem específica, existindo uma gramática responsável pela caracterização dessa mesma linguagem.

A formalização gramatical desenvolvida para sistemas dinâmicos comandados conduz a uma formulação em termos de linguagens dependentes do contexto, tendo sido estabelecido um algoritmo de inferência gramatical responsável pelo estabelecimento das produções gramaticais que representam a linguagem gerada pelo sistema dinâmico.

Apresentam-se ainda um conjunto de resultados experimentais que validam a utilização dos algoritmos de inferência gramatical em sistemas de accionamento electromecânico, tanto ao nível da modelização automática como do comando, controlo e detecção de anomalias.

**Palavras chave:** Máquina de indução, Accionamento electromecânico, Aprendizagem através de exemplos, Linguagem formal, Inferência gramatical, Controlo inteligente.



# ABSTRACT

The benefits of using squirrel-cage induction machines are well known. Its use in variable speed electrical drive systems it's widespread through various industrial processes, with growing economical and performing demands. However, the control of this type of systems is a complex task, requiring complicated numerical and theoretical procedures.

During the past years, technological developments provide the application of new command and control methodologies, known as *soft computing*, within intelligent electrical drive systems.

Formal language theory has been used in the past to study autonomous systems. However, for controlled dynamical systems new specifications are needed in order to distinguish between the information that is generated by the dynamics of the system and the one that depends on the free will of the controlling operator.

In this work, elements of the formal language theory are used with learning through examples algorithms. The dynamical system is assumed as a linguistic source producing a certain language, characterized by a grammar.

The modeling framework for controlled dynamical systems leads naturally to a formulation in terms of context-dependent grammars. A grammatical inference algorithm is developed in order to extract the productions that govern the grammar representing the dynamical system's language.

A set of experimental learning results validates the use of grammatical inference algorithms within electrical drive systems automatic modelling, command, control and anomaly detection.

**Keywords:** Induction machine, Electrical drive, Learning through examples, Formal language, Grammatical inference, Intelligent control.



# AGRADECIMENTOS

A elaboração de um trabalho de doutoramento implica um contributo generoso e amigo de inúmeras pessoas. Quero aqui expressar o meu sincero reconhecimento a todos os que não vou mencionar, e que no seu anonimato foram muito importantes para a concretização deste trabalho. Bem hajam.

Depois há aqueles que pela sua importância e natureza do seu contributo não posso, nem devo, deixar de mencionar.

Agradeço de uma forma muito especial ao meu orientador científico Prof. Dr. António Dente, pelo seu profundo empenho, trabalho, total disponibilidade, imprescindível apoio e incentivo. Através das suas sugestões e críticas soube sempre, principalmente nos momentos de maior dúvida, contribuir da melhor forma para a conclusão do presente trabalho.

Ao meus co-orientadores científicos Prof. Dr. Rui Vilela Mendes e Prof. Dr. Armando Pires, o mais grato agradecimento pelo empenho e confiança em mim depositados. Agradeço a total solicitude, compreensão das dificuldades, discussões e sugestões, que muito serviram para o superar das dificuldades. Desejo também reconhecer a amizade demonstrada e o bom ambiente de trabalho proporcionado.

Agradeço também a todos os meus colegas da Escola Superior de Tecnologia de Setúbal pelo incentivo sempre demonstrados.

Agradeço também a todos os docentes e colegas da Secção de Máquinas Eléctricas e Electrónica de Potência, do Instituto Superior Técnico, a amizade e incentivos demonstrados ao longo da realização do trabalho.

Pelo apoio material e financeiro desejo agradecer às seguintes entidades:

- Escola Superior de Tecnologia de Setúbal e Instituto Politécnico de Setúbal;
- Secção de Máquinas Eléctricas e Electrónica de Potência, Departamento de Engenharia Electrotécnica e de Computadores, Instituto Superior Técnico, Universidade Técnica de Lisboa.

Um agradecimento muito especial a todos os meus amigos, pelo tempo que forçosamente lhes roubei.

Finalmente um agradecimento, difícil de expressar por palavras, à minha família. Aos meus pais, pelo carinho de longa data. À Ritinha, a quem roubei alguns momentos únicos de camaradagem. À Manuela, pelo carinho, amizade, compreensão, apoio e incentivo que me ajudaram definitivamente a tomar menos árdua a execução deste trabalho. Para elas um enorme beijinho de reconhecido agradecimento.

## SIMBOLOGIA

$A$	- Conjunto de saídas de um autómato.
$A, B, C$	- Matrizes do modelo resultante da linearização em torno de um ponto de funcionamento.
$B$	- Coeficiente de atrito.
$B$	- Conjunto de entradas de um autómato.
$C$	- Classe de gramáticas.
$c(\cdot)$	- Custo.
$c_k$	- Comando de cada braço do ondulator autónomo de tensão.
$d(\cdot)$	- Distância.
$E$	- Erro quadrático médio.
$e_i$	- Operação de edição.
$e_1 \dots e_n$	- Entradas dos nós de uma rede neuronal.
<b>Ex</b>	- Conjunto de exemplos.
$f(\cdot)$	- Função da dinâmica do modelo de estado discreto.
$F(\cdot)$	- Função da dinâmica do estado de um autómato.
$q(\cdot)$	- Função de aprendizagem para autómatos.
$f_R$	- Frequência do relógio da placa de aquisição de dados.
$G$	- Gramática.
$g(\cdot)$	- Relação funcional do modelo de estado discreto.
$G_{dq}$	- Matriz das variações dos coeficientes de indução, em ordem à posição angular do rotor, dos vários circuitos da máquina assíncrona, na representação dq.

$\hat{g}(\cdot)$	- Relação funcional do modelo de estado discreto, extraída por um algoritmo de aprendizagem.
$\tilde{g}(\cdot)$	- Relação funcional do modelo discreto de um sistema dinâmico sujeito a anomalia.
$h(\cdot)$	- Função de saída do modelo de estado discreto.
$H(\cdot)$	- Função da dinâmica da saída de um autómato.
$\mathbf{i}_{abc}$	- Vector das correntes nos diversos circuitos da máquina assíncrona, na representação abc.
$i_C$	- Corrente que percorre o condensador do filtro LC.
$\mathbf{i}_{dq}$	- Vector das correntes nos diversos circuitos da máquina assíncrona, na representação dq.
$i_{ds}, i_{qs}$	- Intensidade da corrente estatórica na máquina assíncrona, em ambas as componentes dq.
$\mathbf{I}_{ds}, \mathbf{I}_{qs}$	- Intensidade da corrente estatórica na máquina assíncrona, em ambas as componentes dq, num dado ponto de funcionamento.
$i_{dsref}, i_{qsref}$	- Referências da intensidade da corrente estatórica na máquina assíncrona, em componentes dq.
$i_k$	- Corrente em cada braço do ondulator autónomo de tensão.
$i_L$	- Corrente de saída do rectificador trifásico em ponte.
$i_{LC}$	- Corrente de saída do filtro LC.
$J$	- Momento de inércia.
$J_i$	- Momento de inércia de um sistema inercial.
$L$	- Linguagem.
$\mathbf{L}_{abc}(\cdot)$	- Matriz dos coeficientes de indução da máquina assíncrona na representação abc.
$\mathbf{L}_{dq}$	- Matriz dos coeficientes de indução da máquina assíncrona na representação dq.
$L_m$	- Inductância interna da fonte de tensão trifásica.

$L_r$	- Coeficiente de auto-indução dos circuitos rotóricos da máquina assíncrona.
$M$	- Coeficiente de indução mútua entre estator e rotor da máquina assíncrona.
$P$	- Conjunto de produções.
$\mathbf{p}$	- Conjunto de parâmetros.
$p_{\max}$	- Tipo de produção máxima.
$R$	- Sistema de rescrita.
$\mathbf{R}_{abc}$	- Matriz das resistências por fase nos diversos circuitos da máquina assíncrona, na representação abc.
$\mathbf{R}_{dq}$	- Matriz das resistências por fase nos diversos circuitos da máquina assíncrona, na representação dq.
$R_m$	- Resistência eléctrica interna da fonte de tensão trifásica.
$R_r$	- Resistência eléctrica dos circuitos rotóricos da máquina assíncrona.
$s$	- Escorregamento da máquina assíncrona.
$S$	- Símbolo inicial.
$s_{op}$	- Sequência de operações.
$S(\cdot)$	- Função de comutação genérica de um sistema em modo de deslizamento.
$S_f$	- Saída de um nó da rede neuronal.
$t$	- Variável tempo.
$t_0$	- Instante de tempo inicial.
$T_{em}$	- Binário electromagnético da máquina assíncrona.
$T_{ext}$	- Binário exterior.
$T_i$	- Binário aplicado de um sistema inercial.
$\mathbf{u}$	- Vector das variáveis de comando.

$U$	- Símbolo não terminal.
$U^+, U^-$	- Comandos genéricos aplicados a um sistema em modo de deslizamento.
$u_1, u_2, u_3$	- Tensões das fases do sistema de tensão trifásico disponibilizado pela fonte.
$u_{ab}, u_{bc}, u_{ca}$	- Tensões compostas do sistema de tensão trifásico gerado pelo ondulator autónomo de tensão.
$\mathbf{u}_{abc}$	- Vector das tensões nos diversos circuitos da máquina assíncrona, na representação abc.
$\mathbf{u}_{dq}$	- Vector das tensões nos diversos circuitos da máquina assíncrona, na representação dq.
$u_{ds}, u_{qs}$	- Tensões estatóricas na máquina assíncrona, em ambas as componentes dq.
$u_{LC}$	- Tensão de saída do filtro LC.
$u_R$	- Tensão de saída do rectificador trifásico em ponte.
$\tilde{\mathbf{u}}$	- Vector das variáveis de entrada, num ponto de funcionamento.
$\mathbf{u}_\delta$	- Vector das variações das variáveis de entrada, em torno de um ponto de funcionamento.
$w_a, w_{ij}, w_{pij}$	- Pesos sinápticos das redes de autómatos.
$W_c$	- Co-energia magnética.
$W_{f,i1,\dots,in}$	- Conexões sinápticas da rede neuronal de ordem n.
$\mathbf{x}$	- Vector das variáveis de estado.
$\mathbf{X}$	- Conjunto das variáveis significativas.
$x$	- Variável de estado de um sistema dinâmico.
$\mathbf{x}_0$	- Vector dos valores iniciais das variáveis de estado.
$\tilde{\mathbf{x}}$	- Vector das variáveis de estado, num ponto de funcionamento.
$\tilde{\mathbf{x}}_0$	- Vector das condições iniciais das variáveis de estado, num ponto de funcionamento.

$\mathbf{x}_\delta$	- Vector das variações das variáveis de estado, em torno de um ponto de funcionamento.
$\mathbf{x}_{0\delta}$	- Vector das variações das condições iniciais das variáveis de estado, em torno de um ponto de funcionamento.
$\mathbf{y}$	- Vector das variáveis de saída.
$y$	- Variável de saída de um sistema dinâmico.
$y$	- Símbolo terminal.
$\mathbf{Y}$	- Conjunto sequencial de saídas de um sistema dinâmico amostrado.
$y_f$	- Polarização de um nó da rede neuronal.
$\hat{\mathbf{y}}$	- Vector das variáveis de saída estimado pela relação funcional que modeliza o sistema.
$\tilde{\mathbf{y}}$	- Vector das variáveis de saída, num ponto de funcionamento.
$\mathbf{y}_\delta$	- Vector das variações das variáveis de saída, em torno de um ponto de funcionamento.
$\tilde{y}$	- Variável de saída de um sistema dinâmico sujeito a anomalia.
$z$	- Palavra.
$\delta$	- Símbolo não terminal especial.
$\Delta T$	- Intervalo de amostragem.
$\varepsilon$	- Erro admissível.
$\varepsilon_{\alpha\beta}$	- Erro apresentado pelas correntes de saída do ondulator autónomo de tensão, na representação $\alpha\beta$ .
$\varepsilon_{i_b}$	- Erro de corrente na fase 'b' do ondulator autónomo de tensão.
$\varepsilon_{i_c}$	- Erro de corrente na fase 'c' do ondulator autónomo de tensão.
$\Phi(\cdot)$	- Função da dinâmica de um modelo não linear.
$\gamma_{ap}$	- Taxa de aprendizagem.
$\gamma_{esq}$	- Taxa de esquecimento.

$\varphi(\cdot)$	- Relação funcional de um modelo contínuo.
$\lambda$	- Palavra vazia.
$\theta$	- Posição angular do veio da máquina assíncrona.
$\Theta$	- Conjunto de estados internos de um autômato.
$\theta_i$	- Posição angular de um sistema inercial.
$\theta_R$	- Posição angular do referencial dq genérico.
$\sigma$	- Coeficiente de dispersão magnética.
$\Sigma$	- Alfabeto.
$\Sigma^*$	- Conjunto de todas as palavras de um alfabeto.
$\Sigma^+$	- Conjunto de todas as palavras não vazias de um alfabeto.
$\Sigma_N$	- Alfabeto não terminal.
$\Sigma_T$	- Alfabeto terminal.
$\tau_m$	- Constante de tempo associada às grandezas mecânicas de um accionamento electromecânico.
$\tau_r$	- Constante de tempo associada aos circuitos rotóricos da máquina assíncrona.
$\tau_s$	- Constante de tempo associada aos circuitos estatóricos da máquina assíncrona.
$\omega$	- Velocidade angular do veio da máquina assíncrona.
$\omega$	- Velocidade angular do veio da máquina assíncrona, num dado ponto de funcionamento.
$\omega_i$	- Velocidade angular de um sistema inercial.
$\omega_R$	- Velocidade angular do referencial dq genérico.
$\omega_s$	- Frequência das grandezas do estator da máquina assíncrona.
$\omega_s$	- Frequência das grandezas do estator da máquina assíncrona, num dado ponto de funcionamento.
$\Psi(\cdot)$	- Função de saída de um modelo não linear.

- $\psi(\cdot)$  - Função para os fluxos magnéticos de ligação, na máquina assíncrona.
- $\Psi_{abc}$  - Vector dos fluxos ligados com os diversos circuitos da máquina assíncrona, na representação abc.
- $\psi_{dr}, \psi_{qr}$  - Fluxos magnéticos ligados com os circuitos rotóricos, em componentes dq.
- $\Psi_{dr}, \Psi_{qr}$  - Fluxos magnéticos ligados com os circuitos rotóricos, em componentes dq, num dado ponto de funcionamento.



# ÍNDICE

<b>Capítulo 1 -</b>	<b>INTRODUÇÃO .....</b>	<b>23</b>
1.1	Enquadramento e motivação .....	25
1.2	Objectivos.....	31
1.3	Estrutura da dissertação.....	33
1.3.1	Organização geral .....	33
1.3.2	Convenções.....	34
<b>Capítulo 2 -</b>	<b>TEORIA FORMAL DA LINGUAGEM .....</b>	<b>37</b>
2.1	Introdução.....	39
2.2	Conceitos fundamentais .....	41
2.3	Inferência gramatical.....	45
<b>Capítulo 3 -</b>	<b>INFERÊNCIA GRAMATICAL DE UM SISTEMA DINÂMICO COMANDADO .....</b>	<b>49</b>
3.1	Introdução.....	51
3.2	As linguagens e os sistemas dinâmicos .....	53
3.3	Conceito de aprendizagem .....	55
3.4	Aprendizagem por exemplos.....	61
3.5	Algoritmo de aprendizagem através de exemplos.....	65
3.5.1	Formalização da gramática.....	65
3.5.1.1	Alfabetos .....	68
3.5.1.2	Produções .....	69
3.5.2	Algoritmo de Inferência gramatical.....	73
3.5.3	Lacunas gramaticais e interpolação gramatical .....	80
3.6	Identificação de palavras como padrões.....	85
3.7	Estudo das características de uma gramática inferida a partir de um sistema dinâmico comandado.....	93
3.7.1	Especificação da gramática.....	94
3.7.2	Reprodução e preenchimento das lacunas gramaticais....	97
3.7.3	Influência do conjunto de treino .....	102
3.7.4	Capacidade de generalização.....	107

3.7.5	Robustez .....	110
3.8	Conclusões.....	113
<b>Capítulo 4 -</b>	<b>IMPLEMENTAÇÃO DE LINGUAGENS FORMAIS POR</b>	
	<b>SISTEMAS CONEXIONISTAS .....</b>	<b>115</b>
4.1	Introdução.....	117
4.2	Autómatos .....	123
4.3	Primeiro nível do sistema conexionista – Identificação de Símbolos.....	125
4.4	Segundo nível do sistema conexionista – Aprendizagem das produções.....	133
4.4.1	Inferência gramatical da rede das produções.....	136
4.5	Conclusões.....	145
<b>Capítulo 5 -</b>	<b>APLICAÇÕES.....</b>	<b>147</b>
5.1	Introdução.....	149
5.2	Modelização .....	151
5.2.1	Caracterização do accionamento electromecânico .....	153
5.2.1.1	Máquina assíncrona.....	153
5.2.1.2	Conversor electrónico de potência .....	158
5.2.1.3	Modelo de estado do accionamento electromecânico.....	163
5.2.1.4	Simplificação do modelo de estado.....	165
5.2.1.4.1	Controlo de corrente .....	165
5.2.1.4.2	Orientação de campo .....	168
5.2.1.4.3	Dependência de parâmetros.....	169
5.2.2	Relação funcional .....	172
5.2.3	Especificação da gramática.....	174
5.2.3.1	Alfabetos terminal e não terminal .....	174
5.2.3.2	Produções .....	177
5.2.4	Resultados experimentais .....	177
5.2.4.1	Interpolação gramatical.....	179
5.2.4.2	Generalização.....	182
5.3	Detecção de anomalias .....	185
5.3.1	Detecção de anomalias por utilização de gramática.....	186

5.3.2	Detecção de anomalias na máquina assíncrona por análise da corrente estatórica .....	190
5.4	Controlo.....	193
5.4.1	Sistema de accionamento electromecânico .....	193
5.4.2	A Gramática como regulador .....	196
5.4.2.1	Capacidade de adaptação e aprendizagem contínua.....	203
5.4.2.2	Mecanismo de interpolação gramatical.....	207
5.4.3	A Gramática como supervisor .....	209
5.5	Conclusões.....	217
<b>Capítulo 6 -</b>	<b>CONCLUSÕES.....</b>	<b>221</b>
6.1	Síntese do trabalho e contribuições originais .....	223
6.2	Perspectivas de trabalho futuro .....	229
	<b>BIBLIOGRAFIA .....</b>	<b>231</b>
<b>Apêndice A -</b>	<b>REDES NEURONAIS E SISTEMAS LÓGICOS .....</b>	<b>245</b>
A.i	Introdução.....	247
A.ii	Representação conexionista de um sistema lógico.....	249



# CAPÍTULO 1 - INTRODUÇÃO

**Resumo:** Neste capítulo apresenta-se o enquadramento geral e a motivação que serviu de base ao presente trabalho, sendo ainda descritos os seus objectivos. Apresenta-se também a estrutura geral da dissertação, a sua organização geral e as convenções utilizadas.



## 1.1 ENQUADRAMENTO E MOTIVAÇÃO

A engenharia, de um ponto de vista abrangente, preocupa-se com a compreensão, utilização e controlo de materiais e/ou forças naturais para o benefício da humanidade. De uma forma mais particular é possível considerar um sub-campo da engenharia onde os aspectos importantes são a compreensão e o controlo de segmentos do meio envolvente, que geralmente se designam por sistemas, de modo a providenciar produtos e soluções, úteis e económicas, para a sociedade. Os objectivos de compreender e controlar apresentam-se como complementares pois o controlo de um qualquer sistema implica o conhecimento e modelização desse mesmo sistema.

Quando se pretende descrever um qualquer sistema dinâmico é possível adoptar uma perspectiva mais ou menos clássica em função do objectivo a atingir. Aqui o termo ‘clássica’ é empregue no seu sentido mais lato e em oposição ao termo *soft computing*. Entende-se por *soft computing* uma compilação de metodologias de computação que desempenham um papel principal na concepção, projecto e utilização de informação, conduzindo aos chamados ‘sistemas inteligentes’. Como áreas de referência no campo do *soft computing* podem referir-se a lógica *fuzzy*, as redes neuronais, os algoritmos genéticos, a computação probabilística ou o *machine learning*.

A utilização de modelos demasiado complexos pode ser posta em causa pela sua dificuldade de utilização. Por outro lado, a utilização de modelos demasiado simples pode não ser satisfatória, do ponto de vista do desempenho, pois estes modelos podem não conter informação crucial sobre aspectos importantes do sistema. Este é um problema difícil na área da engenharia, que não se encontra completamente resolvido, devido essencialmente à consideração ou não de fenómenos que se prendem com a eventual existência de dinâmicas fortemente não lineares, a dificuldade em obter várias variáveis essenciais, o tipo de comando gerado pelo *hardware* de controlo, a variação de parâmetros dos sistemas, entre outros.

Os modernos sistemas de accionamento electromecânico, em particular os de velocidade variável equipados com máquinas assíncronas de rotor em gaiola – máquinas robustas e de manutenção pouco exigente – desempenham um papel fundamental na maioria das actuais aplicações industriais. As recentes evoluções ao nível de todos os seus componentes – máquina eléctrica, conversor electrónico de potência, processadores

digitais e sensores electrónicos ou mecânicos – têm sido determinantes no gradual aumento da automatização e produtividade, com a conseqüente e cada vez melhor utilização racional de energia, aliados a um aumento da complexidade de todo o sistema.

Neste domínio dos accionamentos electromecânicos, área onde o presente trabalho se enquadra, o conhecimento destes sistemas físicos traduz-se, do ponto de vista clássico, na existência de equações algébricas e diferenciais que formam um conjunto de relações matemáticas estabelecidas a partir do conhecimento dos fenómenos electromagnéticos, termodinâmicos e mecânicos.

No entanto este conjunto de equações apresenta-se inadequado ao estudo da dinâmica da máquina pelo que se utilizam modelos baseados na teoria de circuitos, onde geralmente se escolhem como variáveis electromagnéticas as tensões de alimentação, as correntes que percorrem os enrolamentos e os fluxos magnéticos de ligação, e como variável mecânica a velocidade e/ou a posição. Certos fenómenos térmicos, mecânicos e electromagnéticos, de difícil modelização e importância relativa reduzida, não são considerados, originando modelos que, apesar de constituírem uma boa representação da realidade, apresentam insuficiências que se podem revelar importantes em aplicações específicas de elevada precisão. Associado a esta dificuldade de modelizar todos os fenómenos físicos presentes, a dinâmica fortemente não linear resultante para o accionamento, a dificuldade em ler algumas das variáveis de estado e a variação de alguns parâmetros em função das condições de funcionamento, dificultam a obtenção de um modelo que seja simultaneamente preciso e fácil de manipular. A máquina assíncrona é modelizada fazendo uso de um conjunto de variáveis de estado, algumas das quais de difícil acesso, nomeadamente os fluxos magnéticos ligados e/ou as correntes que percorrem os enrolamentos do rotor. Esta inacessibilidade coloca problemas sérios na aplicação prática do seu modelo.

Este tipo de abordagem conduz assim a modelos nos quais podem ser considerados vários tipos de limitações: as que resultam da dificuldade de tipificação e quantificação de fenómenos físicos; as que resultam de restrições de ordem matemática, nomeadamente em sistemas não lineares ou de parâmetros variáveis; e as que resultam de limitações impostas pelo próprio utilizador, por forma a reduzir a sua complexidade.

Em resumo, a teoria clássica requer a utilização de um conjunto de equações algébricas e diferenciais que, analiticamente, relacionam variáveis internas, de entrada e saída do accionamento electromecânico. Contudo, estes modelos são geralmente complexos, baseiam-se em algumas considerações prévias, contêm parâmetros de difícil medição e que podem variar durante a operação, e implicam o conhecimento de variáveis de difícil acesso. Existem ainda as limitações referentes à natureza do sistema, pois a teoria clássica encontra-se bem desenvolvida para sistemas lineares e invariantes no tempo, o que não acontece necessariamente num accionamento electromecânico. Estes problemas podem ser minimizados através da utilização de algoritmos ditos ‘inteligentes’, que podem ser utilizados mesmo quando o modelo matemático do sistema não é exactamente conhecido, apresentando-se mais robustos e menos sensíveis a uma eventual variação de parâmetros.

Assim, e em complemento à teoria clássica, é possível considerar a utilização de algoritmos de aprendizagem através de exemplos, dos quais se referenciam alguns, pela sua capacidade de modelizar um sistema físico a partir de dados experimentais e sem necessidade de recorrer a uma análise explícita de um modelo matemático do accionamento. A utilização de filtros de Kalman (Candy, 1988) como estimadores de estado baseia-se num processo recursivo onde a estimativa é constantemente corrigida em função do erro de predição, sendo a sua utilização óptima quando existe conhecimento, *à priori*, do tipo de equações do sistema dinâmico bem como da estrutura do observador utilizado. A utilização da decomposição bi-ortogonal (Aubry, Guyonnet e Lima, 1991) baseia-se na decomposição de um sinal em modos ortogonais por forma a identificar estruturas dinâmicas espaço-temporais complexas. O uso de modelos autoregressivos (Johnsson, 1993) para estimar sinais lineares baseia-se numa regressão estatística efectuada sobre o próprio sinal. A reconstrução através de espaço de estados (Kantz e Schreiber, 1997) apresenta-se como um potente método de análise de sinais não lineares, mas computacionalmente pesado. A utilização de redes neuronais (McClelland e Rumelhart, 1988) recorre a representações distribuídas e atinge as suas conclusões por aplicações de regras evolutivas a valores numéricos, que podem variar de forma contínua. Na modelização por recurso a lógica fuzzy (Branco e Dente, 1998) o comportamento do sistema é descrito por um processo de inferência a partir de um conjunto de regras do tipo ‘se (um conjunto de factos é verificado) então (uma conclusão é considerada)’.

A adopção de algumas perspectivas não clássicas tem encontrado forte aplicação nos modernos sistemas de accionamento electromecânico, podendo ser apresentados como exemplos os trabalhos de (Dente, 1994), (Maia, 1994), (Simões e Bose, 1994), (Martins, 1996), (Vas, Drury e Stronach, 1996) e (Branco, 1998). Estas novas técnicas apresentam algumas vantagens sobre as clássicas: a sua estrutura não obriga à existência de um modelo matemático do sistema, podendo os modelos utilizados serem estabelecidos a partir de conhecimento do tipo linguístico obtido a partir de especialistas ou, na ausência destes, serem estabelecidos a partir de um conjunto de dados obtidos do sistema (ou se possível, recorrer a ambos os tipos de informação); a sua parametrização é mais fácil; apresentam boas capacidades de generalização e de rejeição de ruído, adaptando-se a novas situações de funcionamento.

Com o desenvolvimento da tecnologia dos sistemas de processamento digital o seu uso tem-se generalizado mesmo em aplicações industriais. A geração de variáveis de comando e a aquisição de variáveis de estado e saída é normalmente baseada em técnicas de computação digital, com a consequente conversão analógico-digital e o estabelecimento de uma determinada frequência de amostragem. Estes sistemas trabalham temporalmente de uma forma discreta, pelo que os sinais dos sistemas físicos, naturalmente contínuos no tempo, necessitam de ser discretizados por forma a serem processados. Por outro lado, ao contrário dos sinais analógicos presentes nos sistemas físicos, os processadores digitais efectuam operações em sinais digitais finitos (números) o que obriga a uma digitalização dos sinais que, conjuntamente com a discretização, pode trazer problemas se não forem adequadamente tratados.

Acrescem algumas limitações à utilização não criteriosa destas novas técnicas que se traduzem em alguns inconvenientes: dificuldade em escolher, de uma forma automática, as variáveis significativas para o processo; os algoritmos de aprendizagem desenvolvidos podem tender a tornar-se complexos; não existe imunidade total à acção de variáveis não correlacionadas, como seja o ruído; a generalização é um processo local e não global; e a resposta a novas situações de funcionamento não é instantânea.

Sem aplicações conhecidas, até à presente data, no domínio dos accionamentos electromecânicos, a teoria formal da linguagem preocupa-se com a obtenção de modelos matemáticos da linguagem e de sistemas utilizados para gerar e processar linguagens. Historicamente, a maior contribuição para este campo tem decorrido da investigação em

aplicações específicas e não do estudo das chamadas linguagens naturais. A ideia base consiste na representação finita de linguagens ou classes de linguagens baseadas no conhecimento da sua estrutura sintáctica.

Esta teoria tem sido objecto de várias aplicações, nomeadamente no reconhecimento de padrões e no estudo de sistemas autónomos, onde, neste último caso, a complexidade das sequências geradas é uma medida da complexidade da dinâmica do sistema (Troll, 1993).

Quando se estuda esta área do conhecimento podem adoptar-se várias abordagens. É possível estudar o modelo de uma linguagem do ponto de vista geral ou optar por tentar resolver um determinado problema específico (aqui existe algum conhecimento *á priori* e restringe-se o âmbito do problema). Outra abordagem reporta-se ao tipo de gramática associada a uma determinada linguagem (regular, livre de contexto, ou outra) e como esta pode ser representada (por meio de um autómato finito, probabilístico, ou qualquer outra entidade). Por fim pode ainda analisar-se o tipo de algoritmo necessário ao estabelecimento de uma dada gramática.



## **1.2 OBJECTIVOS**

O principal objectivo do presente trabalho consiste em desenvolver um sistema de modelização, baseado na teoria formal da linguagem, que seja aplicável de uma forma simples e clara a sistemas dinâmicos, em particular a sistemas electromecânicos.

A maioria dos sistemas dinâmicos considerados em aplicações industriais não são autónomos, mas sim sistemas comandados. É assim importante distinguir, desde o início, entre a informação gerada pelo sistema dinâmico e a que depende do comando do mesmo, seja este automático ou dependente da livre vontade de um qualquer operador. Este facto contém algumas implicações na natureza das entidades formais que são utilizadas para descrever as variáveis do sistema e a natureza das linguagens que são geradas.

Outro aspecto importante, relacionado com as aplicações práticas, é a inferência gramatical. Isto é, a possibilidade de extrair gramáticas a partir de dados gerados pelo sistema, independentemente do eventual modelo analítico considerado. Estas gramáticas deverão ser estabelecidas, a partir de um conjunto de informação entrada/saída, por recurso a um algoritmo de aprendizagem expedito que será responsável pela constituição das produções gramaticais. Sendo a gramática obtida a partir de informação real proveniente do sistema dinâmico, a mesma deverá ser obrigatoriamente validada experimentalmente. O mecanismo de aprendizagem deverá ainda prever procedimentos de generalização por forma a estabelecer produções gramaticais aplicáveis a novas situações de funcionamento não verificadas durante o processo de aprendizagem.

A gramática assim obtida, e representativa da eventual linguagem gerada pelo sistema dinâmico, poderá ser estabelecida como uma rede de autómatos. Estas entidades, pela sua não linearidade intrínseca e pelo facto de possuírem parâmetros de nó que dependem da sua actividade em instantes anteriores, possuem características inerentes adequadas à aprendizagem da informação temporal presente nas produções linguísticas. Adicionalmente as características conexionistas destes sistemas potenciam novos paradigmas na representação e extracção de produções gramaticais.

A existência de um modelo gramatical que represente a dinâmica associada a um sistema dinâmico comandado permite considerar a possibilidade da sua utilização no controlo desse mesmo sistema. Por recurso à utilização das produções gramaticais, previamente estabelecidas, determinar-se-á o comando que permita cumprir os objectivos desejados, independentemente das variações paramétricas do sistema e de eventuais perturbações exteriores que sobre ele actuem.

Sendo o reconhecimento de padrões uma das áreas de aplicação por excelência da teoria formal da linguagem, a detecção de anomalias surge como um domínio de aplicação natural destas técnicas. O termo anomalia é aqui entendido como uma variação no sistema que tende a degradar o seu desempenho, sem que isso necessariamente represente uma falha completa, do seu sistema ou de algum dos seus componentes. Deste modo estudar-se-á a utilização dos modelos gramaticais desenvolvidos para efectuar uma atempada detecção de anomalias.

## **1.3 ESTRUTURA DA DISSERTAÇÃO**

### **1.3.1 Organização geral**

A dissertação apresenta uma organização genérica comum a este tipo de trabalho, estando dividida em seis capítulos.

Procurou-se com esta organização apresentar uma sequência lógica do trabalho desenvolvido pelo autor. Os vários temas abordados são interligados por forma a facilitar a sua leitura.

A generalidade dos capítulos são constituídos por Resumo, Introdução, Desenvolvimento e Conclusões. Como excepção, e pela sua natureza, contam-se o presente e o último capítulos, onde não se consideram nem Introdução nem Conclusões, e o próximo capítulo, onde não se consideram Conclusões.

No Resumo é efectuada uma breve descrição do capítulo, com chamada de atenção para os aspectos mais relevantes. Na Introdução são apresentados os temas abordados no capítulo, com uma breve descrição da metodologia empregue nessa abordagem. No Desenvolvimento do capítulo desenvolve-se, por secções, o estudo dos respectivos temas. Nas Conclusões apresenta-se um resumo dos resultados atingidos no capítulo.

No primeiro capítulo introduz-se o tema a tratar, realçando o seu interesse para a área de Engenharia em que se insere. É apresentado o enquadramento do trabalho, a estrutura adoptada para a dissertação e as convenções utilizadas.

No capítulo dois são apresentados os fundamentos da teoria formal da linguagem. Introduzem-se os seus conceitos fundamentais e descreve-se o processo de inferência gramatical. Este capítulo destina-se a que o leitor menos familiarizado com o tema possa concentrar-se nos aspectos mais relevantes para o entendimento deste trabalho.

No capítulo três são apresentados os conceitos desenvolvidos, e que serviram de base a todo o trabalho, com vista à aplicação do mecanismo de inferência gramatical a sistemas

dinâmicos comandados. São investigadas as propriedades dos algoritmos desenvolvidos, por recurso a exemplos de aplicação simples.

No capítulo quatro é descrito o processo de implementação da gramática obtida recorrendo a redes de entidades conexionistas, designadas por autómatos. Adicionalmente é também proposto um método para a geração automática da referida rede de autómatos, a partir da inferência gramatical de informação experimental de entrada/saída.

No capítulo cinco, após descrição do sistema de accionamento electromecânico utilizado na montagem laboratorial, são verificadas experimentalmente as propriedades descritas no capítulo anterior. O algoritmo de inferência gramatical é aplicado por forma a extrair uma relação funcional representativa do comportamento dinâmico do sistema experimental. É dada particular importância aos aspectos da aprendizagem, reprodução e generalização. Além dos aspectos relacionados com a modelização são ainda apresentadas duas outras possíveis aplicações das gramáticas inferidas: a detecção de anomalias e o controlo de sistemas de accionamento electromecânico.

No capítulo seis é efectuada a síntese do trabalho e apresentadas as conclusões gerais mais significativas, bem como as contribuições originais do presente trabalho. Apresentam-se também perspectivas de trabalho futuro.

### **1.3.2 Convenções**

Todos os números de ordem utilizados apresentam-se em numeração árabe, com excepção das páginas iniciais da dissertação (Resumo, Agradecimentos, Simbologia e Índice).

A numeração de figuras, tabelas e equações foi efectuada de forma sequencial ao longo de cada capítulo, referenciadas por dois números separados por um ponto. O primeiro número refere-se ao capítulo e o segundo ao número de ordem da figura, tabela ou equação. O número de ordem da numeração de tabelas é apresentado em numeração romana. A numeração de equações é inserida entre parêntesis.

A legenda de figuras e tabelas é apresentada a carregado.

Na escrita de equações foram consideradas as seguintes convenções: as variáveis e funções são escritas em itálico; as matrizes e vectores em carregado; as constantes, parâmetros e índices em normal. Nas equações referentes a elementos da linguagem formal os símbolos do alfabeto terminal são apresentados em maiúsculo e os símbolos do alfabeto não terminal em minúsculo.

As escalas dos vários gráficos foram graduadas, salvo raras excepções, em ‘por unidade’ para o eixo das ordenadas e ‘segundos’ para o eixo das abcissas.

A citação de referências bibliográficas foi realizada utilizando a forma (Autor,...,Autor,Ano). Nestas citações 'Autor' refere-se ao apelido de cada autor e 'Ano' ao ano de publicação. Quando o autor, ou conjunto de autores, possuírem mais de uma publicação, no mesmo ano, citada no presente trabalho, é colocada uma letra minúscula na sequência de algarismos referentes ao ano. A ordenação das várias referências bibliográficas foi efectuada por ordem alfabética do apelido do primeiro autor da publicação.

Sempre que possível tentou-se evitar o uso de termos estrangeiros na presente dissertação. Quando a utilização de algum termo técnico, que não faça parte da Língua Portuguesa, se encontre vulgarizada optou-se por utilizar este, escrevendo-o em itálico (por exemplo: *hardware*, *fuzzy*, etc.).

O sistema de unidades de medida utilizado foi o Sistema Internacional (SI), salvo referência em contrário.



## CAPÍTULO 2 - TEORIA FORMAL DA LINGUAGEM

**Resumo:** Neste capítulo introduzem-se os conceitos fundamentais da teoria formal da linguagem, necessários ao entendimento dos algoritmos utilizados e desenvolvidos ao longo do presente trabalho. Nesta descrição, necessariamente não exaustiva, apresentam-se um conjunto de referências fundamentais a uma ampla compreensão dos assuntos em discussão. Tecem-se alguns comentários relativos ao conceito de aprendizagem e aprendizagem através de exemplos, descrevendo o mecanismo de aprendizagem subjacente aos algoritmos apresentados.



## **2.1 INTRODUÇÃO**

A aplicação dos conceitos da teoria formal da linguagem têm conhecido no passado recente bastantes aplicações que não se limitam apenas aos domínios da linguística propriamente dita. A origem da teoria formal da linguagem remonta a meados dos anos 50, concretamente ao trabalho de Noam Chomsky, cujo objectivo consistiu no desenvolvimento de modelos matemáticos de gramáticas relacionados com o seu trabalho em linguagens naturais (Chomsky, 1965).

Segundo Steven Pinker (Pinker, 2000) a linguagem natural pode ser um campo por excelência para a aplicação da teoria da evolução. Ela é universal, hereditária, rapidamente adquirida por crianças e adaptável à transmissão de informação bastante complexa. Os princípios encontram-se definidos e os seus parâmetros sofrem uma necessária adaptação. Neste contexto, Nowak (Nowak, Plotkin e Jansen, 2000) defende como uma das propriedades universais da linguagem o agrupar de símbolos (possivelmente sem significado e em número finito) em sequências que definem conceitos (com significado e em número potencialmente infinito). Para uma sequência (palavra) sobreviver num dado vocabulário a mesma deve ser utilizada com uma frequência suficiente.

A descrição de linguagens naturais, por recurso a modelos gramaticais, começou por ser um dos principais objectivos dos linguistas. Apesar desse objectivo ainda não ter sido atingido na globalidade, muito trabalho tem sido desenvolvido nomeadamente noutras áreas que não a linguagem natural como sejam o projecto de compiladores, as linguagens de programação, a teoria de autómatos ou o reconhecimento de padrões.

Hoje em dia as preocupações dos linguistas centram-se em aplicar e desenvolver a teoria formal da linguagem para o desenvolvimento de áreas tão diversas como os dicionários electrónicos ou o processamento de linguagem natural. No entanto, nos últimos anos tem-se assistido a um número crescente de aplicações, derivando dessa mesma teoria formal da linguagem, em distintas áreas de engenharia como sejam, entre outras, o reconhecimento de padrões, a recuperação de informação, as linguagens gráficas, a comunicação homem-máquina e a inteligência artificial (Fu e Booth, 1975a).

Tanto as linguagens naturais, como as linguagens de programação, como inclusive outras que ainda nem sequer tomámos consciência da sua existência, podem ser vistas como conjuntos de palavras, isto é, conjuntos de *strings* finitas de elementos de um mesmo vocabulário básico.

Tradicionalmente a teoria formal da linguagem preocupa-se mais com a especificação sintáctica de uma linguagem e menos com os seus aspectos semânticos. Quando estamos em presença de linguagens que possibilitam a existência de um número infinito de palavras, torna-se impossível listá-las todas. Neste contexto, o objectivo da teoria formal da linguagem é estabelecer especificações finitas para linguagens possivelmente infinitas.

Em presença de uma amostra finita e reduzida de uma dada linguagem reveste-se de particular importância a possibilidade de identificar uma das gramáticas que a represente, recorrendo a esse conjunto finito de exemplos. Este processo, designado por inferência gramatical, revela-se de extrema importância quando se assume que as linguagens provêm de sistemas físicos.

## 2.2 CONCEITOS FUNDAMENTAIS

Uma linguagem, na forma em que é vulgarmente entendida, pode ser definida como um conjunto de sinais típicos perceptíveis por um dado grupo de humanos. Deste modo, uma possível caracterização pode socorrer-se de duas importantes propriedades:

- existe nas formas escrita e falada;
- é criativa, no sentido de um utilizador nativo poder gerar e compreender um conjunto infinito de frases, mesmo aquelas que não tenham sido utilizadas até à presente data.

Uma gramática pode assim ser definida como uma ferramenta cujo objectivo é descrever uma determinada linguagem. Deste modo uma linguagem descrita por uma determinada gramática corresponde ao conjunto de todas as frases geradas por essa mesma gramática. Neste contexto uma gramática é constituída por um conjunto de regras que possibilitam a um dado grupo de humanos exprimirem as suas ideias através de frases, mesmo que não se apercebam que estão a utilizar esse conjunto de regras.

Como já referido, o principal objectivo da teoria formal da linguagem é o estudo das especificações finitas referentes a linguagens possivelmente infinitas. De seguida apresentam-se alguns conceitos fundamentais à teoria formal da linguagem, necessários ao entendimento dos capítulos subsequentes.

Um alfabeto é um conjunto finito, não vazio, representado por  $\Sigma$ , cujos elementos são geralmente referidos como símbolos, letras ou caracteres.

Uma palavra de um dado alfabeto, representada por  $z$ , é uma *string* finita constituída por zero ou mais símbolos do alfabeto  $\Sigma$ , na qual o mesmo símbolo pode ocorrer várias vezes. A palavra constituída por zero símbolos é designada por palavra vazia e representada por  $\lambda$ .

O conjunto de todas as palavras de um alfabeto  $\Sigma$  é representado por  $\Sigma^*$ . O conjunto de todas as palavras não vazias de um dado alfabeto  $\Sigma$ , é representado por  $\Sigma^+$ . Ambos os conjuntos  $\Sigma^*$  e  $\Sigma^+$  podem ser infinitos para qualquer alfabeto  $\Sigma$ . Por outras palavras, o

conjunto  $\Sigma^*$ , os seus elementos e as *strings* de seus elementos podem ser referenciados, respectivamente, como vocabulário, palavras e frases.

A concatenação de duas palavras –  $z_1$  e  $z_2$  – representa-se por  $z_1z_2$ . A palavra vazia funciona como elemento neutro da concatenação –  $z\lambda = \lambda z = z$ . Sendo a concatenação associativa, é possível utilizar a notação  $z^i$  referente à concatenação sucessiva da mesma palavra, onde  $i$  é um número inteiro positivo.

O comprimento de uma palavra, representado por  $|z|$ , corresponde ao número de símbolos presentes nessa mesma palavra, sendo que cada símbolo é contado o número de vezes que aparece. Por definição a palavra vazia apresenta um comprimento nulo, isto é,  $|\lambda|=0$ .

O comprimento de uma palavra goza das seguintes propriedades:

$$\begin{cases} |z_1z_2| = |z_1| + |z_2| \\ |z^i| = i|z| \end{cases}, \text{ para qualquer } z_j \text{ e } i \geq 0 \quad (2.1)$$

A noção de gramática foi inicialmente introduzida para aplicações em linguagens naturais. Esta ideia foi originalmente introduzida por Noam Chomsky (Chomsky, 1959) com o objectivo de descrever linguagens naturais. No entanto as gramáticas também podem descrever conjuntos de palavras (linguagens formais) reconhecidas por vários tipos de máquinas de computação, como veremos mais adiante.

Uma gramática é definida como um quádruplo ( 2.2 ) onde,  $R = (\Sigma, P)$  é um par denominado sistema de rescrita,  $\Sigma_N$  é um alfabeto não terminal,  $\Sigma_T$  é um alfabeto terminal,  $S \in \Sigma_N$  é um símbolo inicial e  $\Sigma = \Sigma_N \cup \Sigma_T$  é um alfabeto completo.

$$G = (R, \Sigma_N, \Sigma_T, S) \quad (2.2)$$

$P$  é um conjunto de produções, na forma  $\alpha \rightarrow \beta$ , onde  $\alpha$  e  $\beta$  são *strings*, sendo que  $\alpha$  contem pelo menos um símbolo não terminal (isto é, pertencente ao alfabeto não terminal).

Os alfabetos terminal e não terminal são necessariamente disjuntos,  $\Sigma_N \cap \Sigma_T = \emptyset$ .

Por outras palavras, é possível afirmar que uma gramática é constituída por um alfabeto terminal cujos símbolos constituem as palavras resultantes, por um alfabeto não terminal utilizado para gerar padrões, por um símbolo não terminal especial denominado “símbolo inicial”, e por um conjunto de produções as quais determinam a geração de padrões.

Deste modo uma linguagem  $L$ , definida por ( 2.3 ) e gerada por uma gramática  $G$ , entende-se como o conjunto de todas as palavras no alfabeto terminal que podem ser derivadas do símbolo inicial por aplicação de uma dada sequência de produções pertencentes ao sistema de rescrita.

$$L(G) = \left\{ z \in \Sigma_T^* \mid S \xRightarrow{R} z \right\} \quad (2.3)$$

Duas gramáticas dizem-se equivalentes se ambas gerarem a mesma linguagem. Considerando as linguagens como conjuntos, as operações booleanas habituais – união, intersecção, complemento e diferença – são definidas normalmente. A concatenação (ou produto) de duas linguagens  $L_1$  e  $L_2$  é definida como

$$L_{12} = L_1 L_2 = \left\{ z_1 z_2 \mid z_1 \in L_1 \text{ e } z_2 \in L_2 \right\} \quad (2.4)$$

Como exemplo típico de uma linguagem considere-se a gramática constituída pelos seguintes elementos:

$$\begin{aligned} \Sigma_N &= \{S, A, B\}; \\ \Sigma_T &= \{a, b\}; \\ P &= \{S \rightarrow AB, B \rightarrow bB, A \rightarrow a, B \rightarrow b\}. \end{aligned}$$

Por forma a obter palavras terminais, a aplicação de produções será desencadeada a partir do símbolo inicial. Sucessivamente serão aplicadas quaisquer das produções, presentes em  $P$ , até existirem apenas símbolos terminais. Aplicando sucessivamente as produções  $S \rightarrow AB$ ,  $A \rightarrow a$ ,  $B \rightarrow bB$ ,  $B \rightarrow bB$ ,  $B \rightarrow bB$  e  $B \rightarrow b$ , obtêm-se as sequências  $AB$ ,  $aB$ ,  $abB$ ,  $abbB$ ,  $abbbB$ , gerando desta forma a palavra ‘abbbb’. Facilmente se pode verificar que a linguagem gerada pela gramática anterior corresponde ao conjunto de

todas as palavras constituídas por um símbolo ‘a’ seguido de um número arbitrário de símbolos ‘b’.

Na Tabela 2.I apresenta-se a hierarquia de gramáticas estabelecida por Noam Chomsky, em função das restrições aplicáveis às suas produções. A ordem hierárquica é considerada no sentido de que uma gramática de tipo  $n+1$  é também uma gramática de tipo  $n$ .

**Tabela 2.I. Hierarquia de gramáticas estabelecida por Chomsky.**

<b>Tipo</b>	<b>Designação</b>	<b>Produções (<math>\alpha \rightarrow \beta</math>)</b>
0	Frase Estruturada	‘ $\alpha$ ’ possui um símbolo não terminal
1	Sensível ao Contexto	‘ $\alpha$ ’ possui um símbolo não terminal e o comprimento de ‘ $\alpha$ ’ é menor ou igual ao comprimento de ‘ $\beta$ ’
2	Livre de Contexto	$\alpha$ é um único não terminal
3	Regular	as produções são da forma $\alpha \rightarrow a\beta$ , $\alpha \rightarrow \beta a$ ou $\alpha \rightarrow a$ , onde ‘ $\alpha$ ’ e ‘ $\beta$ ’ são símbolos não terminais únicos e ‘a’ é um símbolo terminal ou ‘ $\lambda$ ’

## 2.3 INFERÊNCIA GRAMATICAL

O processo de inferência gramatical consiste em identificar o mais correctamente possível uma gramática que represente a linguagem em análise, a partir de um número finito de exemplos dessa mesma linguagem. Por outras palavras, consiste na aprendizagem automática de gramáticas formais a partir de um subconjunto finito de exemplos provenientes de uma linguagem que se pretende representar gramaticalmente.

Em 1967, Gold (Gold, 1967) estabeleceu o conceito de identificação no limite. Assuma-se que é possível fornecer uma sequência de informação completa  $I(L(G))$  a um determinado algoritmo de inferência. Dada uma gramática  $G$  e uma linguagem por ela gerada  $L(G)$ ,  $I(L(G))$  representa uma sequência temporalmente ordenada de informação sobre todas as palavras de  $L(G)$  – informação positiva – e de  $\Sigma^* - L(G)$  – informação negativa.  $I(L(G))$  pode ser entendida como uma sequência infinita de pares ordenados  $(z, I)$ , de  $\Sigma^* \times \{0,1\}$ , tal que  $I=1$  somente se  $z$  for gerada por  $G$ , e cada palavra de  $\Sigma^*$  aparece pelo menos uma vez como primeiro elemento de algum par ordenado na sequência  $I(L(G))$ .

Uma classe de linguagens  $C$  é dita ‘identificável no limite’ caso exista um algoritmo de inferência, tal que para cada gramática  $G \in C$  e para qualquer sequência de informação completa  $I(L(G))$  existe um  $\tau$  tal que

$$\forall t > \tau \quad \begin{aligned} G_t &= G_\tau \\ L(G_t) &= L(G_\tau) \end{aligned} \quad (2.5)$$

onde  $G$  resulta da aplicação do algoritmo de inferência, com os dados referentes às primeiras palavras de  $I(L(G))$ .

Quando se trabalha com amostras finitas, de linguagens desconhecidas e possivelmente infinitas, não é possível verificar as propriedades de convergência. Deste modo o binómio generalização/exactidão terá de ser avaliado conforme o objectivo. Conforme a aplicação, poderá ser mais importante inferir uma gramática que, mesmo com alguns erros de exactidão, gere uma linguagem que represente de uma forma global toda a

linguagem original; ou então inferir uma gramática local que represente bem um subconjunto da linguagem sem a preocupação de representar os restantes elementos dessa linguagem.

O aspecto da generalização subjacente à primeira das duas opções anteriores encontra-se relacionado com a qualidade dos exemplos da linguagem apresentadas na sequência de informação positiva. Será importante que estes se distribuam uniformemente por  $L(G)$  e não sejam condensados num seu subconjunto restrito.

Tentando evitar o anterior paradigma, Angluin (Angluin, 1988) estabeleceu um algoritmo de aprendizagem onde um ‘professor’ responde a perguntas sobre a gramática a inferir. Tipicamente, estas referem-se a perguntas sobre ‘pertença’ (uma dada palavra é ou não gerada pela gramática a inferir) ou de ‘equivalência’ (uma dada gramática é ou não equivalente à gramática a inferir). Deste modo uma gramática correcta pode ser inferida em tempo finito.

Em 1984 Valiant (Valiant, 1984) estabeleceu o conceito de *probably approximately correct learning (PAC learning)*, no qual a aprendizagem é feita a partir de exemplos aleatórios provenientes do domínio  $\Sigma^*$ , com uma distribuição de probabilidade  $DP$  possivelmente arbitrária e desconhecida. O algoritmo de inferência gerará uma gramática com sucesso caso o seu erro – definido como a soma das probabilidades  $DP(z)$  das palavras  $z$  na diferença simétrica da linguagem gerada pela gramática inferida e da linguagem a inferir, com respeito a  $DP$  – seja menor que um dado parâmetro de precisão.

Este paradigma de inferência é aquele que se encontra mais próximo do utilizado neste trabalho. Ao inferir uma gramática responsável pela linguagem gerada por um dado sistema dinâmico possuímos apenas informação positiva, além de que esta se refere apenas a uma amostra, que se pretende o mais representativa possível, do domínio de funcionamento do referido sistema dinâmico.

A eficiência de um algoritmo de inferência encontra-se dependente da relação entre o número de exemplos disponíveis e o tamanho da gramática a inferir. O número de exemplos, na forma de conjunto de palavras, corresponde ao comprimento dessas palavras. O tamanho da gramática a inferir pode ser assumido como o número de

produções inferidas. Notar que um elevado número de palavras exemplo pode ser de fraca qualidade pelo facto de estas serem todas semelhantes, fazendo uso essencialmente das mesmas produções.



# CAPÍTULO 3 - INFERÊNCIA GRAMATICAL DE UM SISTEMA DINÂMICO COMANDADO

**Resumo:** Neste capítulo apresenta-se o processo de estabelecimento de gramáticas, representativas de linguagens geradas por sistemas dinâmicos comandados, através de um mecanismo de inferência gramatical. Verificar-se-á que a organização subjacente aos sistemas dinâmicos comandados conduz naturalmente à formulação de gramáticas dependentes do contexto. São ainda estudadas várias propriedades exibidas pelas gramáticas quando representam sistemas dinâmicos.



### **3.1 INTRODUÇÃO**

A teoria formal da linguagem tem sido utilizada, no passado, para estudar sistemas autónomos, sendo que a complexidade das sequências geradas, de acordo com uma dada codificação, é utilizada para medir a complexidade do sistema dinâmico (Troll, 1993).

Contudo, a maioria dos sistemas utilizados em aplicações tecnológicas, nomeadamente no domínio dos sistemas de accionamento electromecânico, não são sistemas dinâmicos autónomos mas sim comandados. Deste modo, torna-se necessário distinguir, desde o início, entre a dinâmica gerada pelo sistema e aquela que depende da vontade do utilizador. Isto conduz a algumas implicações ao nível da natureza das entidades formais que são utilizadas para codificar as variáveis do sistema, bem como da natureza das linguagens geradas.

Outro aspecto importante quando se consideram aplicações práticas, prende-se com a possibilidade de extrair gramáticas através de um processo de aprendizagem por recurso a dados gerados pelo próprio sistema, independentemente de se conhecer ou não um modelo analítico do mesmo.

A impossibilidade de obter exemplos que representem a totalidade da linguagem gerada por sistemas tecnológicos conduz ao desenvolvimento de mecanismos de interpolação gramatical que se revelam fundamentais no aumento das capacidades de generalização apresentadas pela gramática.

Neste capítulo assumem-se os sistemas dinâmicos como entidades capazes de gerar uma linguagem possível de ser descrita por uma dada gramática. Esta gramática é obtida mediante um processo de aprendizagem por exemplos. Apresentam-se assim os algoritmos desenvolvidos para extrair essas gramáticas capazes de descrever a linguagem gerada por um dado sistema dinâmico. Serão considerados alguns exemplos simples, com o objectivo de estudar as principais propriedades dos algoritmos considerados.

Na sequência da metodologia seguida, os algoritmos estudados apresentam-se como ferramentas importantes na identificação de sistemas não lineares, onde a influência da informação passada pode variar conforme o contexto, isto é, depende de quão passada é essa informação.

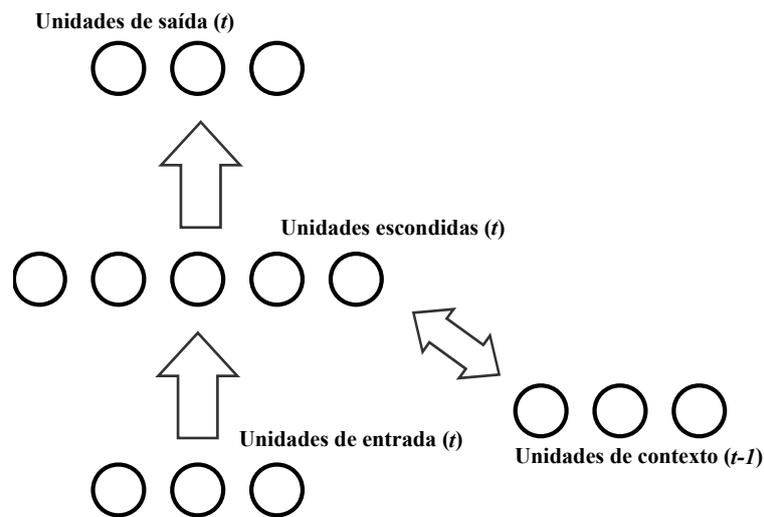
## **3.2 AS LINGUAGENS E OS SISTEMAS DINÂMICOS**

Desde que S. Wolfram (Wolfram, 1984) utilizou a Teoria Formal da Linguagem nos seus estudos sobre autómatos, a associação entre linguagens formais e sistemas dinâmicos tem sido objecto de estudo. Vários autores (Auerbach e Procaccia, 1990) (Wang e Xie, 1994) (Collet, Galves e Lopes, 1995) (Lakdawala, 1996) utilizam a noção de gramática para medir a complexidade de vários tipos de sistemas dinâmicos autónomos.

As linguagens podem assim ser entendidas como sistemas dinâmicos, ao invés de serem consideradas apenas como um conjunto de produções e símbolos (Elman, 1995). Os símbolos são vistos como regiões no espaço de estados e as produções como uma dinâmica embebida no sistema.

Uma abordagem semelhante pode ser encontrada em (Martins e Vilela Mendes, 2001) no que se refere a elementos da lógica formal. Neste trabalho, apresentado em apêndice, é estabelecida uma correspondência entre os elementos dos sistemas de raciocínio lógico e as operações dinâmicas das redes neuronais. Esta correspondência é estabelecida como um dicionário tradutor entre as formulações simbólica e conexionista. Apesar de os elementos da lógica formal e das gramáticas não serem equivalentes existem trabalhos que apontam certas relações de equivalência entre eles. Este aspecto reforça a noção de equivalência entre gramáticas e sistemas dinâmicos.

Na área de conhecimento das redes neuronais, tem sido demonstrado o potencial de codificar as propriedades temporais de sequências de entradas oferecido pelas redes neuronais recorrentes (Hertz, Krogh e Palmer, 1991). Vários autores (Elman, 1990) (Omlin e Giles, 1994) (Sakakibara e Golea, 1995) utilizam redes neuronais recorrentes – Figura 3.1 – para ‘capturar’ as produções gramaticais de uma dada linguagem regular, reforçando assim o carácter de sistema dinâmico dessa mesma linguagem.



**Figura 3.1. Rede Neuronal Recorrente no instante  $t$ .**

Nesta abordagem as unidades de contexto retêm uma cópia das ativações das unidades escondidas no instante de tempo anterior. Assim, no próximo instante de tempo o estado das unidades escondidas é determinado pelas entradas e pelo estado das unidades de contexto. Esta estrutura dinâmica funciona como uma máquina de estados finitos (representável por um autômato do tipo finito), representando a rede neuronal a função de transição de estado. Estas arquiteturas de redes desempenham as mesmas funções que um autômato do tipo finito e assim podem inferir gramáticas do tipo regular, representando gramáticas de uma forma não gramatical, isto é, não recorrendo a um sistema de rescrita.

Neste capítulo propõe-se o processo inverso: partir de um sistema dinâmico e construir uma gramática representativa de uma eventual linguagem que o sistema gere. Desenvolve-se um algoritmo que, adaptado às características funcionais de um sistema dinâmico, extrai a gramática correspondente a uma linguagem representativa desse mesmo sistema. O sistema dinâmico pode ser não estacionário, pois as características de aprendizagem evolutiva do algoritmo adaptam-se bem a variações na dinâmica estrutural do sistema.

### **3.3 CONCEITO DE APRENDIZAGEM**

Em psicologia o conceito de aprendizagem refere-se à formação de conceitos, pensamentos e imagens, bem como à relação entre a experiência e a organização da mente.

O ser humano associa a noção de aprendizagem às suas capacidades de aquisição de conhecimento e mudança. Perante uma experiência concreta o ser humano altera os seus conceitos, hábitos ou certezas por forma a obter melhores desempenhos numa futura repetição da mesma experiência.

A noção de inteligência presente num qualquer algoritmo encontra-se directamente associada à sua capacidade de aprendizagem e adaptação automática e autónoma a sistemas não totalmente conhecidos e sujeitos a variações.

Como já visto, quando se analisa uma linguagem desconhecida o processo de inferência gramatical consiste em identificar uma gramática que gere essa linguagem. Os esforços desenvolvidos nos processos de inferência e *machine learning* têm como objectivo atingir algoritmos que reduzam o trabalho humano de produzir descrições sintácticas ou estruturais de classes, através do tratamento de exemplos. Em certos casos um algoritmo de inferência consegue ‘descobrir’ estruturas menos óbvias ou mais complexas do que conseguiria um ser humano.

Este último aspecto encontra-se directamente relacionado com a codificação da informação fornecida pelo meio. A forma como a codificação da informação é efectuada (isto é, qual o alfabeto escolhido para representar essa informação disponível) condiciona a forma da gramática inferida.

Tome-se como exemplo um qualquer sistema dinâmico. A forma mais usual de o representar é através de um conjunto de equações diferenciais ( 3.1 ), que determinam a evolução das suas variáveis. O vector das variáveis de estado é representado por  $\mathbf{x}$ , o vector das variáveis de comando por  $\mathbf{u}$ , o vector das variáveis de saída por  $\mathbf{y}$  e o vector

das condições iniciais por  $\mathbf{x}_0$ . As funções  $\Phi(\cdot)$  e  $\Psi(\cdot)$  podem ser quaisquer funções não lineares.

$$\begin{cases} \dot{\mathbf{x}}(t) = \Phi(\mathbf{x}(t), \mathbf{u}(t)) \\ \mathbf{y}(t) = \Psi(\mathbf{x}(t), \mathbf{u}(t)) \end{cases}, \quad \mathbf{x}_0 \quad (3.1)$$

A informação disponível pode assumir-se como o conhecimento da evolução das variáveis do referido sistema. No entanto, a codificação que é feita dessa informação pode ser bastante variada. É possível considerar uma codificação contínua, discreta, ou mesmo *fuzzy* das suas variáveis. É ainda possível considerar a sua evolução no espaço de estados ou codificar essa informação com outros atributos, como sejam a estacionariedade das variáveis, a sua tendência crescente ou decrescente, ou a correlação entre elas. A codificação (alfabeto) considerada originará diferentes linguagens representadas por diferentes gramáticas. No entanto todas as gramáticas terão como entidade geradora da linguagem o mesmo sistema dinâmico. As produções inferidas a partir de um dado conjunto de exemplos vão assim ser mais ou menos óbvias para um ser humano em função do alfabeto escolhido para descrever o sistema dinâmico.

Outro aspecto importante, e que condiciona ou influencia a aprendizagem, relaciona-se com as características da informação disponível, independentemente da codificação adoptada. Assim, em função das características da informação disponível referente a um dado processo, os problemas relativos à modelização e controlo de sistemas podem ser classificados em três categorias: determinísticos, aleatórios e adaptativos.

Nos processos determinísticos é assumido o conhecimento completo de todo o processo e perturbações exteriores. Apesar de simplificações habitualmente consideradas a modelização e controlo de sistemas não lineares torna-se difícil, mesmo para sistemas simples, devido a problemas analíticos e computacionais.

Nos processos aleatórios, o processo bem como as perturbações são consideradas como aleatórias. As características probabilísticas de ambos são assumidas como dadas em termos das suas distribuições relevantes. Os problemas de optimização e procedimentos numéricos são contudo semelhantes aos considerados nos processos determinísticos.

Nos processos adaptativos considera-se um conhecimento mínimo acerca do sistema e das perturbações. Através da observação do processo, à medida que este evolui, estes sistemas vão adquirindo mais conhecimento com vista a melhorar os procedimentos de optimização. O conceito de aprendizagem considerado neste capítulo integra-se neste tipo de processos, onde o conceito de adaptativo se traduz no de aprendizagem.

Um sistema adaptativo com capacidade de aprendizagem tem, em cada instante, de decidir entre investigar novas características do meio por forma a melhorar o seu conhecimento (e assim tomar melhores decisões no futuro), ou tomar a melhor decisão baseado apenas no seu conhecimento passado. O dilema reside no facto de que a melhoria do conhecimento envolve recursos adicionais e pode resultar apenas em pequenos benefícios marginais. Este conflito é conhecido como a dualidade identificação/controlo.

Retomando o exemplo de um sistema dinâmico, a evolução do seu estado interno pode ser determinada pela resolução da equação diferencial ( 3.1 ), em presença de um determinado comando. O conhecimento das condições iniciais e da evolução do comando, aplicado ao sistema, determina univocamente a evolução do estado do mesmo. É deste modo possível estabelecer uma relação funcional ( 3.2 ) entre as variáveis de estado e o comando, com representação num dado espaço.

$$\mathbf{y} = \varphi_{\mathbf{x}_0, \mathbf{p}}(\mathbf{x}, \mathbf{u}) \quad ( 3.2 )$$

Para um mesmo conjunto de condições iniciais –  $\mathbf{x}_0$  – e de parâmetros do sistema –  $\mathbf{p}$  – a relação funcional ( 3.2 ) é representativa do comportamento dinâmico do sistema.

Assumindo uma qualquer codificação associada às variáveis do sistema dinâmico e considerando-se este responsável pela geração de uma determinada linguagem, existem alguns problemas a que o processo de inferência de uma gramática pode estar sujeito. A própria relação funcional anterior ( 3.2 ) não possui garantia de existência em todo o

domínio considerado, além de que pode também sofrer alterações em função das variações sofridas quer pelas condições iniciais quer pelos parâmetros do sistema; pelo que a linguagem gerada pelo sistema pode sofrer alterações à medida que o próprio sistema evolui. O aparecimento de novas palavras ou desaparecimento de outras traduz-se em alterações no sistema de rescrita da gramática que representa essa linguagem. Assim é importante que o processo de aprendizagem seja dinâmico, por forma a que a gramática inferida acompanhe eventuais evoluções da linguagem.

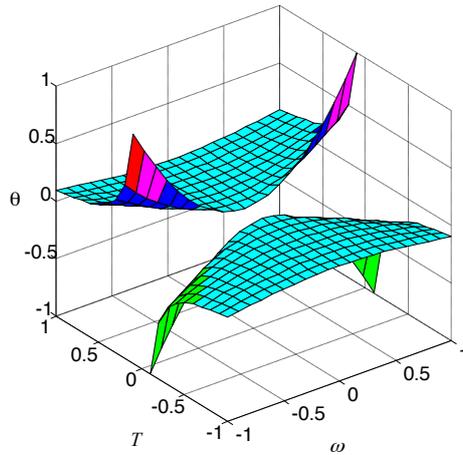
Considere-se, como exemplo, um sistema inercial ( 3.3 ), onde  $\theta_i$  representa a posição,  $\omega_i$  a velocidade,  $T_i$  o binário aplicado e  $J_i$  o momento de inércia.

$$\begin{cases} \frac{d\theta_i}{dt} = \omega_i \\ \frac{d\omega_i}{dt} = \frac{1}{J_i} T_i \end{cases} \quad ( 3.3 )$$

A evolução do estado interno do sistema pode ser determinada por resolução da equação diferencial anterior, em presença de um determinado comando. O conhecimento das condições iniciais ( $\theta_{i0}$  e  $\omega_{i0}$ ) e da evolução do comando, aplicado ao sistema, determina univocamente a evolução do estado do mesmo. É deste modo possível estabelecer uma relação funcional entre as variáveis de estado e o comando ( 3.4 ), com representação num espaço  $[\theta_i, \omega_i, T_i]$  apresentado na Figura 3.2.

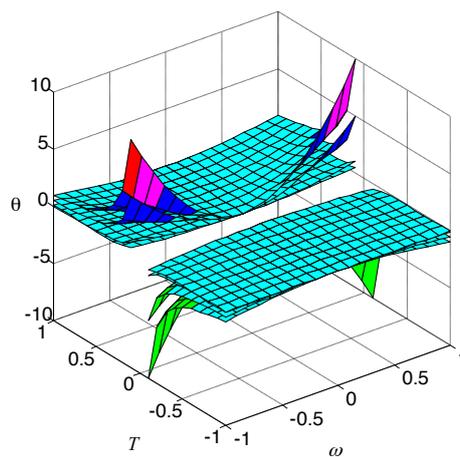
$$\theta_i = \frac{J_i}{2T_i} (\omega_i^2 - \omega_{i0}^2) + \theta_{i0} \quad ( 3.4 )$$

Para um mesmo conjunto de condições iniciais e de parâmetros do sistema a relação funcional ( 3.4 ) é representativa do comportamento dinâmico do sistema inercial ( 3.3 ).

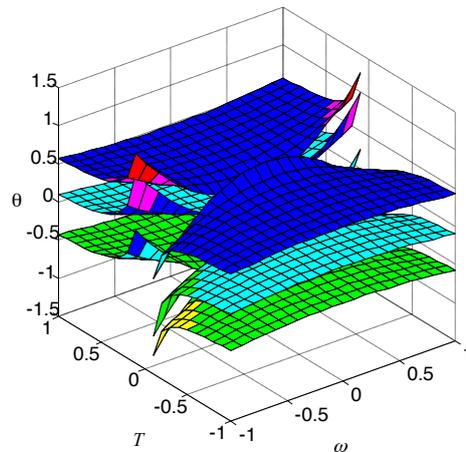


**Figura 3.2. Representação da relação funcional ( 3.4 ) para  $\theta_{i0} = \omega_{i0} = 0$ .**

Este exemplo é representativo de alguns dos problemas a que o processo de extracção de uma relação funcional pode estar sujeito. Esta relação não possui garantia de existência em todo o domínio considerado, como se pode observar na Figura 3.2, onde no plano definido por um binário nulo a posição se apresenta indefinida. Por outro lado a própria relação funcional pode sofrer alterações, em função das variações sofridas quer pelas condições iniciais quer pelos parâmetros do sistema. Nas Figura 3.3 e Figura 3.4 representam-se as variações sofridas pela relação funcional devidas aos condicionalismos referidos.



**Figura 3.3. Representação da relação funcional ( 3.4 ) em função da variação do parâmetro  $J_i$ .**



**Figura 3.4. Representação da relação funcional ( 3.4 ) em função da variação das condições iniciais.**

Um maior momento de inércia aumenta a curvatura exibida pelos planos definidos pela relação funcional ( 3.4 ), e diferentes condições iniciais provocam o deslocamento desses mesmos planos. Assim é importante que o processo de aprendizagem seja efectuado de uma forma contínua, por forma a que a relação funcional inferida acompanhe as eventuais variações exibidas pelo sistema dinâmico.

### **3.4 APRENDIZAGEM POR EXEMPLOS**

Para um qualquer sistema dinâmico entende-se por aprendizagem a extracção, baseada num conjunto de exemplos, de uma relação funcional ( 3.5 ) representativa do funcionamento do sistema em causa (Dente, 1994). Nesta relação, geralmente não conhecida e sem garantia de existência, assume-se a saída do sistema -  $y$  - como função de um conjunto de variáveis consideradas significativas -  $X$ . O conjunto de exemplos apresentados ao algoritmo de aprendizagem é constituído por um número limitado de pares ordenados ( 3.6 ), formados respectivamente pela saída e pelas variáveis das quais esta dependerá.

$$y = f(X) \quad ( 3.5 )$$

$$E_X = \{(y_1, X_1), (y_2, X_2), \dots, (y_n, X_n)\} \quad ( 3.6 )$$

Este processo é em tudo idêntico à inferência gramatical de uma linguagem, recorrendo apenas a informação positiva, onde a aprendizagem é efectuada a partir de exemplos (palavras) provenientes de  $\Sigma^*$ .

O algoritmo de aprendizagem ideal, vulgarmente designado por ‘caixa preta’, não necessitaria de nenhum conhecimento prévio sobre o sistema e limitar-se-ia a efectuar uma recolha dos dados disponíveis, por forma a extrair uma eventual relação funcional que os relacionasse entre si. Obviamente que tal algoritmo apresenta-se como um objectivo muito difícil de atingir para a generalidade dos sistemas, pois a complexidade destes obriga a um estudo prévio dos mesmos para facilitar o processo de aprendizagem, através de uma escolha criteriosa do conjunto de variáveis significativas.

Assim o processo de aprendizagem propriamente dito deverá ser precedido de uma fase prévia de estudo do sistema dinâmico, das variáveis envolvidas e suas inter-relações e do seu domínio de funcionamento, de modo a definir as variáveis significativas, bem como a sua codificação, por forma a estabelecer o alfabeto da linguagem.

Este estudo prévio é de extrema importância na implementação de algoritmos de aprendizagem por exemplos. Uma má especificação dos fenómenos físicos presentes, com a consequente escolha errada de variáveis, ou uma má codificação das mesmas, compromete todo o processo de aprendizagem, tornando-o impossível ou extremamente complexo. A escolha de um conjunto de variáveis não correlacionadas entre si implica a não existência de uma relação funcional com a consequente impossibilidade de aprendizagem. A escolha de um conjunto de variáveis extremamente vasto, com a eventual inclusão de variáveis não influentes no processo, dificulta o processo de aprendizagem tornando-o extremamente moroso. Uma codificação muito específica, que origine um alfabeto com um elevado número de símbolos, pode conduzir a uma gramática com um grande número de produções, tornando-a extremamente complexa.

Para que o algoritmo de aprendizagem através de exemplos apresente sucesso deverá possuir importantes características de memorização, generalização, robustez e aprendizagem evolutiva.

A capacidade de memorização advém do facto de a linguagem gerada pelo sistema dinâmico ser extraída por recurso a um conjunto limitado de exemplos, os quais terão de ser retidos.

A partir dos exemplos memorizados o algoritmo deverá ser capaz de aproximar a linguagem que originou esses exemplos. A capacidade de generalização advém do facto de a gramática inferida ser uma boa representação da linguagem não apenas para os referidos exemplos mas também para outros distintos, não considerados na aprendizagem.

As boas características de robustez traduzem-se na insensibilidade ao ruído presente na aquisição das várias variáveis, na insensibilidade à presença de variáveis não influentes no processo, e na capacidade de rejeição de eventuais maus exemplos na fase de aprendizagem.

A capacidade de aprendizagem evolutiva, em funcionamento estacionário, proporciona boa capacidade de adaptação a diferentes condições de funcionamento – novas zonas do domínio de funcionamento e eventuais alterações de parâmetros. Tal como já referido, um aspecto muito importante na aprendizagem relaciona-se com eventuais variações da

relação funcional ao longo do tempo. O que em determinado instante de tempo é tido como verdadeiro poderá já não o ser num instante de tempo posterior. Tal obriga à necessidade de efectuar a aprendizagem em funcionamento estacionário, por forma a manter a gramática o mais actualizada possível em função das novas situações que vão surgindo.

Esta aprendizagem evolutiva deverá ser efectuada de uma forma autónoma e automática. O algoritmo usará os dados recolhidos em tempo real, durante o processo para o qual foi projectado, como informação relevante para a aprendizagem, devendo esta decorrer em paralelo com o processo.

O processo de aprendizagem inerente a estes algoritmos de aprendizagem é algo semelhante à evolução cognitiva de um ser humano. Inicialmente é possível extrair um conjunto de produções tomando por base dados teóricos mais ou menos precisos, sendo tal conhecimento entendido como o comportamento instintivo genético, próprio da espécie e presente à nascença. A obtenção de uma nova gramática, agora com recurso a dados experimentais, corresponde à fase escolar de um indivíduo, onde este apreende os conhecimentos necessários a futuros desempenhos. A capacidade de aprendizagem evolutiva tem por paralelo a fase pós-escolar da vida de qualquer indivíduo, onde este, mediante as suas capacidades de aprendizagem, se adapta a novas situações (potenciando uma certa economia na memória genética).



## **3.5 ALGORITMO DE APRENDIZAGEM ATRAVÉS DE EXEMPLOS**

Pretende-se nesta secção estabelecer as características do algoritmo de inferência gramatical responsável por estabelecer a gramática que descreve a linguagem gerada por um dado sistema dinâmico.

### **3.5.1 Formalização da gramática**

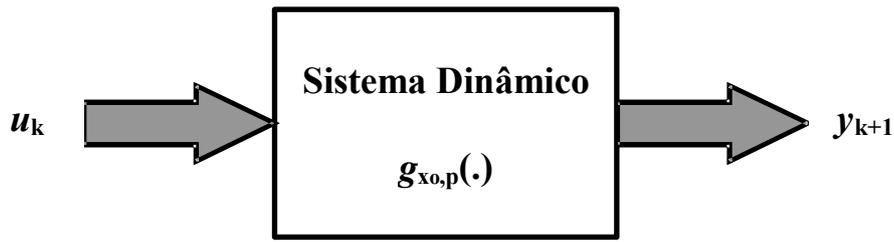
A geração de variáveis de comando e a aquisição experimental de variáveis de estado e saída é geralmente baseada em sistemas digitais. Considerando uma frequência de amostragem fixa, o sistema dinâmico será descrito por ( 3.7 ), onde  $f(\cdot)$  e  $h(\cdot)$  podem ser funções não lineares.

$$\begin{cases} \mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k) \\ \mathbf{y}_k = h(\mathbf{x}_k, \mathbf{u}_k) \end{cases}, \quad \mathbf{x}_0 \quad (3.7)$$

O sistema assume-se como estável, com parametrização conhecida, mas com os valores dos parâmetros não totalmente conhecidos. Uma possível relação funcional representativa da dinâmica do sistema será dada pela expressão ( 3.8 ), sendo as relações entre as variáveis de comando e de estado impostas por ( 3.7 ). Atendendo à não linearidade do modelo de estado, bem como à incerteza no conhecimento de alguns parâmetros, determinar com exactidão esta relação funcional apresenta-se como uma tarefa difícil.

$$\mathbf{y}_{k+1} = \mathcal{G}_{\mathbf{x}_0, \mathbf{p}}(\mathbf{x}_k, \mathbf{u}_k) \quad (3.8)$$

É apresentada na Figura 3.5 apresenta-se uma representação esquemática desta perspectiva do sistema dinâmico, onde  $\mathbf{x}_0$  representa o valor inicial das variáveis de estado e  $\mathbf{p}$  representa o conjunto dos parâmetros do sistema.



**Figura 3.5. Sistema Dinâmico.**

Temos assim uma descrição do sistema dinâmico na forma de espaço de estados e não na forma espaço-tempo. No entanto, a relação funcional anterior, que depende do estado inicial e dos parâmetros do sistema, permite obter a evolução temporal da saída em função da evolução temporal do comando.

Para um sistema dinâmico não linear, ou para um sistema dinâmico em que não se conhecem todas as variáveis de estado, a evolução da variável de saída não é determinada univocamente pelo conhecimento da variável de comando e das variáveis de estado acessíveis (Martins, Pires e Dente, 1997). Além de depender das variáveis de comando e de estado (conhecidas ou não), a evolução da variável de saída irá também depender da evolução passada das variáveis de estado acessíveis.

Sem perda de generalidade, assumindo que o vector de saída têm dimensão unitária e que esta apenas depende de variáveis de estado acessíveis, a expressão ( 3.7 ) pode ser reescrita como ( 3.9 ), onde  $\mathbf{x}_1$  representa as variáveis de estado acessíveis e  $\mathbf{x}_2$  as variáveis de estado não acessíveis.  $f_1(\cdot)$ ,  $f_2(\cdot)$  e  $h(\cdot)$  continuam a ser funções não lineares definidas como  $f_1: \mathfrak{R}^l \times \mathfrak{R}^m \times \mathfrak{R}^p \rightarrow \mathfrak{R}^l$ ,  $f_2: \mathfrak{R}^l \times \mathfrak{R}^m \times \mathfrak{R}^p \rightarrow \mathfrak{R}^m$  e  $h: \mathfrak{R}^l \rightarrow \mathfrak{R}$ , onde  $l$  representa o número de variáveis de estado acessíveis,  $m$  o número de variáveis de estado não acessíveis e  $p$  o número de variáveis de comando.

$$\begin{cases} \mathbf{x}_{1_{k+1}} = f_1(\mathbf{x}_{1_{k+1}}, \mathbf{x}_{2_{k+1}}, \mathbf{u}_k) \\ \mathbf{x}_{2_{k+1}} = f_2(\mathbf{x}_{1_{k+1}}, \mathbf{x}_{2_{k+1}}, \mathbf{u}_k) \\ y_k = h(\mathbf{x}_{1_k}) \end{cases} \quad ( 3.9 )$$

De ( 3.9 ) é desejável obter uma relação funcional que represente a dinâmica do sistema envolvendo apenas variáveis acessíveis. Assim é necessário considerar as  $m$  anteriores observações da variável de saída ( 3.10 ).

$$\left\{ \begin{array}{l} y_{k-(m-1)} = h(\mathbf{x}_{1_{k-(m-1)}}) = h(f_1(\mathbf{x}_{1_{k-m}}, \mathbf{x}_{2_{k-m}}, \mathbf{u}_{k-m})) \\ y_{k-(m-2)} = h(\mathbf{x}_{1_{k-(m-2)}}) = h(f_1(\mathbf{x}_{1_{k-(m-1)}}, \mathbf{x}_{2_{k-(m-1)}}, \mathbf{u}_{k-(m-1)})) \\ \quad = h(f_1(\mathbf{x}_{1_{k-(m-1)}}, f_2(\mathbf{x}_{1_{k-m}}, \mathbf{x}_{2_{k-m}}, \mathbf{u}_{k-m}), \mathbf{u}_{k-(m-1)})) \\ \quad \vdots \\ y_k = h(\mathbf{x}_{1_k}) = h(f_1(\dots, f_2(\mathbf{x}_{1_{k-(m-1)}}, f_2(\mathbf{x}_{1_{k-m}}, \mathbf{x}_{2_{k-m}}, \mathbf{u}_{k-m}), \mathbf{u}_{k-(m-1)}), \dots, \mathbf{u}_{k-1})) \end{array} \right. \quad (3.10)$$

Assumindo o sistema como observável, é possível considerar a existência de uma função  $p(\cdot)$  que permite determinar a observação das variáveis não acessíveis no instante  $k-m$  (3.11).

$$\mathbf{x}_{2_{k-m}} = p(y_k, y_{k-1}, \dots, y_{k-(m-2)}, y_{k-(m-1)}, \mathbf{x}_{1_{k-1}}, \dots, \mathbf{x}_{1_{k-(m-1)}}, \mathbf{u}_k, \dots, \mathbf{u}_{k-(m-1)}, \mathbf{u}_{k-m}) \quad (3.11)$$

Combinando as expressões (3.11) e (3.9) é possível obter a relação funcional desejada (3.12), onde a saída é apenas função das variáveis de comando e de estado acessíveis.

$$y_{k+1} = h(f_1(\mathbf{x}_{1_k}, f_2(\mathbf{x}_{1_{k-1}}, \dots, p(\cdot), \dots, \mathbf{u}_{k-1}), \mathbf{u}_k)) \quad (3.12)$$

Por fim, admitindo a variável de saída como a única acessível e que a variável de comando apresenta dimensão unitária, e é relativamente estável face à dinâmica própria do sistema, a anterior relação funcional pode ser dada por (3.13).

$$y_{k+1} = g_{x_0, p}(y, y_{k-1}, \dots, y_{k-(m-2)}, u_k) \quad (3.13)$$

Além do número de variáveis não acessíveis a influência das anteriores observações da variável de saída pode ainda depender de outros factores, como sejam os parâmetros do sistema dinâmico, a frequência de amostragem considerada ou a zona do domínio de funcionamento considerada. Deste modo, e considerando a relação funcional (3.13), é possível considerar várias relações funcionais  $g_i$  representativas da evolução da variável de saída do sistema dinâmico, mas cujos argumentos são distintos para distintas condições de funcionamento.

$$\begin{aligned}
 y_{k+1} &= g_0(u_k) \\
 y_{k+1} &= g_1(y_k, u_k) \\
 y_{k+1} &= g_2(y_k, y_{k-1}, u_k) \\
 y_{k+1} &= g_3(y_k, y_{k-1}, y_{k-2}, u_k) \\
 &\vdots
 \end{aligned}
 \tag{3.14}$$

As anteriores hipóteses de evolução da saída do sistema dinâmico vão traduzir-se, de uma forma natural, no que, numa perspectiva da teoria da linguagem, se designará por '*produções do tipo p*', onde *p* se refere à dependência da saída face à sua evolução anterior. Assim numa *produção do tipo 0* a evolução da saída dependerá unicamente do comando, enquanto que numa produção do tipo superior a evolução da saída irá também depender da sua própria evolução anterior.

Para desenvolver uma descrição gramatical de um sistema dinâmico controlado são necessários três passos. Os dois primeiros, que se referem à formalização da gramática, são a codificação das variáveis, com o conseqüente estabelecimento dos alfabetos, e a definição da natureza das produções. O terceiro refere-se ao estabelecimento do algoritmo de aprendizagem necessário à extracção das produções a partir de dados experimentais.

### 3.5.1.1 Alfabetos

Como primeiro passo para estabelecer o algoritmo de inferência gramatical torna-se necessário definir o alfabeto da linguagem. Para tal ter-se-á de codificar a informação disponível, isto é, codificar as variáveis de comando e saída. Como já visto anteriormente, esta codificação pode ser uma qualquer, em função das necessidades de descrição assumidas, devendo no entanto escolherem-se codificações que não originem gramáticas muito complexas, as quais colocam sérios problemas de processamento.

Na codificação proposta, o alfabeto terminal –  $\Sigma_T$  – será associado à variável de saída *y*, e o alfabeto não terminal –  $\Sigma_N$  – será associado à variável de comando *u*. Assim a representação das variáveis é efectuada através de um conjunto finito de símbolos, representativos da evolução de um qualquer atributo de cada variável ( 3.15 ), onde *n* representa o número de símbolos terminais e *m* o número de símbolos não terminais.

$$\begin{aligned} y &\xrightarrow{\text{codificação}} y_j \in \{y_1, y_2, \dots, y_n\} \\ u &\xrightarrow{\text{codificação}} U_i \in \{U_1, U_2, \dots, U_m\} \end{aligned} \quad (3.15)$$

Para facilitar a descrição do algoritmo, e mais uma vez sem perda de generalidade, vamos considerar que se codifica a evolução das variáveis de uma forma discreta, quantificando o domínio das referidas variáveis de comando e estado.

Deste modo, existirá uma relação entre o valor assumido pela variável de saída  $y$  e o índice  $j$ , e entre o valor assumido pela variável de comando  $u$  e o índice  $i$ , do seguinte modo

$$\begin{cases} y = y_{\min} & \Leftrightarrow j = 1 \\ y = y_{\max} & \Leftrightarrow j = n \\ u = u_{\min} & \Leftrightarrow i = 1 \\ u = u_{\max} & \Leftrightarrow i = m \end{cases} \quad (3.16)$$

O alfabeto terminal –  $\Sigma_T$  – será constituído pelos símbolos associados à quantificação efectuada para a variável de saída do sistema dinâmico. O alfabeto não terminal –  $\Sigma_N$  – será constituído pelos símbolos resultantes da quantificação da variável de comando. Neste alfabeto devem ser ainda considerados dois símbolos adicionais: o símbolo ‘S’ que representa o símbolo inicial, e um símbolo especial ‘ $\delta$ ’ cuja função será descrita adiante. Assim a constituição de ambos os alfabetos é dada por ( 3.17 ).

$$\begin{aligned} \Sigma_T &= \{y_1, y_2, \dots, y_n\} \\ \Sigma_N &= \{S, U_1, U_2, \dots, U_m, \delta\} \end{aligned} \quad (3.17)$$

Após ter sido definido o alfabeto completo  $\Sigma = \Sigma_T \cup \Sigma_N$ , é necessário estabelecer a definição do sistema de rescrita, constituído pelo anterior alfabeto e pelo conjunto de produções.

### 3.5.1.2 Produções

As produções são definidas como sendo regras de substituição da forma ( 3.18 ), onde  $y_1 \dots y_p$  representam uma sequência de símbolos terminais,  $U_k$  um símbolo

não terminal e  $\delta$  um símbolo não terminal especial. Este último símbolo é utilizado para permitir a conclusão, ou não, da palavra gerada, através da utilização das produções especiais ( 3.19 ).

$$y_1 \dots y_p U_k \rightarrow y_1 \dots y_p y_{p+1} \delta \quad (3.18)$$

$$\begin{aligned} \delta &\rightarrow U_j, \quad j=1 \dots m \\ \delta &\rightarrow \lambda \end{aligned} \quad (3.19)$$

Uma *produção do tipo p* codifica a evolução da variável de saída, dependendo das suas *p* codificações passadas e da codificação da variável de comando *u*. Considera-se assim uma relação directa ( 3.20 ) entre as relações funcionais não lineares  $g_i$  ( 3.14 ) e as *produções do tipo p*, estabelecendo uma relação entre estas e a dinâmica do sistema.

$$x_{k+1} = g_i(.) \Leftrightarrow \text{produção do tipo } p, \text{ com } p = i \quad (3.20)$$

Considerando  $y_a, y_b$  e  $y_c$  como quaisquer símbolos terminais e  $U_A$  como um qualquer símbolo não terminal, as várias produções definem-se como se apresenta na Tabela 3.I.

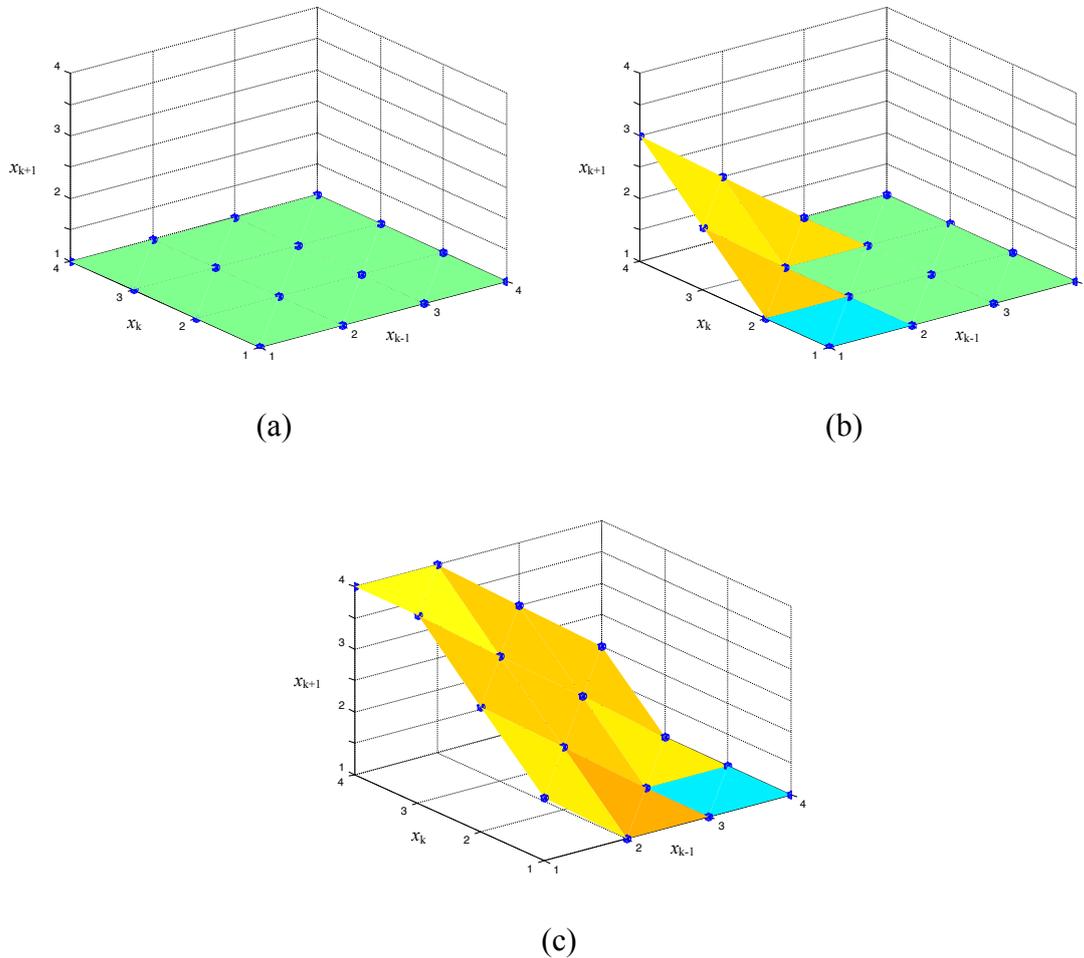
**Tabela 3.I – Definição das produções do Sistema de Rescrita.**

Produção do tipo 0	$U_A \rightarrow y_a \delta$
Produção do tipo 1	$y_a U_A \rightarrow y_a y_b \delta$
Produção do tipo 2	$y_a y_b U_A \rightarrow y_a y_b y_c \delta$
...	...

Considere-se como exemplo de aplicação o seguinte sistema dinâmico não linear ( 3.21 ), onde *x* representa o estado, *u* o comando e *y* a saída. Tanto a variável de estado como a de comando apenas podem tomar os valores apresentados, sendo que  $x_k=1$  se  $x_k < 1$  e  $x_k=4$  se  $x_k > 4$ .

$$\begin{cases} x_{k+1} = x_k - x_{k-1} + 2u_k \\ y_k = x_k \end{cases} \text{ com } \begin{cases} x_k \in \{1,2,3,4\} \\ u_k \in \{-1,0,+1\} \end{cases} \quad (3.21)$$

O domínio de funcionamento deste sistema dinâmico é assim definido pelas três superfícies que se apresentam na Figura 3.6.



**Figura 3.6. Domínio de funcionamento do sistema dinâmico ( 3.21 ) com:  
 (a)  $u_k = -1$ ; (b)  $u_k = 0$ ; (c)  $u_k = +1$ .**

Considere-se a codificação apresentada na Tabela 3.II, onde as letras minúsculas se referem a símbolos do alfabeto terminal e as letras maiúsculas a símbolos do alfabeto não terminal. A linguagem gerada por este sistema dinâmico é representada por uma gramática, designada por  $G_{ex}$ , cujo sistema de rescrita completo apresenta uma *produção do tipo 0*, duas *produções do tipo 1* e vinte e quatro *produções do tipo 2*, além de quatro produções especiais do tipo ( 3.19 ).

**Tabela 3.II. Codificação escolhida para o sistema dinâmico ( 3.21 ).**

$y_k \leftrightarrow \sum_T$	$u_k \leftrightarrow \sum_N$
1 $\leftrightarrow$ a	-1 $\leftrightarrow$ A
2 $\leftrightarrow$ b	0 $\leftrightarrow$ B
3 $\leftrightarrow$ c	+1 $\leftrightarrow$ C
4 $\leftrightarrow$ d	

Na Tabela 3.III exemplifica-se a aplicação sucessiva de algumas das produções pertencentes ao sistema de rescrita da gramática anterior, bem como a palavra resultante dessa aplicação.

**Tabela 3.III. Exemplo de aplicação do Sistema de Rescrita da Gramática  $G_{ex}$ .**

Produções aplicadas	Evolução da palavra
	S
$S \rightarrow b \delta$	b $\delta$
$\delta \rightarrow B$	b B
$b B \rightarrow b a \delta$	b a $\delta$
$\delta \rightarrow C$	b a C
$b a C \rightarrow b a a \delta$	b a a $\delta$
$\delta \rightarrow A$	b a a A
$A \rightarrow a \delta$	b a a a $\delta$
$\delta \rightarrow \lambda$	b a a a

Partindo do símbolo não terminal inicial, e aplicando sucessivamente a sequência de produções indicada obtém-se a palavra ‘baaa’. Esta palavra, pertencente à linguagem gerada pelo sistema dinâmico ( 3.21 ), corresponde à evolução da variável de saída  $y$  (que após descodificação seria  $y_{t=0..3}=\{2,1,1,1\}$ ) face à evolução da variável de comando, que se encontra associada à evolução dos símbolos não terminais presentes nas produções. A codificação ‘BCA’ é obtida a partir da evolução da variável de comando  $u_{t=0..2}=\{0,+1,-1\}$ . O símbolo terminal ‘b’ com que se inicia a palavra resultante corresponde ao estado/saída inicial assumido para o sistema dinâmico ( $x_0=2$ ).

Conclui-se assim que, com esta especificação dos alfabetos terminal e não terminal, e após aplicação das várias produções da gramática resulta uma palavra, pertencente à linguagem gerada pelo sistema dinâmico, que nos fornece evolução da codificação (qualquer que ela seja) da variável de saída. A evolução da codificação da variável de comando encontra-se associada aos símbolos do alfabeto não terminal presentes nas sucessivas produções aplicadas.

### 3.5.2 Algoritmo de Inferência gramatical

O processo de inferência gramatical encontra-se esquematicamente representado na Figura 3.7.

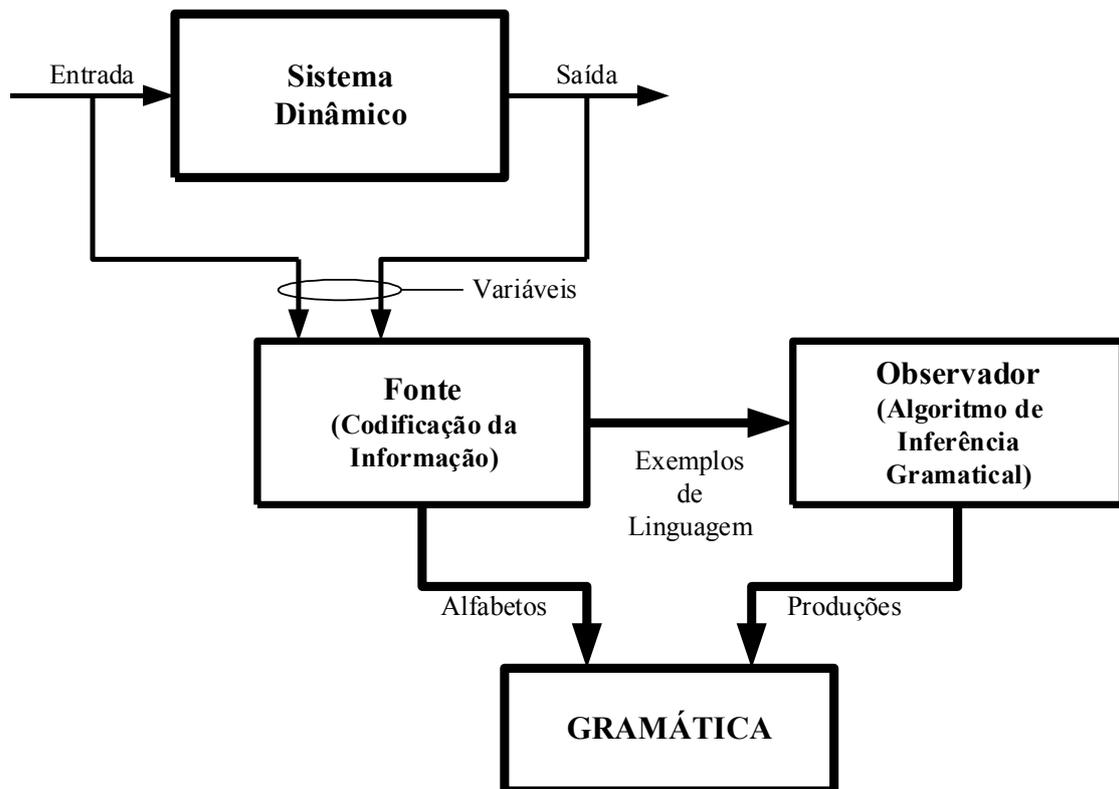


Figura 3.7. Inferência Gramatical.

As variáveis provenientes de um dado sistema dinâmico funcionam como a informação disponível acerca desse mesmo sistema. Esta informação deverá ser codificada por uma fonte (de acordo como os alfabetos escolhidos), gerando um conjunto de palavras

representativas da linguagem. Assume-se que estas palavras possuem propriedades estruturais únicas que podem ser caracterizadas por uma dada gramática e que todas as palavras provenientes da fonte estão contidas na linguagem gerada pelo sistema dinâmico. A partir deste conjunto de informação positiva o observador, mediante um algoritmo de inferência gramatical, deverá inferir as produções da gramática desconhecida que se pretende inferir.

Por forma a constituir uma amostra finita da linguagem gerada pelo sistema dinâmico é necessário fazer evoluir o referido sistema através do seu domínio de funcionamento, por aplicação de uma sucessão de comandos conhecidos.

Tal como referido quando foi abordado o conceito de aprendizagem através de exemplos, a constituição de um conjunto de treino (aplicando ao sistema dinâmico uma evolução conhecida da variável de comando) resulta numa sucessão de pares ordenados, para um dado estado inicial.

$$\mathbf{E}_{x_0} = \left\{ \begin{pmatrix} u_1 \\ y_1 \end{pmatrix} \begin{pmatrix} u_2 \\ y_2 \end{pmatrix} \begin{pmatrix} u_3 \\ y_3 \end{pmatrix} \cdots \begin{pmatrix} u_n \\ y_n \end{pmatrix} \right\} \quad (3.22)$$

Deste conjunto de treino retirar-se-ão, após codificação, palavras que constituem exemplos da linguagem gerada pelo sistema dinâmico. A cada uma dessas palavras (alfabeto terminal) ( 3.23 ) encontram-se associadas as respectivas evoluções da variável de comando (alfabeto não terminal) ( 3.24 ).

$$y_p \ y_{p+1} \ y_{p+2} \ \cdots \ y_{p+q} \ , \ \forall p, q \geq 0 \quad (3.23)$$

$$U_p \ U_{p+1} \ U_{p+2} \ \cdots \ U_{p+q} \ , \ \forall p, q \geq 0 \quad (3.24)$$

Como informação necessária ao funcionamento do algoritmo de inferência gramatical consideram-se assim as palavras retiradas do conjunto de treino, as evoluções da variável de comando associadas, e o tipo de produção máximo que se pretende obter – representado por  $p_{\max}$ . O algoritmo retorna as produções inferidas a partir da análise das palavras consideradas.

Para cada palavra, o algoritmo de inferência gramatical, apresentado na Figura 3.8, funciona do seguinte modo:

1. Por cada novo símbolo não terminal, assume-se uma nova *produção do tipo 0*.
2. Uma nova *produção do tipo n+1* é gerada sempre que os dados sejam contraditórios com as *produções do tipo n* anteriormente estabelecidas. As *produções do tipo n* que estejam em conflito são também promovidas a *produções do tipo n+1* ou são eliminadas se não existir informação passada suficiente para tal.

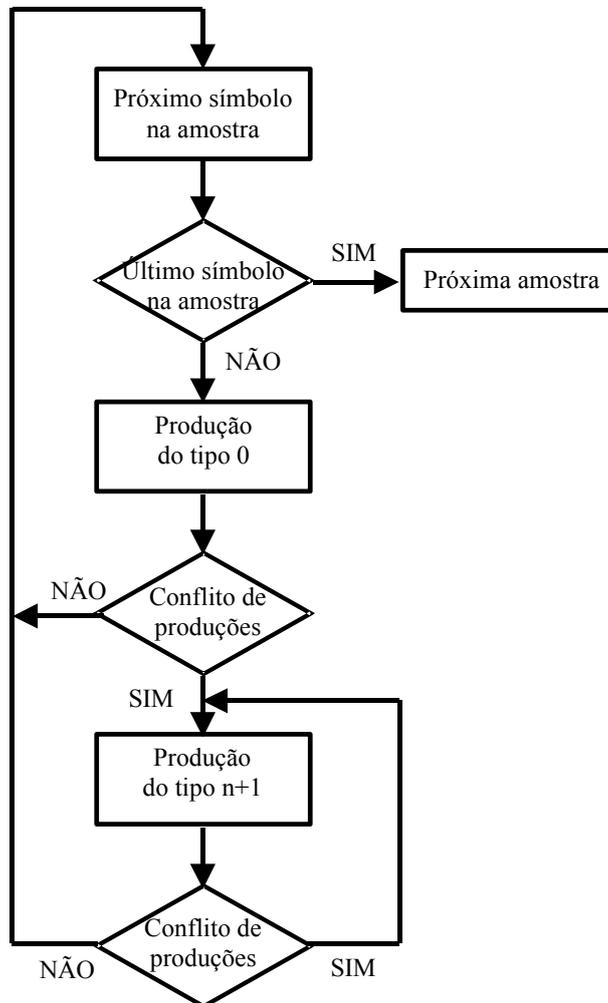


Figura 3.8. Algoritmo de Inferência Gramatical.

Por forma a ser possível rever as *produções do tipo n*, promovendo-as a *produções do tipo n+1*, é necessário manter uma janela de memória por forma a registar um certo número de passos passados da evolução do sistema. O comprimento desta janela de memória não necessita de ser superior a  $p_{max}$ . De uma forma geral não é grave não considerar uma janela de memória e apenas apagar todas as produções conflituosas de ordem inferior. Isto porque todas as produções mais relevantes aparecerão novamente ao longo do processo de aprendizagem.

Existem duas propriedades interessantes associadas a este algoritmo de inferência gramatical, que se revelam bastante úteis quando o mesmo é implementado computacionalmente. A primeira diz-nos que, não considerando as produções especiais ( 3.19 ) o número máximo de produções de um determinado tipo  $p$  é dado por ( 3.25 ), onde  $num(.)$  representa o número de símbolos de um dado alfabeto. No alfabeto não terminal excluem-se os símbolos inicial – S – e especial –  $\delta$ .

$$\max \text{prod}_p = (\text{num}(\Sigma_T))^p (\text{num}(\Sigma_N)) \quad ( 3.25 )$$

A segunda diz-nos que uma *produção do tipo p* válida anula um número de *produções do tipo p+1* dado por ( 3.26 ), onde  $p_{max}$  se refere ao tipo de produções máximo considerado.

$$\sum_{i=1}^{p_{max}-p} (\text{num}(\Sigma_T))^i \quad ( 3.26 )$$

Ao contrário do que acontece com as regras inferidas, por exemplo através dos processos usuais de lógica fuzzy, a estrutura das produções, obtidas pelo anterior algoritmo de inferência gramatical, não é estabelecida *á priori*. Uma mistura de diferentes tipos de produções pode facilmente ser obtida na gramática. Esta propriedade providencia uma adaptação flexível a sistemas dinâmicos que exibam tipos de comportamento distinto em diferentes regiões do seu domínio de funcionamento.

A anterior propriedade revela-se muito importante quando se pretende modelizar processos industriais, os quais são geralmente sistemas não lineares que não podem ser modelizados em todas as condições de operação com um único modelo. Vários autores substituem complicados modelos não lineares por modelos locais afins (Leonaritis e Billings, 1985), obtidos por processos de identificação, para representar o

comportamento global de todo o processo. O algoritmo de identificação deverá identificar as várias zonas de operação do processo, nas quais os dados extraídos podem ser aproximados por modelos locais afins. Os vários pontos de operação podem ser obtidos por métodos de *clustering* (Babuska e Verbruggen, 1995) e a estrutura e parâmetros de cada modelo local por utilização do esquema de Frisch (Beghelli, Guidorzi e Soverini, 1990). Por utilização da modelização gramatical esta identificação resulta de uma forma automática.

Num sistema dinâmico comandado, os dois tipos de variáveis, de saída e de comando, têm uma natureza distinta. Isto é representado no presente formalismo pela atribuição de símbolos terminais às variáveis dinâmicas e de símbolos não terminais às variáveis de comando. Assim as produções representam a acção do comando no contexto da dinâmica, passada e presente, do sistema. Com esta especificação para a gramática, a linguagem gerada pelo sistema dinâmico é, segundo a hierarquia de Chomsky, sensível ao contexto.

Em aplicações práticas, como veremos adiante, ao algoritmo de inferência gramatical são adicionados mecanismos de interpolação gramatical e de rejeição de ruído. A interpolação gramatical será apresentada no ponto seguinte, enquanto que a rejeição de ruído é implementada mantendo apenas as produções que aparecem um número suficiente de vezes nos dados experimentais. Este procedimento simples assegura a estabilidade e robustez do algoritmo de aprendizagem. Tal como já referido anteriormente, e por razões de ordem prática, é assumido um limite máximo para a ordem das produções inferidas.

Retomando o sistema dinâmico do exemplo anterior ( 3.21 ), considere-se a seguinte sequência, já codificada, obtida a partir da sua evolução dinâmica:

$$\begin{array}{l} \text{Variável de comando } u: C C B C C B A B \\ \text{Variável de saída } y: c b a a b a a a \end{array} \quad ( 3.27 )$$

No instante zero o algoritmo analisa ambos os símbolos iniciais de ambas as variáveis: comando e saída. Uma vez que ainda não se possui mais nenhuma informação, é estabelecida a *produção do tipo 0*  $C \rightarrow c\delta$ . Após a análise do segundo símbolo o algoritmo estabelecerá uma nova *produção do tipo 0*  $C \rightarrow b\delta$ . No entanto, esta produção

entra em conflito com a anteriormente estabelecida  $C \rightarrow c\delta$ , uma vez que a gramática não se considera probabilística. Assim, neste caso, é estabelecida uma *produção do tipo 1*  $cC \rightarrow cb\delta$ . A *produção do tipo 0*  $C \rightarrow c\delta$  é eliminada porque não se tem nenhum conhecimento sobre o passado anterior ao primeiro símbolo  $c$ . Da análise do terceiro símbolo obtém-se a *produção do tipo 0*  $B \rightarrow a\delta$ . Ao analisar o quarto símbolo obtém-se a *produção do tipo 1*  $aC \rightarrow aa\delta$ , pois a *produção do tipo 0*  $C \rightarrow a\delta$  entra em conflito com as anteriormente invalidadas. Um novo conflito surge aquando da análise do quinto símbolo. É estabelecida a *produção do tipo 2*  $aaC \rightarrow aab\delta$  e a *produção do tipo 1*  $aC \rightarrow aa\delta$  é corrigida para a *produção do tipo 2*  $baC \rightarrow baa\delta$ . Da análise do sexto símbolo não resulta nenhuma nova produção. A análise do sétimo símbolo gera uma *produção do tipo 0*  $A \rightarrow a\delta$  e da análise do último símbolo não resulta nenhuma nova produção. Deste modo ficamos com o seguinte conjunto de produções:

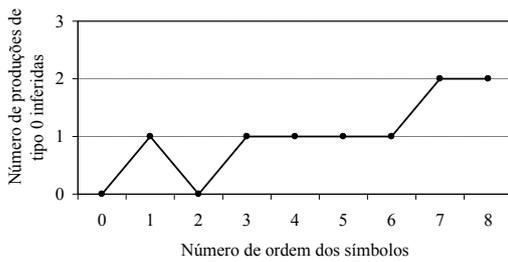
$$\begin{aligned}
 A &\rightarrow a\delta \\
 B &\rightarrow a\delta \\
 cC &\rightarrow cb\delta \\
 aaC &\rightarrow aab\delta \\
 baC &\rightarrow baa\delta
 \end{aligned}
 \tag{ 3.28 }$$

Este conjunto de produções – obtido por aplicação do algoritmo de inferência gramatical a partir de uma pequena amostra – está longe das vinte e quatro *produções de tipo 2*, duas *produções de tipo 1* e uma *produção de tipo 0* que constituem o sistema de rescrita da gramática que representa este sistema dinâmico ( 3.21 ). Este exemplo reforça a necessidade de considerar mais informação quando se pretende inferir uma gramática, isto é, devem-se considerar mais e diferentes palavras ou palavras com maior comprimento no processo de inferência.

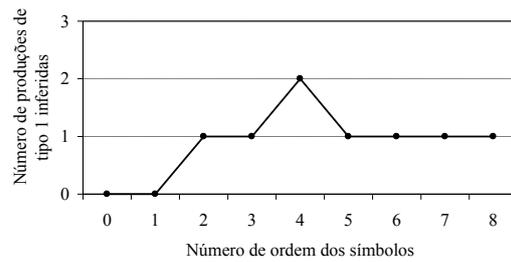
Às anteriores produções devem ser necessariamente adicionadas as produções especiais, também pertencentes ao sistema de rescrita:

$$\begin{aligned}
 \delta &\rightarrow A \\
 \delta &\rightarrow B \\
 \delta &\rightarrow C \\
 \delta &\rightarrow \lambda
 \end{aligned}
 \tag{3.29}$$

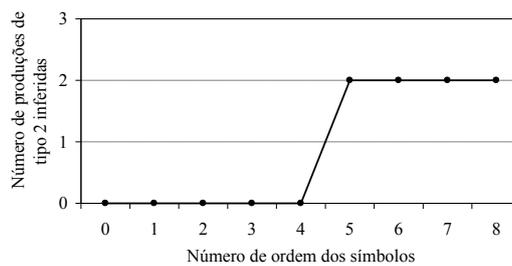
À medida que se avança na análise de uma palavra, o algoritmo começa por estabelecer produções de tipo inferior, que podem ir sendo consideradas inválidas e substituídas por produções de tipo superior. Esta característica é típica de um algoritmo de aprendizagem através de exemplos, onde o conhecimento do sistema (neste caso da linguagem por ele gerada) vai evoluindo à medida que se obtém mais informação sobre o mesmo. Na Figura 3.9 apresenta-se a evolução das produções, obtidas a partir dos dados anteriores, onde se pode verificar que o número de produções de um determinado tipo vai variando à medida que se evolui no processo de aprendizagem.



(a)



(b)



(c)

**Figura 3.9. Evolução do número de produções de tipo  $p$  obtidas pelo mecanismo de inferência gramatical para a linguagem proveniente do sistema dinâmico ( 3.21 ):**  
**(a) produções de tipo 0; (b) produções de tipo 1; (c) produções de tipo 2.**

Tal como descrito anteriormente, e a título de exemplo, verifica-se que após a análise do segundo símbolo é estabelecida uma *produção do tipo 1* e a única *produção do tipo 0* existente é eliminada, porque não existe nenhum conhecimento anterior que a possa promover. Após análise do quinto símbolo são estabelecidas duas novas *produções do tipo 2*, uma directamente e outra por promoção de uma *produção do tipo 1* já existente.

### **3.5.3 Lacunas gramaticais e interpolação gramatical**

Uma aplicação importante de uma qualquer gramática é o reconhecimento de palavras pertencentes à linguagem por ela descrita. Dada uma palavra, o modelo gramatical deverá possuir a capacidade de reconhecer se a palavra pertence ou não à linguagem descrita por essa gramática.

Outra aplicação prende-se com o facto de não verificar apenas se uma palavra pertence ou não à linguagem descrita por uma gramática, mas também determinar a sequência de produções utilizadas para derivar essa palavra. Este problema é conhecido na literatura por análise gramatical.

Existem duas abordagens fundamentais ao problema da análise gramatical. Na primeira (*top-down*) inicia-se o processo com o símbolo inicial e tenta-se derivar a palavra em análise aplicando sucessivamente as produções do sistema de rescrita. Na segunda (*bottom-up*) o processo inicia-se pela palavra em análise e aplicando as produções, de uma forma inversa, tenta-se reduzi-la ao símbolo inicial.

Uma aplicação específica da gramática formalizada na secção 3.5.1 e resultante do algoritmo de inferência apresentado na secção 3.5.2, corresponde a estimar o símbolo seguinte face ao conhecimento dos primeiros símbolos de uma dada palavra, pertencente à linguagem descrita pela gramática. Assim, em função dos símbolos presentes torna-se necessário determinar quais as produções aplicáveis por forma a estimar o próximo símbolo da palavra.

Quando se utilizam algoritmos de inferência gramatical, onde a aprendizagem é efectuada através de exemplos, a quantidade e qualidade desses exemplos é muito importante. O número de palavras consideradas como exemplos tem de ser representativo dos vários aspectos da linguagem. Como vimos no exemplo apresentado

no final da secção 3.5.2, utilizando somente uma pequena palavra foram apenas inferidas duas *produções de tipo 0* uma de tipo 1 e duas de tipo 2, para uma linguagem descrita por um sistema de rescrita com uma *produção de tipo 0*, duas *produções de tipo 1* e vinte e quatro *produções de tipo 2*. Além disto algumas das produções inferidas não se encontram correctas.

Problema comum aos aspectos focados anteriormente está relacionado com o facto de poderem existir produções que não foram inferidas aquando do processo de aprendizagem. Deste modo o reconhecimento de palavras, a determinação das produções necessárias para as derivar, ou mesmo a determinação do próximo símbolo numa dada palavra podem ficar comprometidos, pois o sistema de rescrita não se apresenta fiável.

Quando este problema ocorre, está-se em presença do que se designará por lacunas gramaticais, isto é, ausência de produções no sistema de rescrita inferido. Em presença de uma palavra que efectivamente pertence à linguagem gerada pelo sistema dinâmico, cuja análise se torna difícil devido à ausência de produções essenciais no sistema de rescrita da gramática inferida, uma possível solução será estimar essas produções a partir do conhecimento das já inferidas pelo algoritmo de aprendizagem. Este processo que consiste no estabelecer de produções que não tenham sido estabelecidas na fase de inferência será designado por interpolação gramatical.

Definindo uma métrica no espaço das palavras, o processo de interpolação gramatical estabelecerá produções relevantes que não tenham sido inferidas durante a aprendizagem, minimizando este problema.

Considere-se por exemplo uma de *produção de tipo 3*, na forma ( 3.30 ), onde  $y$  e  $U$  se referem respectivamente a símbolos terminais e não terminais.

$$y_{k-3} y_{k-2} y_{k-1} U_k \rightarrow y_{k-3} y_{k-2} y_{k-1} y_k \delta \quad ( 3.30 )$$

Perante uma qualquer produção inexistente, por exemplo  $y_a y_b y_c U_1 \rightarrow y_a y_b y_c y_{?} \delta$ , o objectivo é determinar a distância da palavra  $y_a y_b y_c$ , para o mesmo símbolo não terminal, às mais próximas pertencentes a produções já efectivamente inferidas. O símbolo  $y_{?}$  refere-se àquele que se pretende estimar por forma a estabelecer a produção

em falta. Em função das distâncias medidas, o símbolo  $y_?$  será estimado recorrendo a uma média ponderada dos símbolos  $y_k$  presentes nas produções vizinhas.

O processo de interpolação gramatical estabelece novas produções recorrendo a um procedimento de semelhança estrutural. A ideia fundamental da semelhança estrutural é baseada na medida de similaridade entre um padrão de entrada desconhecido e as estruturas de dados disponíveis. Assim a medida de similaridade é a distância entre as produções não existentes e as mais próximas.

Existem vários algoritmos desenvolvidos para determinar a semelhança estrutural entre palavras. Segundo o algoritmo básico a distância entre duas palavras é função das operações de edição – substituir, inserir e apagar – necessárias para transformar uma palavra noutra. Dadas duas quaisquer palavras existem várias sequências de operações que possibilitam a transformação de uma noutra. Associando a cada tipo de operação um determinado custo, define-se como custo de uma sequência de operações ( 3.31 ) a soma dos custos de cada operação, onde  $s_{op}$  representa a sequência de operações e  $c(e_i)$  representa o custo de cada operação de edição  $e_i$ .

$$c(s_{op}) = \sum_{i=1}^n c(e_i) \quad ( 3.31 )$$

Dadas duas palavras  $z_1$  e  $z_2$ , a distância entre elas  $d(z_1, z_2)$  é definida como o mínimo dos custos associados a cada sequência de operações passível de transformar uma noutra.

$$d(z_1, z_2) = \min \left\{ c(s_{op}) \left| \begin{array}{l} \text{sendo } s_{op} \text{ a sequência de operações} \\ \text{necessárias para transformar } z_1 \text{ em } z_2 \end{array} \right. \right\} \quad ( 3.32 )$$

Entre as várias soluções existentes na literatura, uma delas baseia-se num algoritmo de programação dinâmica que procura o mínimo custo entre as várias possibilidades de transformação (Bunke, 1990).

Aplicando o procedimento da semelhança estrutural a produções do mesmo tipo restringe-se o conjunto de operações de edição à operação substituição. Admitindo a construção do alfabeto a partir da codificação quantificada das variáveis envolvidas,

surge de uma forma natural a métrica no espaço das palavras. Assim o custo da operação substituição é função das distância entre símbolos.

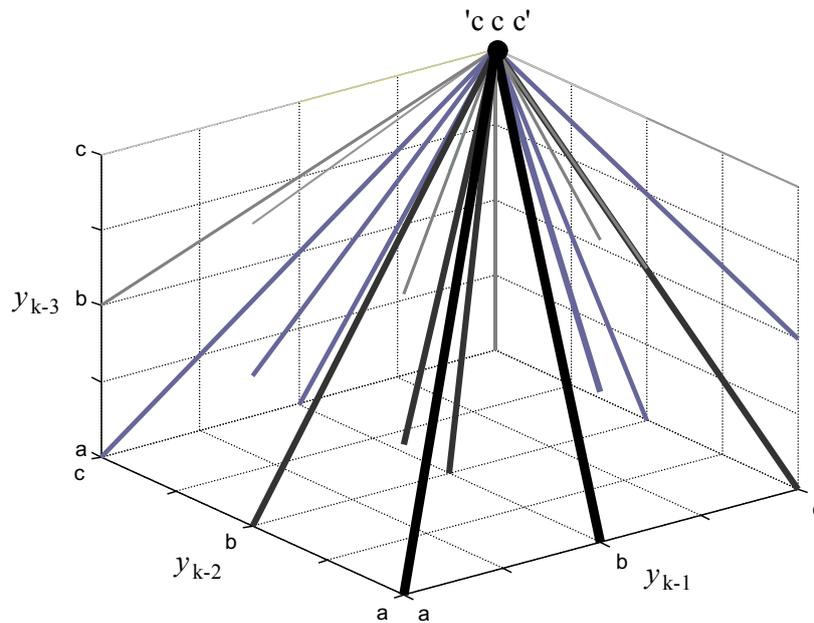
$$c(a \rightarrow b) = |a - b| \quad ( 3.33 )$$

É possível estender a definição de distância entre palavras considerando dependência do contexto. Os custos das operações de edição dependem assim dos símbolos e da sua posição na palavra ou mesmo da sua vizinhança.

Este aspecto reveste-se de considerável importância quando se descrevem linguagens geradas por sistemas dinâmicos. Um conhecimento prévio, ainda que insipiente, do sistema pode dar-nos uma indicação da importância relativa de cada símbolo numa dada produção. Trabalhos anteriores (Martins, Pires e Dente, 1998) descrevem a importância deste estudo quando se considera a modelização de um accionamento electromecânico baseado numa máquina assíncrona de rotor em curto-circuito sem recorrer à leitura dos fluxos magnéticos. Numa *produção de tipo 3* ( 3.30 ) a influência do símbolo  $y_{k-1}$  pode ser muito mais determinante na evolução do sistema dinâmico do que a dos símbolos  $y_{k-2}$  e  $y_{k-3}$ . Deste modo os custos associados a uma substituição do símbolo  $y_{k-1}$  poderão ser maiores do que os associados aos outros, tal como se apresenta em ( 3.34 ). É importante sublinhar que esta dependência do contexto está obrigatoriamente vinculada a uma análise prévia do sistema dinâmico.

$$\begin{cases} c(a \rightarrow b : y_{k-3} y_{k-2} \underline{y_{k-1}}) = \alpha |a - b| \\ c(a \rightarrow b : y_{k-3} \underline{y_{k-2}} y_{k-1}) = \beta |a - b| \\ c(a \rightarrow b : \underline{y_{k-3}} y_{k-2} y_{k-1}) = \gamma |a - b| \end{cases} \quad \text{com } \alpha \geq \beta \geq \gamma > 0 \quad ( 3.34 )$$

Como exemplo considere-se um alfabeto terminal  $\Sigma_T = \{a, b, c\}$ , os parâmetros  $\alpha=3$ ,  $\beta=2$  e  $\gamma=1$ , e a palavra ‘ccc’ correspondente os símbolos  $y_{k-1}$ ,  $y_{k-2}$  e  $y_{k-3}$ . de uma produção inexistente. Assumindo que todas as palavras vizinhas existem, e segundo as definições anteriores, a distância mínima ocorre para  $d(ccc, bcc)=1$  e a distância máxima para  $d(ccc, aaa)=12$ . Na Figura 3.10 apresentam-se todas as distâncias da palavra ‘ccc’ às suas vizinhas, onde os traços mais grossos e mais escuros referenciam distâncias maiores.



**Figura 3.10. Distância dependente do contexto entre a palavra ‘ccc’ e as suas vizinhas.**

Quando uma palavra não pode ser reconhecida porque não existe nenhuma produção que gere um símbolo nessa palavra, utiliza-se uma fórmula de interpolação gramatical ( 3.35 ) para obter o símbolo terminal  $y_{p+1}$  no lado direito da produção.

$$y_{p+1|prod\ j} = \text{codificação} \left( \left( \sum_{\substack{i=1 \\ i \neq j}}^n d_i \right) \sum_{i=1}^n \frac{1}{d_i} y_{p+1|prod\ i} \right) \quad ( 3.35 )$$

A fórmula de interpolação gramatical recorre a uma média ponderada das produções disponíveis do tipo considerado. A distância  $d_i$  é a distância entre as palavras  $y_1 \dots y_p U_k$  no lado esquerdo das produções existentes e a palavra da nova produção.

A aplicação deste mecanismo possibilita o estabelecimento de novas produções, antes inexistentes na gramática inferida, a partir de um conjunto restrito de exemplos. Apesar de poder ser aplicado a produções de qualquer tipo será desejável considerá-lo apenas nas produções do tipo máximo admitido, não reduzindo a riqueza do sistema de rescrita da gramática representativa da linguagem proveniente do sistema dinâmico.

### **3.6 IDENTIFICAÇÃO DE PALAVRAS COMO PADRÕES**

A evolução de um sistema dinâmico comandado depende, não apenas dos seus parâmetros, mas também da evolução do seu comando. No entanto, e dependendo do tipo de sistema, a influência do comando pode ser diminuta face à dinâmica própria do sistema. Nestes casos é interessante extrair padrões do comportamento próprio do sistema.

A identificação de padrões revela-se também interessante quando se perspectiva a detecção de anomalias, onde é fundamental efectuar a distinção entre padrões de comportamento normal e anormal.

Por identificação de sistemas entende-se o processo de determinação de equações diferenciais ou às diferenças, ou os parâmetros dessas equações, por forma a descrever um processo físico de acordo com algum critério pré-determinado (Sage e Melsa, 1971). Entende-se por reconhecimento de padrões o agrupar desses mesmos padrões em categorias (Kanal, 1974).

Na análise do comportamento de sistemas dinâmicos autónomos (ou cuja dinâmica é pouco influenciada pelo comando) os objectos de interesse apresentam-se como funções dependentes, entre outros parâmetros (conjunto designado por  $\mathbf{p}$ ), do tempo ( 3.36 ).

$$y = \varphi_{\mathbf{p}}(t) \quad (3.36)$$

Assumindo que se efectuam amostragens em tempo discreto –  $t_1, t_2, \dots, t_n$  – é possível construir um vector ( 3.37 ), cujas componentes são quantidades reais no contra domínio da função  $\varphi_{\mathbf{p}}(\cdot)$ .

$$\mathbf{Y} = \{ \varphi_{\mathbf{p}}(t_1), \varphi_{\mathbf{p}}(t_2), \dots, \varphi_{\mathbf{p}}(t_n) \} \quad (3.37)$$

Estes vectores  $\mathbf{Y}$  podem assim ser interpretados como pontos num espaço euclidiano de dimensão  $n$ , sendo deste modo possível identificar padrões de comportamento na evolução dinâmica dos sistemas.

Anteriormente descreveu-se o estabelecimento de gramáticas cujos sistemas de rescrita estabelecem produções entre os símbolos que formam as palavras pertencentes à linguagem. É também possível estender as produções gramaticais às relações que se podem estabelecer entre palavras, na mesma linguagem. Esta abordagem apresenta-se muito útil quando se analisam sinais periódicos, pois, aparte das facilidades de identificação, permite também reduzir o número de produções necessárias à representação da linguagem.

Apesar da capacidade de representação do comportamento da dinâmica de sistemas, por parte das gramáticas, o número de produções pode tornar-se excessivo quando estamos em presença de sinais que apresentam componentes de frequência muito inferior à frequência de amostragem.

Considere-se um sinal periódico genérico variante no tempo ( 3.36 ), amostrado periodicamente ( 3.38 ). O intervalo de amostragem é imposto por razões de ordem prática, sendo determinante na sua escolha a frequência mais elevada do espectro do sinal que ainda se pretende observar. A partir deste sinal genérico, e após o necessário processo de codificação, é possível gerar palavras de uma determinada linguagem.

$$y_i = \varphi_p(t_i) \quad ; \quad i = 1 \dots n \quad ( 3.38 )$$

Admita-se também a existência de produções de tipo máximo  $q$  ( 3.39 ). Sem perda de generalidade pode assumir-se o símbolo não terminal  $U$  como único, denotando o facto de a evolução do sinal não estar influenciada por nenhum comando.

$$y_1 \dots y_q U \rightarrow y_1 \dots y_q y_{q+1} \delta \quad ( 3.39 )$$

Assim, a linguagem associada à codificação do sinal amostrado só pode ser bem representada na totalidade do seu espectro de frequência, pelo sistema de rescrita, quando o período da sua componente de mais baixa frequência for inferior a  $2q$  símbolos. Em presença de sinais com componentes de muito baixa frequência, para

representar a linguagem na sua totalidade será necessário aumentar o tipo máximo de produção permitida, com o conseqüente aumento da complexidade da gramática. Uma alternativa consiste em considerar a existência de produções gramaticais entre palavras e não apenas entre símbolos. A existência de sinais de período  $2q$  pode deste modo ser representada por apenas duas *produções de tipo 1* que relacionam palavras de comprimento  $q$ .

Considere-se, como exemplo, uma onda quadrada amostrada e codificada originando uma seqüência de símbolos onde se pode identificar um período igual a oito símbolos ( 3.40 ).

$$Y_1 = \{a a a a c c c c a a a a c c c c a a a a c c c c \dots\} \quad ( 3.40 )$$

Após o processo de aprendizagem, a gramática inferida apresenta duas *produções de tipo 2*, duas *produções de tipo 3* e quatro *produções de tipo 4* ( 3.41 ), representando a totalidade da linguagem. Uma vez que cada *produção de tipo p* é constituída por um conjunto de  $2p+3$  símbolos, necessitamos de uma totalidade de 76 símbolos no sistema de rescrita. Nesta análise não se contabilizam as produções especiais definidas em ( 3.19 ), pois não fazem aumentar o número de símbolos envolvidos.

$$\begin{aligned} a c U_k &\rightarrow a c c \delta \\ c a U_k &\rightarrow c a a \delta \\ a c c U_k &\rightarrow a c c c \delta \\ c a a U_k &\rightarrow c a a a \delta \\ a c c c U_k &\rightarrow a c c c c \delta \\ c a a a U_k &\rightarrow c a a a a \delta \\ c c c c U_k &\rightarrow c c c c a \delta \\ a a a a U_k &\rightarrow a a a a c \delta \end{aligned} \quad ( 3.41 )$$

Assumindo a existência de palavras com comprimento de quatro símbolos, o alfabeto terminal da gramática inferida pode ser constituído por um conjunto de duas palavras ( 3.42 ), e o sistema de rescrita por apenas duas *produções de tipo 1* ( 3.43 ). Estas duas

produções da gramática de palavras também representam a totalidade da linguagem, recorrendo apenas a um total de 10 símbolos.

$$\begin{aligned} z_a &= a a a a \\ z_c &= c c c c \end{aligned} \tag{3.42}$$

$$\begin{aligned} z_a U_k &\rightarrow z_a z_c \delta \\ z_c U_k &\rightarrow z_c z_a \delta \end{aligned} \tag{3.43}$$

Considere-se agora que, na anterior sequência de símbolos, o seu período aumenta para doze símbolos ( 3.44 ), mantendo  $p_{\max}=4$  o tipo da produção máxima do sistema de rescrita da gramática a inferir.

$$Y_2 = \{a a a a a c c c c c c a a a a a c c c c c c a a a a a c c c c c c \dots\} \tag{3.44}$$

Após o novo processo de aprendizagem, e mantendo  $p_{\max}=4$ , a gramática inferida continua a apresentar duas *produções de tipo 2*, duas *produções de tipo 3* e duas *produções de tipo 4*, revelando-se insuficiente para representar a totalidade da nova linguagem. Na Figura 3.11 pode verificar-se que apenas patamares de três símbolos são bem representados, sendo que a gramática apenas detecta as transições com um atraso de um símbolo.

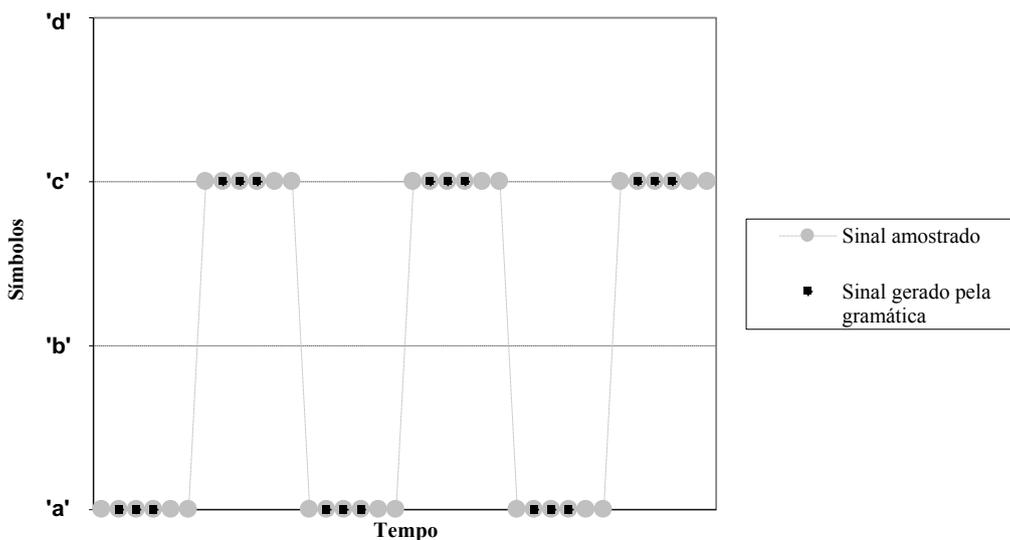


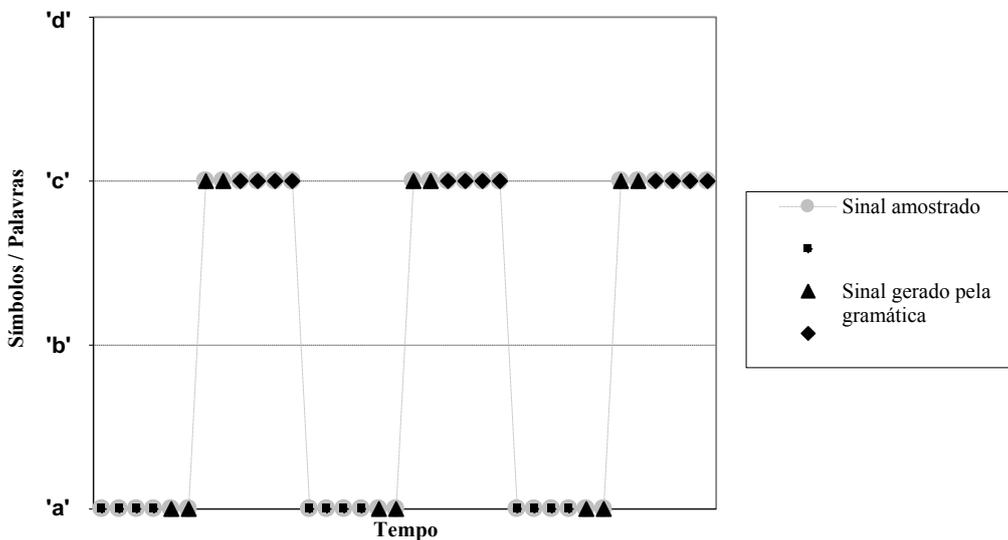
Figura 3.11. Insuficiência das produções gramaticais de símbolos.

Considerando a existência de três palavras ( 3.45 ) na linguagem, o algoritmo de inferência gramatical extrai três *produções do tipo 1* do sistema de rescrita da gramática de palavras ( 3.47 ).

$$\begin{aligned} z_1 &= a a a a \\ z_2 &= a a c c \\ z_3 &= c c c c \end{aligned} \quad ( 3.45 )$$

$$\begin{aligned} z_1 U_k &\rightarrow z_1 z_2 \delta \\ z_2 U_k &\rightarrow z_2 z_3 \delta \\ z_3 U_k &\rightarrow z_3 z_1 \delta \end{aligned} \quad ( 3.46 )$$

Estas produções, que relacionam as anteriores palavras, são agora suficientes para representar a totalidade da linguagem, como se pode observar na Figura 3.12. Nesta figura a sequência de símbolos ‘■’ refere-se à palavra  $z_1$ , a sequência de símbolos ‘●’ a  $z_2$  e a sequência de símbolos ‘◆’ a  $z_3$ .



**Figura 3.12. Produções gramaticais de palavras.**

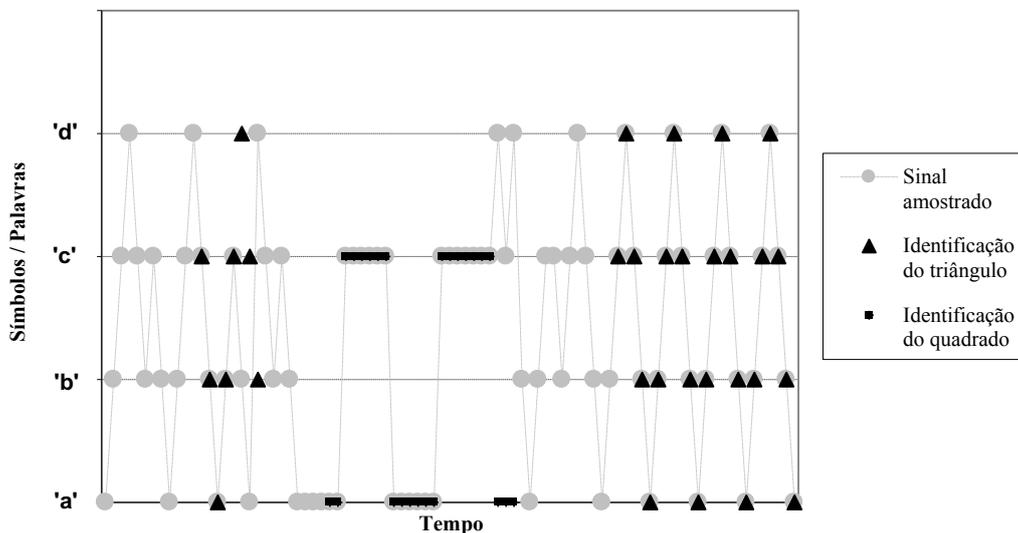
Mantendo a gramática anteriormente inferida considere-se agora um novo sinal, também periódico, cuja codificação resulta triangular e que se apresenta em ( 3.47 ).

Esta sequência de símbolos assume-se como também pertencente à anterior linguagem, pelo que são acrescentados dois símbolos ao alfabeto terminal – ‘b’ e ‘d’ –, e oito novas palavras às já existentes (através da aplicação do algoritmo de inferência gramatical).

$$Y_3 = \{a b c d c b a b c d c b a b c d c b a b c d c b a b c d c b a b c d c b \dots\} \quad (3.47)$$

Após um processo de aprendizagem são geradas mais três produções gramaticais relacionando palavras. Assim a gramática obtida deverá representar uma linguagem onde são válidas tanto a sequência ( 3.44 ) com a sequência ( 3.47 ), podendo deste modo identificar qualquer um dos padrões definidos pelas sequências anteriores.

Gerando uma sequência aleatória onde são introduzidas secções triangulares e quadradas, respectivamente retiradas de  $Y_2$  e  $Y_3$ , verifica-se que estes padrões característicos são identificados pela gramática anteriormente extraída, como se pode verificar na Figura 3.13. Estas produções gramaticais que relacionam palavras numa dada linguagem podem assim ser usadas para identificar características típicas nessa mesma linguagem. Isto é, padrões típicos de sinais podem ser identificados recorrendo à utilização de linguagens formais.



**Figura 3.13. Identificação das características de um sinal.**

Na figura anterior pode verificar-se que a identificação dos padrões presentes na linguagem resultante da codificação de num dado sinal fica sempre sujeito a um atraso.

Em termos de símbolos terminais do alfabeto de símbolos, este atraso é no máximo igual ao produto do tipo de produção pelo tamanho das palavras consideradas.

Outro processo de introduzir características do sinal ao nível das produções poderá recorrer à introdução de regras semânticas (Duda e Hart, 1973). Nestas, a cada regra gramatical associa-se um atributo que pode ser útil num processo de identificação de padrões. Estas regras apresentam-se na forma ( 3.48 ), onde  $V(\alpha)$  representa os atributos do símbolo  $\alpha$ , que são calculados com base na função  $\nu$ , cujos argumentos são o símbolo  $\beta$  e alguma informação adicional  $V'$ .

$$\alpha \rightarrow \beta \quad ; \quad V(\alpha) \leftarrow \nu(V(\beta), V') \quad ( 3.48 )$$



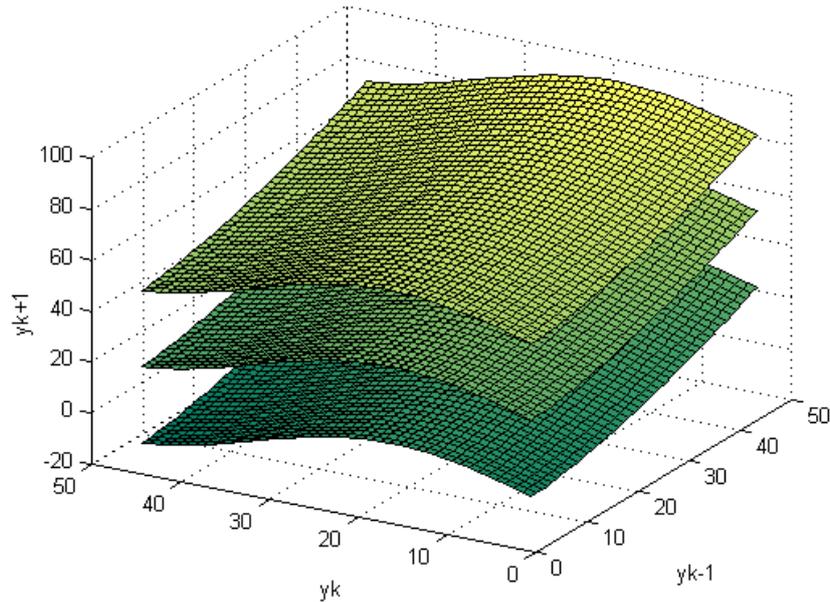
### **3.7 ESTUDO DAS CARACTERÍSTICAS DE UMA GRAMÁTICA INFERIDA A PARTIR DE UM SISTEMA DINÂMICO COMANDADO**

Nesta secção apresentam-se algumas das características do algoritmo de inferência gramatical desenvolvido. No contexto do presente trabalho, a inferência de uma gramática tem como objectivo estabelecer uma representação de um sistema dinâmico em função de uma linguagem que se considera a ele associada.

Para uma mais fácil compreensão e visualização das características das gramáticas inferidas, considere-se o sistema dinâmico não linear ( 3.49 ) que servirá de exemplo à descrição dos processos mais relevantes do algoritmo, como sejam a especificação da própria gramática, o processo de reprodução após interpolação gramatical (preenchimento das lacunas gramaticais), a influência dos exemplos considerados, a capacidade de generalização e a robustez.

$$\begin{cases} x_{k+1} = \alpha \sin(\gamma x_k) + \beta(x_{k+1})^\tau + u_k \\ y_k = x_k \end{cases} \quad ( 3.49 )$$

Na Figura 3.14 apresentam-se as superfícies de evolução da variável de saída do anterior sistema dinâmico, para alguns valores da variável de comando  $u_k \in \{0,30,60\}$ , considerando-se  $\alpha = 10$ ,  $\beta = \gamma = 1/10$  e  $\tau = 3/2$ .

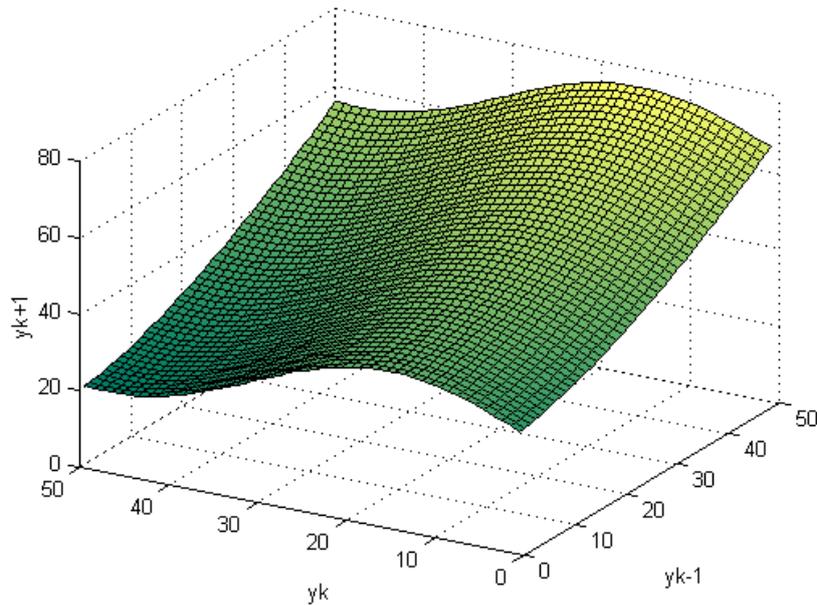


**Figura 3.14. Superfícies de evolução do sistema dinâmico exemplo ( 3.49 ) com  $u_k \in \{0,30,60\}$ .**

### 3.7.1 Especificação da gramática

Como primeiro passo para possibilitar a aplicação do algoritmo de inferência gramatical torna-se necessário especificar os alfabetos da gramática representativa da linguagem gerada pelo sistema dinâmico.

Como informação disponível consideram-se as evoluções das variáveis de saída e de comando. Sem perda de generalidade, e com o único objectivo de visualizar graficamente os resultados, assumir-se-á o comando como constante ( $u_k=30$ ), pelo que as possibilidades de evolução do sistema encontram-se confinadas à superfície apresentada na Figura 3.15.



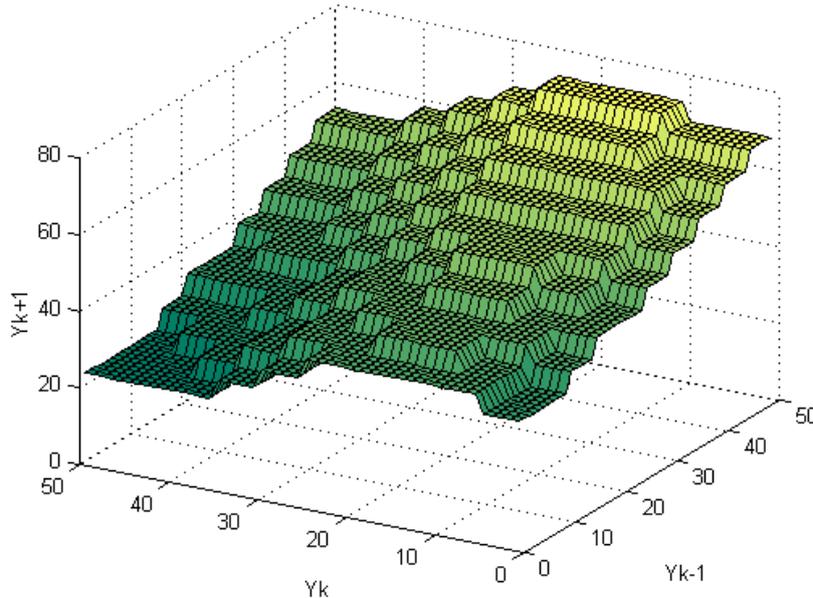
**Figura 3.15. Superfície de evolução do sistema dinâmico exemplo considerando a variável de comando constante ( $u_k=30$ ).**

Optou-se, por razões de simplicidade de compreensão das propriedades do algoritmo, por uma codificação da informação disponível que quantifica a evolução das variáveis presentes. Assim, considerando uma discretização de 16 níveis, o alfabeto terminal –  $\Sigma_T$  – é constituído por 16 símbolos  $y_j$ , referentes à divisão equitativa do domínio da variável de saída entre  $y_{\min}=0$  e  $y_{\max}=80$ . O alfabeto não terminal –  $\Sigma_N$  – reflectirá a evolução da variável de comando. Como se assume que esta é constante, codificada por  $U$ , este alfabeto ( 3.50 ) será composto por 3 símbolos.

$$\Sigma_N = \{S, U, \delta\} \quad ( 3.50 )$$

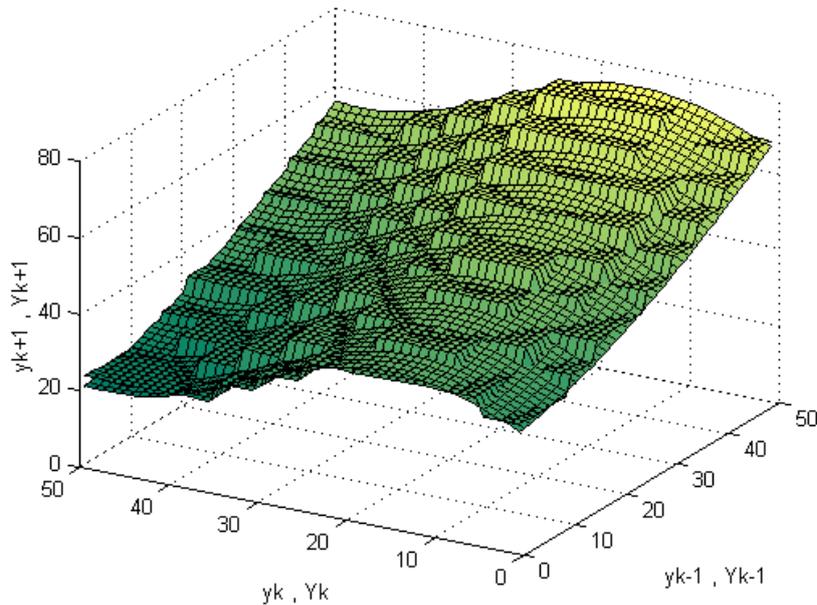
Considerando todo o alfabeto  $\Sigma = \Sigma_T \cup \Sigma_N$ , apresenta-se na Figura 3.16 a evolução da linguagem codificada a partir do sistema dinâmico exemplo. As discontinuidades surgem pelo facto do alfabeto terminal ser construído a partir de uma discretização da evolução da variável de saída. Sendo a evolução dinâmica do exemplo perfeitamente conhecida facilmente se verifica que, com o alfabeto considerado e para o domínio de funcionamento apresentado na Figura 3.15, são necessárias 100 *produções do tipo 2* para representar toda a linguagem no domínio de funcionamento considerado.

Deste modo o sistema de rescrita teórico, designado por  $R_0$ , é constituído por um alfabeto com 19 símbolos e 100 produções do tipo 2. A gramática assim resultante será designada por  $G_0 = \{R_0, \Sigma_N, \Sigma_T, S\}$ .



**Figura 3.16. Linguagem obtida a partir do sistema de rescrita teórico (Gramática  $G_0$ ).**

Na Figura 3.17 pode verificar-se a forma como a linguagem se distribui, de uma forma discreta, em torno da superfície referente ao domínio de funcionamento contínuo do sistema dinâmico. A distribuição da linguagem pode aproximar-se ou afastar-se da evolução real do sistema dinâmico, na medida em que é sempre possível aumentar ou diminuir o nível de discretização do alfabeto, o que certamente influi no grau de complexidade da gramática inferida. A um nível de discretização mais preciso está associado um sistema de rescrita de maior dimensão (alfabeto de maior dimensão e maior número de produções).



**Figura 3.17. Distribuição da superfície de evolução da variável de saída do sistema dinâmico exemplo ( 3.49 ) e respectiva linguagem.**

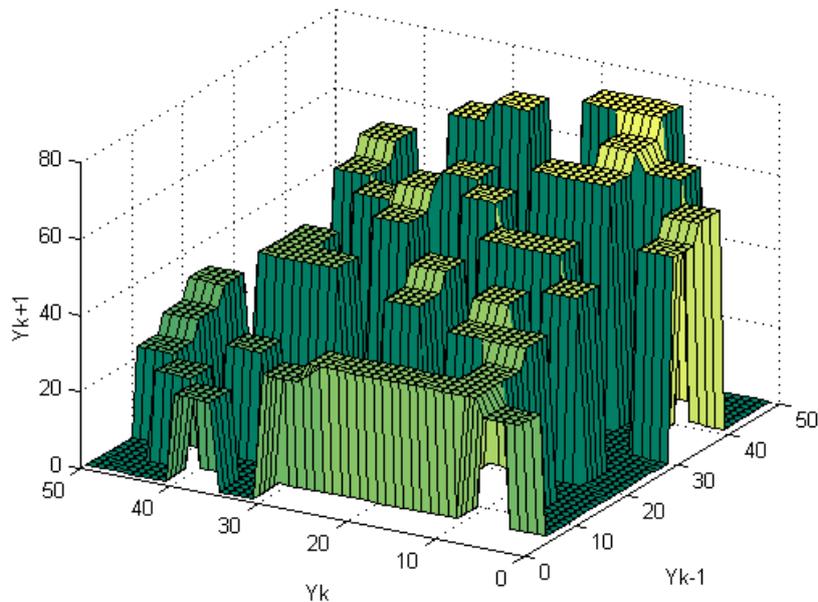
Estando o alfabeto já definido, a especificação do sistema de rescrita da gramática conclui-se com o estabelecimento do conjunto de produções, o que será efectuado por recurso ao algoritmo de inferência gramatical apresentado na secção 3.5.2.

Como a ordem do sistema exemplo é conhecida considera-se  $p_{\max}=2$  como o tipo máximo de produções do sistema de rescrita. Esta escolha é também determinada por limitações de visualização gráfica da distribuição da linguagem.

### **3.7.2 Reprodução e preenchimento das lacunas gramaticais**

Como amostra finita da linguagem, a ser utilizada pelo algoritmo de inferência gramatical, consideraram-se várias palavras exemplo, aleatórias e de comprimentos diferentes distribuídas pela superfície de evolução do sistema dinâmico. Estas palavras correspondem a um total de 75 símbolos. Após aplicação do algoritmo de inferência gramatical foram inferidas 41 produções a partir dos exemplos apresentados. Considerando este conjunto de produções obtém-se uma gramática designada por  $G_1$ . A

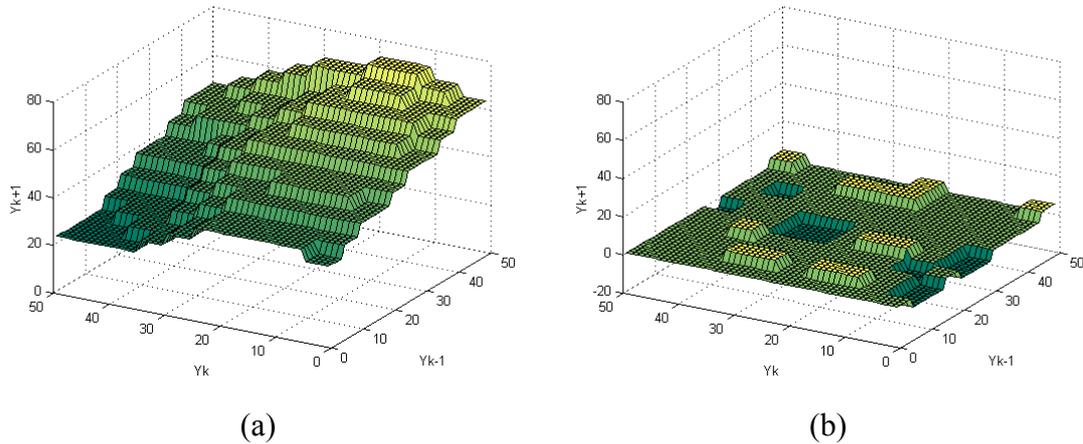
utilização destas produções origina a linguagem apresentada na Figura 3.18. Pode verificar-se que uma zona significativa do domínio de funcionamento do sistema dinâmico considerado não se encontra reproduzido por impossibilidade de aplicação de alguma produção, traduzindo-se na existência de um número significativo de lacunas gramaticais (zonas do domínio de funcionamento não cobertas por nenhuma produção).



**Figura 3.18. Linguagem obtida a partir da gramática  $G_1$ .**

A aplicação do mecanismo de interpolação gramatical, em torno de uma vizinhança de cada lacuna, tal como descrito na secção 3.5.3, permite preencher essas lacunas gramaticais aumentando a dimensão da gramática mas também a sua capacidade de reprodução da linguagem original, tal como se apresenta na Figura 3.19 (a). Este aumento da dimensão da gramática é consequência do aumento do número de produções. Designar-se-á esta nova gramática, resultante da aplicação do mecanismo de interpolação gramatical à gramática obtida por aplicação do algoritmo de inferência gramatical a um conjunto de 75 símbolos agrupados em várias palavras exemplo (Gramática  $G_1$ ), por  $G_2$ .

Na Figura 3.19 (b) apresenta-se a diferença entre a linguagem obtida, através da utilização de  $G_2$ , e a obtida a partir do sistema de rescrita teórico  $R_0$ . Esta diferença é nula na quase totalidade do domínio de funcionamento, facto que resulta da grande semelhança entre ambas as linguagens.



**Figura 3.19. (a) Linguagem obtida a partir da gramática  $G_2$ ; (b) Diferença entre as linguagens obtidas a partir das gramáticas  $G_2$  e  $G_0$ .**

Quando o interesse da aplicação da gramática inferida passar pela obtenção de uma reprodução mais precisa da evolução do sistema dinâmico, e não querendo aumentar a dimensão da gramática por aumento da dimensão dos alfabetos, poder-se-á recorrer a um simples mecanismo de interpolação numérica habitualmente utilizado. A uma qualquer evolução da variável de saída corresponde uma codificação no alfabeto terminal que nunca será perfeita, pois a evolução forçosamente não coincidirá com a codificação considerada. Deste modo, cada valor estará próximo de um símbolo do alfabeto terminal sem ser exactamente igual a ele.

Considere-se a evolução  $\{y_1, y_2, y_3\}$  da variável de saída do sistema dinâmico codificada, no alfabeto terminal, sob a forma ‘ $y_a y_b y_c$ ’. Se no processo de obtenção da linguagem for considerada apenas a produção resultante da codificação, dada por ( 3.51 ) considerando uma produção do tipo 2, existirá sempre um erro associado à própria discretização do alfabeto.

$$y_a y_b U \rightarrow y_a y_b y_c \delta \quad ( 3.51 )$$

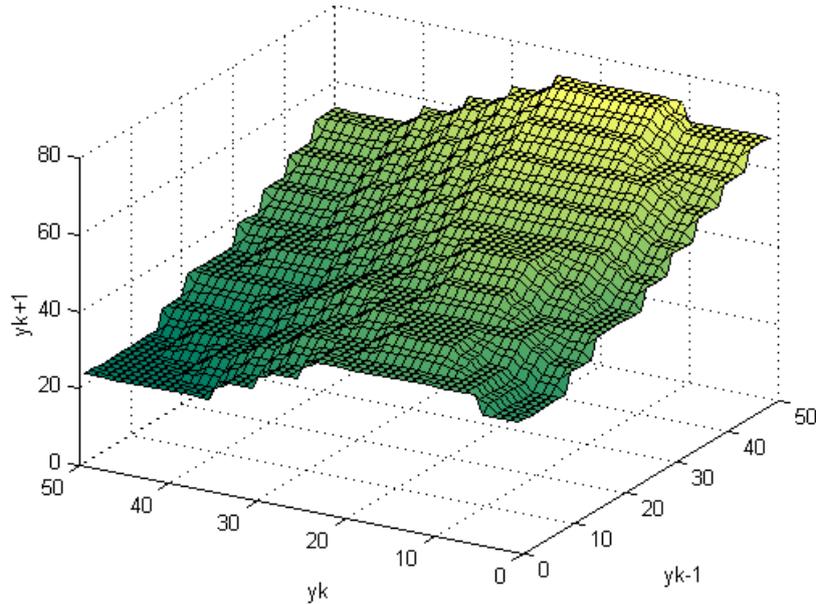
Se, por hipótese, o processo de codificação do alfabeto terminal não conduzir a ‘ $y_a y_b$ ’, mas aos símbolos do alfabeto terminal imediatamente seguintes, poderiam ser consideradas as produções ( 3.52 ).

$$\begin{aligned} y_{a+1} y_b U &\rightarrow y_{a+1} y_b y_d \delta \\ y_a y_{b+1} U &\rightarrow y_a y_{b+1} y_e \delta \end{aligned} \quad ( 3.52 )$$

Deste modo a fórmula de interpolação numérica considerada ( 3.53 ) efectua uma média ponderada do resultado da produção ( 3.51 ) com as produções “próximas” ( 3.52 ). Na fórmula de interpolação o asterisco referencia o valor assumido para a variável de saída obtida por um processo inverso à codificação.

$$y_{c\text{ponderado}}^* = y_c^* + (y_1 - y_a^*)(y_d^* - y_c^*) + (y_2 - y_b^*)(y_e^* - y_c^*) \quad ( 3.53 )$$

Na Figura 3.20 apresenta-se a superfície de evolução resultante da aplicação do anterior processo de interpolação numérica à linguagem obtida a partir do sistema de rescrita teórico – gramática  $G_0$ . Esta evolução corresponde ao melhor desempenho possível, obtido para um perfeito conhecimento de toda a linguagem obtida a partir do sistema dinâmico exemplo ( 3.49 ).

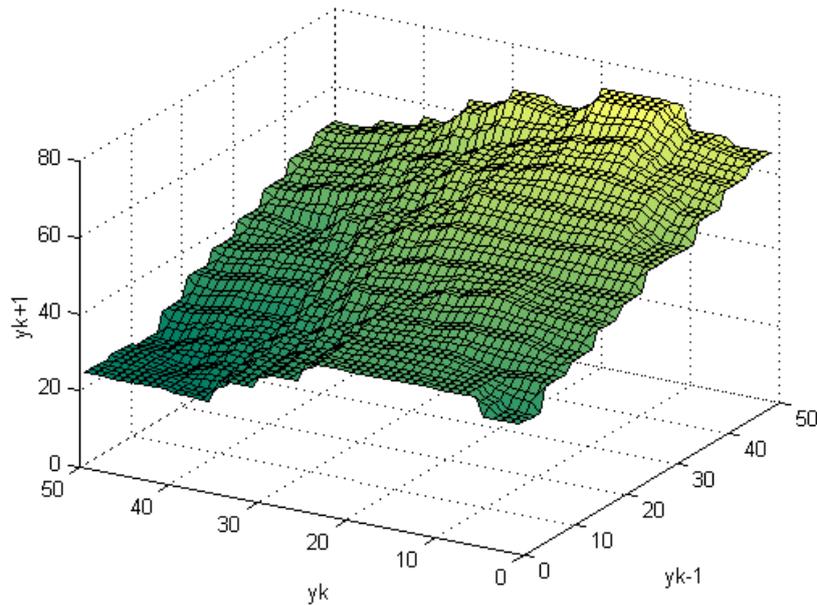


**Figura 3.20 Evolução, após interpolação numérica, obtida a partir da linguagem gerada pela gramática  $G_0$ .**

Por forma a quantificar os desempenhos das gramáticas, considerou-se como possível factor de desempenho o erro quadrático médio ( 3.54 ), onde  $y_{k+1}$  denota a evolução real do sistema dinâmico,  $y_{k+1}^*$  a codificação inversa do valor resultante da aplicação do conjunto de produções inferidas e  $n$  o número de símbolos considerados. O erro quadrático mínimo obtém-se para a evolução apresentada na Figura 3.20 (proveniente do sistema de rescrita teórico – Gramática  $G_0$ ), e vale 1,15.

$$E = \frac{1}{n} \sum_{i=1}^n (y_{k+1} - y_{k+1}^*)^2 \quad (3.54)$$

Na Figura 3.21 apresenta-se a superfície de evolução obtida aplicando o mecanismo de interpolação numérica à linguagem gerada pela gramática  $G_2$ . Como se pode verificar os erros associados à discretização, quando se considera uma codificação do alfabeto terminal associada à evolução dos valores assumidos pela variável de saída, resultam bastante atenuados, traduzindo-se num erro quadrático médio de 1,49.

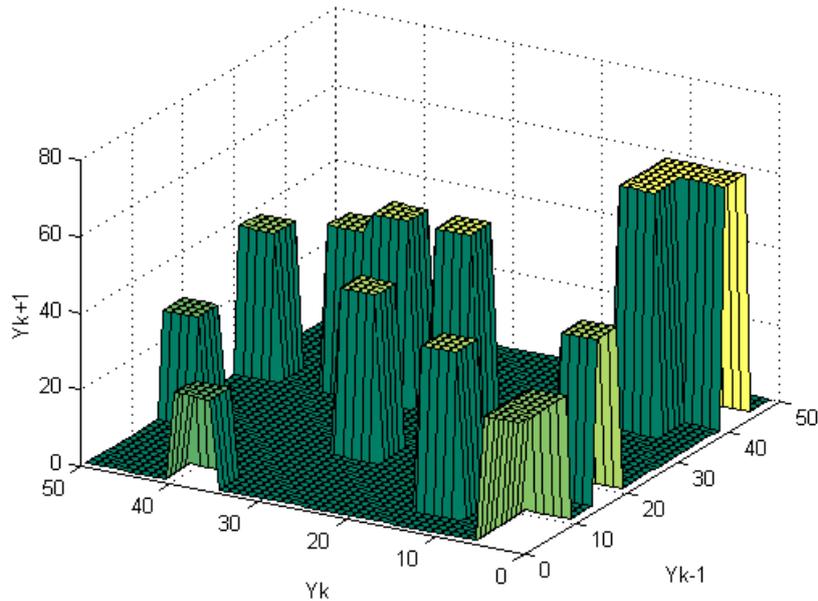


**Figura 3.21** Evolução, após interpolação numérica, obtida a partir da linguagem gerada pela gramática  $G_2$ .

### **3.7.3** Influência do conjunto de treino

Para verificar a influência do conjunto de palavras consideradas no processo de aprendizagem, constituiu-se um conjunto de exemplos de qualidade inferior ao considerado na secção anterior. Estas novas palavras, também elas distribuídas pela superfície de evolução do sistema dinâmico, totalizam um total de 25 símbolos.

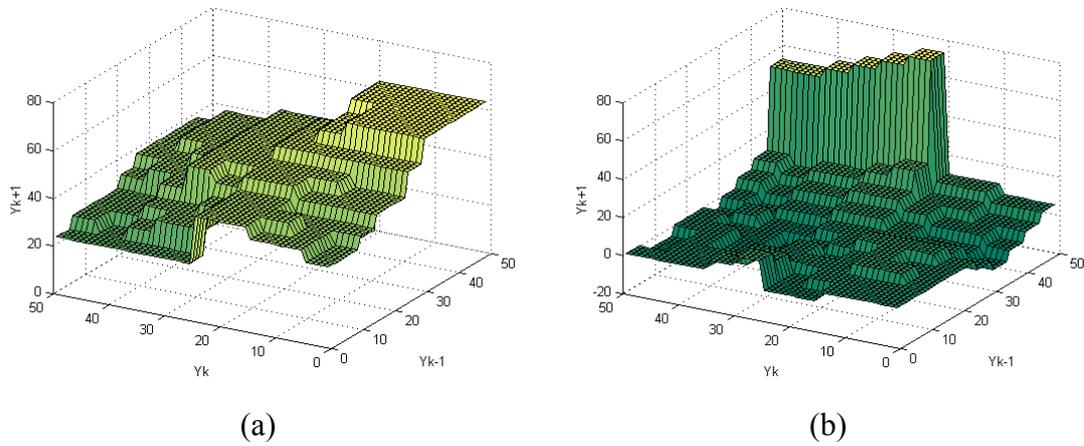
O número de produções inferidas pelo algoritmo de aprendizagem é agora cerca de 34% inferior ao anteriormente obtido, quando se consideraram 75 símbolos nas palavras exemplo distribuídas pela superfície de evolução. Considerando este novo conjunto de produções obtém-se uma nova gramática designada por  $G_3$ . A utilização destas produções traduz-se na linguagem, associada ao sistema dinâmico exemplo, apresentada na Figura 3.22.



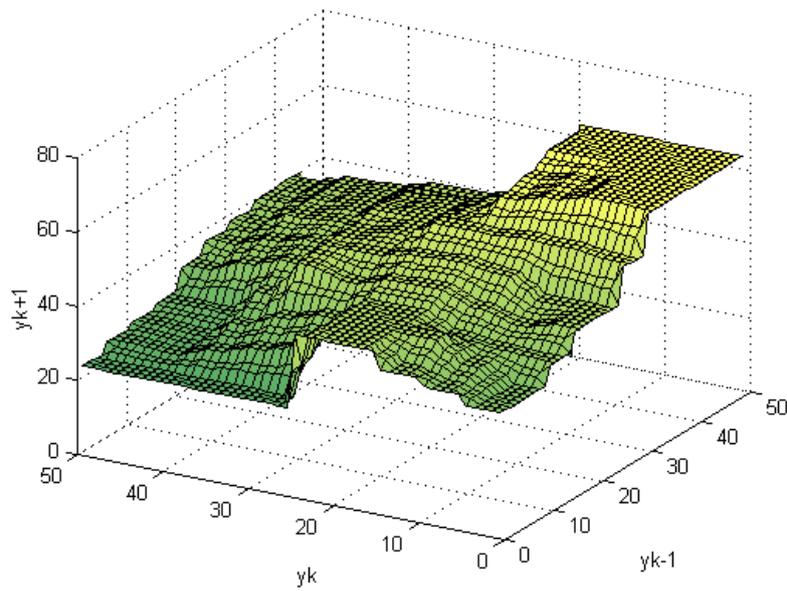
**Figura 3.22. Linguagem obtida a partir da gramática  $G_3$ .**

Referente a esta nova gramática, apresenta-se na Figura 3.23 (a) a respectiva linguagem após a aplicação do mecanismo de interpolação gramatical. Este mecanismo implica um aumento das produções, designando-se esta nova gramática por  $G_4$ . O menor número de produções inicialmente inferido origina necessariamente maiores diferenças entre a linguagem proveniente de  $G_4$  e a obtida a partir do sistema de rescrita teórico, como se pode verificar na Figura 3.23 (b).

Enquanto que anteriormente, com a gramática  $G_2$ , o erro quadrático médio obtido foi de 1,49, agora este sobe para 7,07. A correspondente reprodução numérica apresenta-se na Figura 3.24, onde é visível a óbvia degradação no desempenho. Esta degradação deve-se ao facto de ter sido inferido um número inferior de produções, deixando a gramática sem capacidade de reproduzir correctamente muitas das palavras pertencentes à linguagem associada ao sistema dinâmico ( 3.49 ).



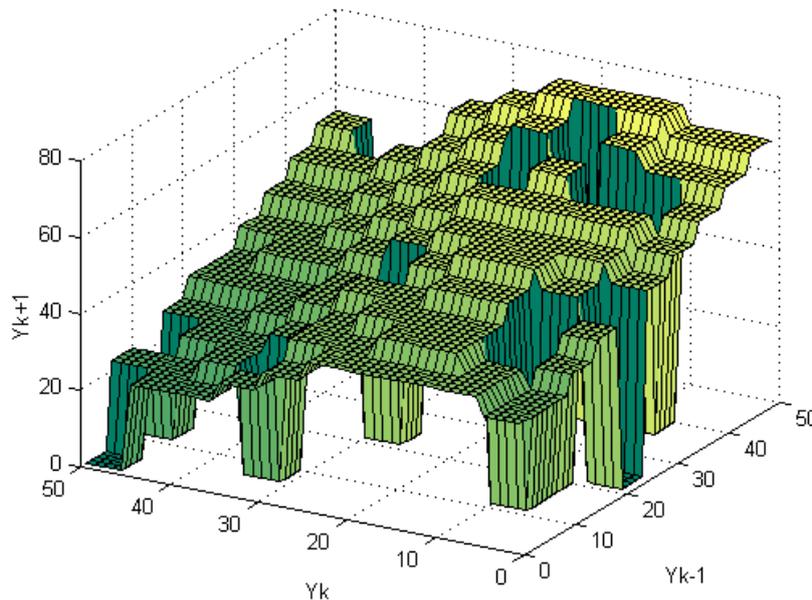
**Figura 3.23. (a) Linguagem obtida a partir da gramática  $G_4$ ; (b) Diferenças entre as linguagens obtidas a partir das gramáticas  $G_4$  e  $G_0$ .**



**Figura 3.24 Evolução, após interpolação numérica, obtida a partir da linguagem gerada pela gramática  $G_4$ .**

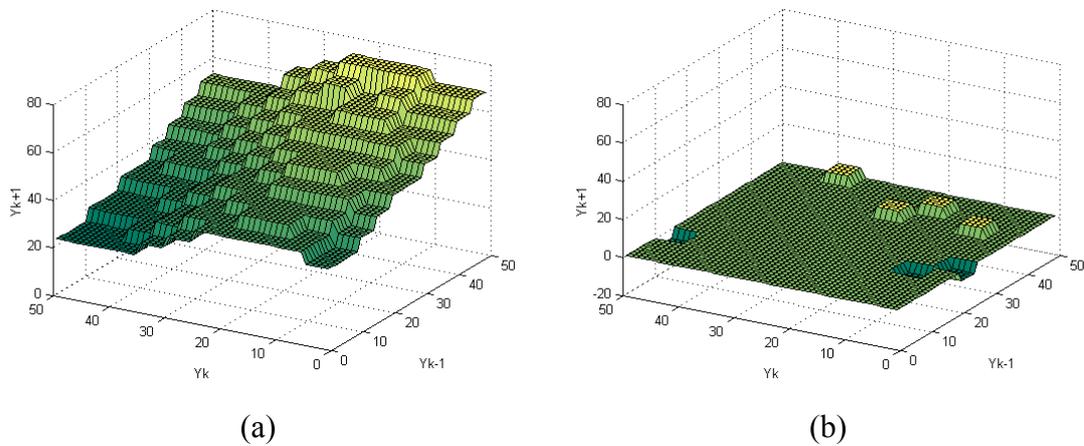
Como exemplo dual, considere-se agora um conjunto de treino constituído a partir de um maior número de palavras exemplo, também elas distribuídas pela superfície de evolução do sistema dinâmico apresentada na Figura 3.15, totalizando 300 símbolos.

O número de produções inferidas pelo algoritmo de aprendizagem é agora cerca de 47% superior ao anteriormente obtido, onde as palavras exemplo distribuídas pela superfície de evolução totalizavam 75 símbolos. A utilização destas novas produções possibilita a inferência de uma nova gramática, designada por  $G_5$ , que origina a linguagem apresentada na Figura 3.25.



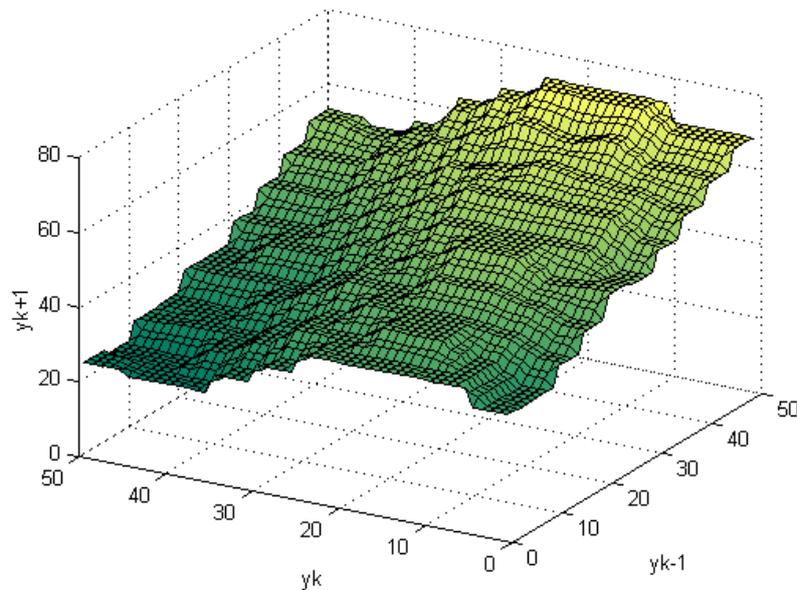
**Figura 3.25. Linguagem obtida a partir da gramática  $G_5$ .**

O considerável aumento dos exemplos presentes no treino diminui de uma forma óbvia as zonas do domínio onde não existem produções inferidas (lacunas gramaticais). No entanto, a partir desta nova gramática, e por aplicação do mecanismo de interpolação gramatical, é possível ainda obter uma nova gramática, designada por  $G_6$ , onde são preenchidas as poucas lacunas gramaticais existentes em  $G_5$ . Apresenta-se na Figura 3.26 (a) a linguagem gerada por esta nova gramática, e na Figura 3.26 (b) a diferença entre esta e a obtida a partir de  $G_0$ . Comparando com os resultados obtidos a partir de  $G_2$  (Figura 3.19) pode verificar-se que o aumento do número de produções não introduz melhorias significativas.



**Figura 3.26. (a) Linguagem obtida a partir da gramática  $G_6$ ; (b) Diferença entre as linguagens obtidas a partir das gramáticas  $G_6$  e  $G_0$ .**

A aplicação do mecanismo de interpolação numérica origina a evolução apresentada na Figura 3.27. Comparando com o valor de 1,49 obtido com a gramática  $G_2$  (resultante da aplicação do mecanismo de interpolação gramatical à gramática obtida a partir de um conjunto de palavras exemplo que totalizavam 75 símbolos) o erro quadrático médio baixou para 1,25, aproximando-se do valor mínimo teórico de 1,15 (obtido a partir da gramática  $G_0$ ).



**Figura 3.27. Evolução, após interpolação numérica, obtida a partir da linguagem gerada pela gramática  $G_6$ .**

Qualquer que seja a dimensão do conjunto de treino considerado, existirão sempre zonas do domínio de funcionamento sem exemplos recolhidos. Estas zonas diminuirão se forem considerados conjuntos de treino de maior qualidade, isto é, apresentando uma maior quantidade e diversidade de exemplos. Contudo, a aplicação do mecanismo de interpolação gramatical leva a que o aumento da quantidade de exemplos não origine necessariamente uma melhoria significativa da qualidade da linguagem reproduzida, isto é, a obtenção de uma melhor gramática. Entende-se por melhor gramática aquela que origina uma linguagem o mais semelhante possível aquela que provem da codificação das variáveis do sistema dinâmico.

Como se viu nos exemplos anteriores, o aumento do número de palavras exemplo utilizadas no processo de inferência gramatical (aumento de 300% na gramática  $G_4$  face a  $G_2$ ), traduz-se numa diminuição do erro quadrático médio de apenas 16%, à custa de um aumento de 47% nas produções inicialmente inferidas.

Por outro lado, diminuições drásticas do conjunto de treino não conseguem ser compensadas pelo mecanismo de interpolação gramatical. A anterior diminuição das palavras exemplo (diminuição 66% na gramática  $G_6$  face a  $G_2$ ) traduz-se num aumento de 375% no valor do erro quadrático médio.

### **3.7.4 Capacidade de generalização**

Os resultados apresentados anteriormente denotam já uma boa capacidade de generalização apresentada pelas gramáticas inferidas, desde que os conjuntos de treino apresentem um número de exemplos razoável. No entanto, esta capacidade evidencia algum carácter local em torno dos exemplos fornecidos durante o processo de aprendizagem.

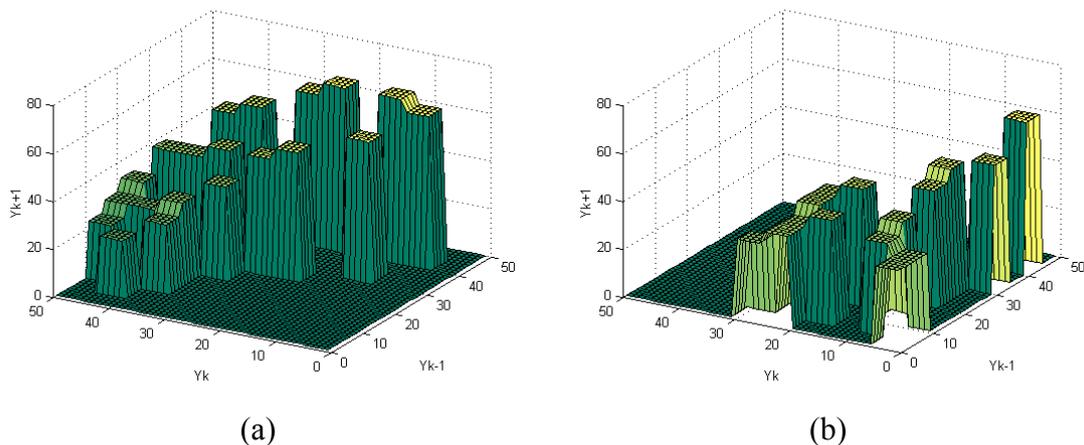
Por forma a testar as características de generalização da gramática inferida numa perspectiva mais global, estabeleceram-se dois conjuntos distintos de exemplos. O primeiro é constituído por palavras que se reportam apenas a evoluções do sistema dinâmico exemplo ( 3.49 ) condicionadas por ( 3.55 ) com  $\xi=44$ , o que corresponde a considerar uma diagonal no plano  $y_{k+1}=0$ .

$$y_k \geq \xi - y_{k-1} \quad ( 3.55 )$$

O segundo conjunto é formado por exemplos complementares, cujas palavras se reportam à evolução do sistema dinâmico condicionado por ( 3.56 ).

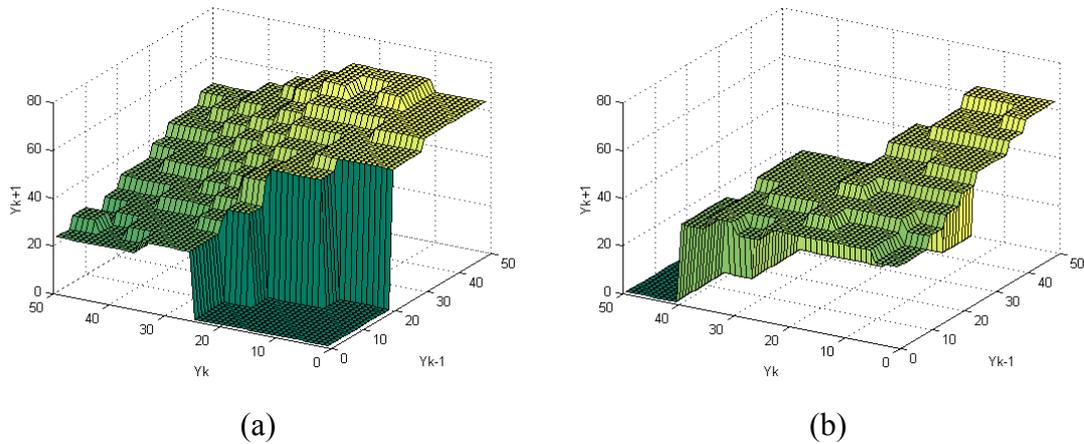
$$y_k \leq \xi - y_{k-1} \quad ( 3.56 )$$

A utilização das produções inferidas pelo algoritmo de aprendizagem, para cada um dos conjuntos anteriores, traduz-se nas linguagens (associadas ao sistema dinâmico exemplo) apresentadas na Figura 3.28. A gramática resultante da aplicação do algoritmo de inferência gramatical aos exemplos condicionados por ( 3.55 ) designar-se-á por  $G_7$ , e a gramática obtida a partir dos exemplos condicionados por ( 3.56 ) designar-se-á por  $G_8$ .



**Figura 3.28. Linguagens obtidas a partir das gramáticas: (a)  $G_7$ ; (b)  $G_8$ .**

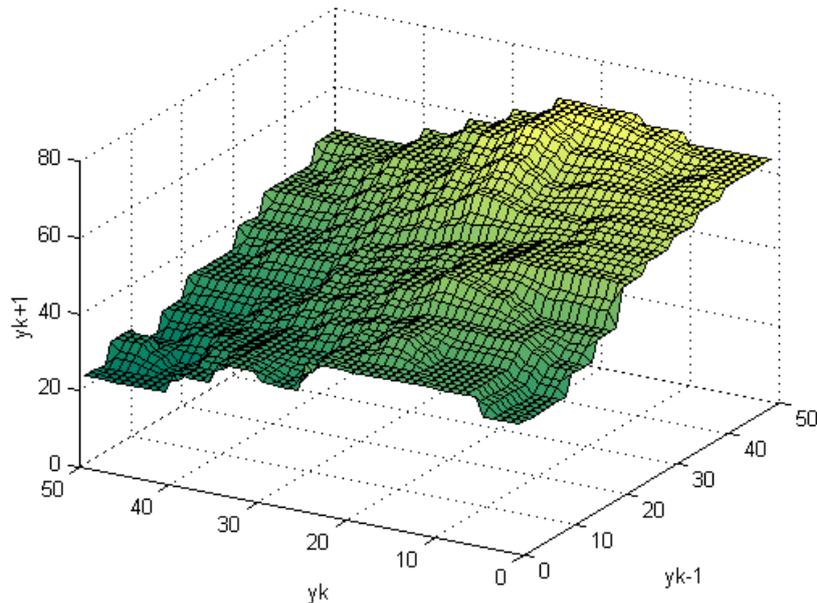
Apresenta-se na Figura 3.29 as linguagens estabelecidas por recurso ao mecanismo de interpolação gramatical com base nos anteriores sistemas de rescrita, pertencentes às gramáticas  $G_7$  e  $G_8$ . A aplicação deste mecanismo aumenta o número de produções, por forma a preencher as lacunas gramaticais, evoluindo a gramática  $G_7$  evolui para uma nova gramática  $G_9$  e a gramática  $G_8$  para  $G_{10}$ .



**Figura 3.29. Linguagens obtidas a partir das gramáticas: (a)  $G_9$ ; (b)  $G_{10}$ .**

Verifica-se que o processo de aprendizagem evidencia um carácter local, não existindo generalização a todo o plano  $y_{k+1}=0$ . Ocorre uma degradação do desempenho que coincide com a aproximação da fronteira definida por ( 3.55 ) e ( 3.56 ). Assim ambas as gramáticas podem ser entendidas como gramáticas locais que não reflectem as características de toda a linguagem, mas apenas de um seu subconjunto.

Considerando que as anteriores gramáticas originam sub-linguagens da linguagem original, designadas respectivamente por  $L_9$  e  $L_{10}$ , e concatenando estas duas linguagens obtém-se uma nova linguagem –  $L' = L_9L_{10}$  – bastante próxima da original  $L_0$  (gerada pela gramática  $G_0$ ). Na Figura 3.30 apresentam-se os resultados da aplicação do método de interpolação numérica à evolução desta nova linguagem  $L'$ .



**Figura 3.30. Evolução, após interpolação numérica, obtida a partir da linguagem  $L'$ , resultante da concatenação das linguagens  $L_9$  e  $L_{10}$ .**

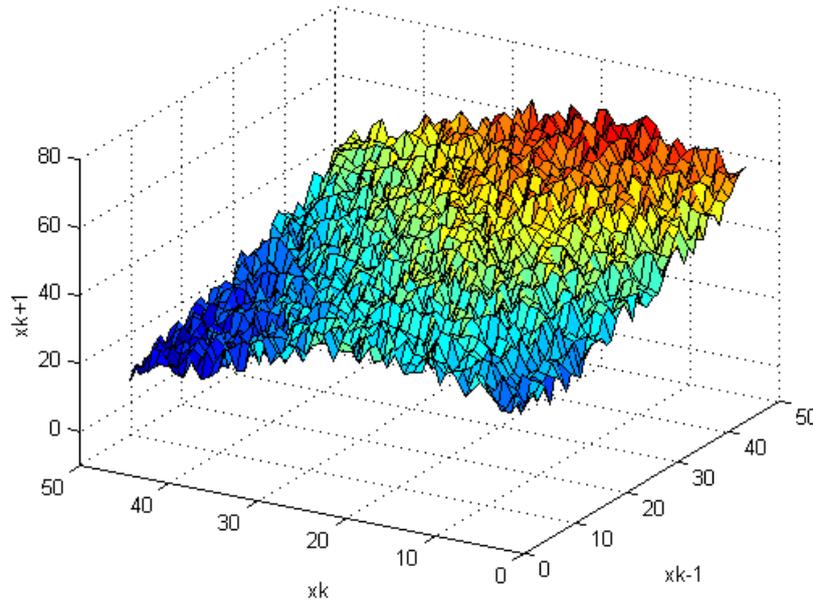
Como se pode verificar, a nova gramática resultante constitui uma boa representação da linguagem associada ao sistema dinâmico, conseguindo atingir um erro quadrático médio bastante aceitável de 2,36. Uma gramática global que represente toda a linguagem original pode assim ser obtida pela concatenação de várias gramáticas locais.

### 3.7.5 Robustez

A aquisição de dados num sistema dinâmico experimental é geralmente um processo fortemente sujeito à influência de ruído. Assim a robustez, isto é, a capacidade de rejeição de informação não fundamental, apresenta-se como um aspecto importante quando se pretende aplicar experimentalmente algoritmos de inferência gramatical.

Para avaliar a capacidade de rejeição de ruído no algoritmo considerado corrompeu-se a evolução do sistema dinâmico com um valor aleatório situado entre  $\pm 5\%$  do valor máximo da variável de saída, originando a superfície de evolução com a forma apresentada na Figura 3.31.

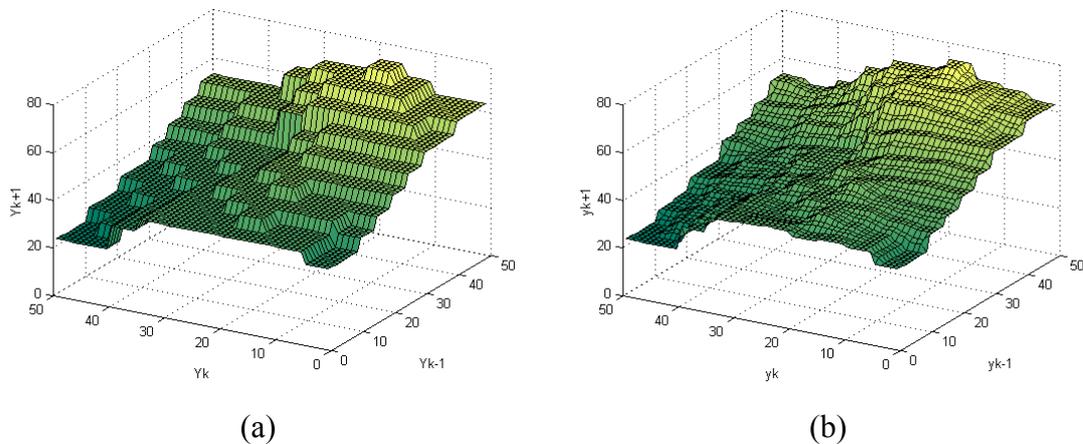
A aplicação do algoritmo de inferência gramatical foi efectuada sobre uma amostra aleatória de palavras obtidas a partir do sistema dinâmico original, num total de 75 símbolos, sendo a adição de ruído efectuada apenas no acto de recolha de cada valor. Este procedimento originou uma gramática designada por  $G_{11}$ .



**Figura 3.31. Superfície de evolução do sistema dinâmico considerando o variável de saída corrompida por ruído ( $u_k=30$ ).**

A presença de ruído vai perturbar os resultados obtidos de uma forma pouco acentuada. A partir da gramática  $G_{11}$ , e por aplicação do mecanismo de interpolação gramatical, obtém-se uma nova gramática, designada por  $G_{12}$ , onde são preenchidas as lacunas gramaticais existentes em  $G_{11}$ . Apresenta-se na Figura 3.32 (a) a linguagem gerada por esta nova gramática.

Por aplicação do mecanismo de interpolação numérica obtém-se a evolução apresentada na Figura 3.32, para a qual o erro quadrático médio atingido foi de 2,37.



**Figura 3.32. (a) Linguagem obtida a partir da gramática  $G_{12}$ ; b) Evolução, após interpolação numérica, obtida a partir da linguagem gerada pela gramática  $G_{12}$ .**

O maior problema que se coloca com a existência de ruído reside no facto de poderem ser inferidas produções que não são características do sistema mas sim resultado do ruído. Uma solução que minimiza este problema reside na forma como está implementado o algoritmo de inferência gramatical, onde apenas são consideradas válidas aquelas produções que aparecem repetidas vezes durante o processo de aprendizagem. O limiar de aceitação encontra-se dependente da dimensão da amostra utilizada na aprendizagem e do nível de ruído estimado. Este procedimento, onde se mantêm apenas as produções que ocorrem um determinado número de vezes nos exemplos de treino, assegura a estabilidade e robustez do algoritmo, actuando como um filtro de rejeição do ruído.

A capacidade de rejeição do ruído encontra-se também dependente do alfabeto terminal considerado, existindo codificações mais influenciáveis pelo ruído do que outras. Se se considerarem discretizações largas as variações resultantes do ruído tendem a oscilar mais dentro da banda de discretização sem influenciarem muito negativamente a inferência da gramática. Isto apresenta-se como uma vantagem em considerar gramáticas de menor dimensão que, apesar de possuírem um menor alfabeto, representem de uma forma aceitável a evolução qualitativa do sistema dinâmico.

## **3.8 CONCLUSÕES**

Este capítulo teve como objectivo apresentar a especificação de uma gramática que represente a evolução de um sistema dinâmico comandado, bem como apresentar o algoritmo de inferência gramatical desenvolvido.

A gramática representativa da linguagem gerada pelo sistema dinâmico foi especificada com base na teoria formal da linguagem. O alfabeto terminal é constituído por símbolos associados à codificação, qualquer que ela seja, das variáveis de saída. Por outro lado, o alfabeto não terminal é constituído por símbolos resultantes da codificação das variáveis de comando. As produções do sistema de rescrita traduzem a evolução do sistema dinâmico, no que se refere à evolução da saída do sistema em função da sua evolução anterior e do comando considerado.

Com base nesta gramática desenvolveu-se um algoritmo de inferência gramatical responsável por estabelecer as produções a partir de um conjunto finito de palavras exemplo, pertencentes à linguagem gerada pelo sistema dinâmico. Este algoritmo começa por considerar produções de tipo inferior que vão sendo substituídas por produções de tipo superior, à medida que a informação sobre o sistema aumenta e que aquelas se vão tornando insuficientes para descrever a dinâmica da linguagem associada ao sistema.

A consideração de produções que relacionem palavras, e não apenas símbolos, apresenta-se como uma ferramenta útil quando se consideram sinais periódicos. Permite, por um lado diminuir o número de produções necessárias, e por outro mais facilmente identificar padrões típicos num dado sinal.

Quando o conjunto de palavras exemplo é limitado ocorre o aparecimento de lacunas gramaticais, pelo facto de produções do sistema de rescrita que não terem sido inferidas durante o processo de aprendizagem. Além da solução óbvia de aumentar o conjunto de treino, apresentou-se um mecanismo de interpolação gramatical para minimizar o problema. Este mecanismo baseia-se no cálculo da distância entre palavras, por forma a estimar as produções inexistentes como uma média ponderada das já existentes. Os

resultados apresentados evidenciam a melhoria do desempenho das gramáticas após a aplicação deste processo de inferência gramatical.

A aplicação das gramáticas inferidas a partir de um conjunto de palavras revela a existência de uma aprendizagem eficaz e capacidade de generalização. Esta última característica apresenta, no entanto, carácter local pois encontra-se dependente das palavras utilizadas na fase de aprendizagem. Verificou-se também a possibilidade de constituir uma gramática representativa da linguagem gerada pelo sistema dinâmico a partir da concatenação de várias gramáticas locais.

O desempenho das gramáticas inferidas melhora quando o número de palavras utilizadas no treino aumenta. No entanto, esta melhoria é conseguida à custa do aumento do número de produções inferidas durante o processo de aprendizagem. A influência do ruído encontra-se dependente do alfabeto terminal considerado, sendo que a forma de implementação do algoritmo de inferência gramatical minimiza bastante a sua influência.

## **CAPÍTULO 4 - IMPLEMENTAÇÃO DE LINGUAGENS FORMAIS POR SISTEMAS CONEXIONISTAS**

**Resumo:** Neste capítulo apresenta-se uma implementação conexionista da gramática inferida a partir do comportamento de um sistema dinâmico. A referida implementação possibilita o processo de aprendizagem e trata a informação recebida de uma forma distribuída e paralela, permitindo a redução do tempo de processamento associado à gramática. A estrutura conexionista desenvolvida é constituída por duas sub-estruturas, sendo uma delas responsável pela identificação dos símbolos dos alfabetos, terminal e não terminal, e a outra pelo estabelecimento e aplicação das produções.



## **4.1 INTRODUÇÃO**

A utilização de linguagens formais, no sentido em que foi apresentada em capítulos anteriores, passa obrigatoriamente pela comparação de palavras provenientes do sistema dinâmico – entendido como fonte linguística – com outras sequências de símbolos – presentes nas produções do sistema de rescrita. Esta comparação, executada a partir de um conjunto de sequências de operações de edição, pode ser entendida como um problema de procura num determinado espaço. Como a dimensão do espaço geralmente cresce de uma forma exponencial com a dimensão do problema os recursos computacionais aumentam rapidamente e de uma forma excessiva (Thomason, 1990). Para evitar esta explosão combinatória vários autores correlacionam estruturas lógicas simbólicas com estruturas conexionistas (Barnden e Pollack, 1991) (Gallant, 1993) (Honavar e Uhr, 1994).

A introdução de autómatos na representação de linguagens potencia também a redução do tempo de processamento, uma vez que este passa a ser distribuído e paralelo. O estudo dos autómatos foi iniciado no início da década de 60 (Tsetlin, 1962) e tem sido abordado, desde então, em diferentes contextos. A teoria dos autómatos é extremamente rica e várias técnicas, envolvendo a sua utilização, têm sido utilizadas numa grande variedade de domínios de aplicação, que não se confinam às ciências de engenharia. São exemplos disso as aplicações em reconhecimento de padrões, algoritmos de encriptação, reconhecimento de escrita manual, reconhecimento óptico de caracteres, compressão de dados, indexação, análise de sistemas operativos, dicionários electrónicos, processamento de linguagem natural, compiladores de programas, modelização, bases de dados, entre outros.

O tipo de autómato a utilizar, na representação de uma dada linguagem, está dependente do tipo de linguagem a representar. Considerando a hierarquia de Chomsky clássica é possível considerar a associação apresentada na Tabela 4.I (Howie, 1991).

**Tabela 4.I. Hierarquia de gramáticas estabelecida por Chomsky.**

<b>Tipo de Gramática</b>	<b>Linguagem</b>	<b>Autômato</b>
0	Frase estruturada	Máquina de Turing
1	Sensível ao contexto	Autômato do tipo ' <i>linear bounded</i> '
2	Livre de contexto	Autômato do tipo ' <i>pushdown</i> '
3	Regular	Autômato do tipo finito

Uma máquina de Turing pode ser entendida como um dispositivo de controlo de estados finitos, com uma fita de leitura-escrita potencialmente infinita e não limitada. As palavras entradas (na fita) possuem comprimento finito, mas o seu comprimento não é limitado de nenhuma forma. Em cada instante de tempo, o dispositivo de controlo lê um símbolo na fita e substitui-o por outro, antes de se efectuar um deslocamento unitário da fita. Um Autômato do tipo '*linear bounded*' é basicamente uma máquina de Turing com uma fita limitada por dois caracteres especiais. O autômato não pode mover a fita além desses limites nem pode escrever sobre eles. Um Autômato do tipo '*pushdown*' possui além da fita e do dispositivo de controlo uma memória adicional, a qual vai influenciar a transição de estados. Neste autômato, ao contrário dos anteriores, existe apenas um sentido para o movimento da fita. Um autômato finito, possui apenas uma fita unidireccional e um dispositivo de controlo que lê um símbolo na fita e, dependendo do estado em que se encontra, produz um novo estado, antes de efectuar uma deslocação da fita.

O conceito de autômato é assim muito variado e pode ser descrito por uma larga variedade de sistemas abstractos. Neste capítulo adoptar-se-á a definição considerada em (Narendra e Thathachar, 1989), segundo a qual um autômato é definido por uma variedade ( 4.1 ), na qual a entrada e o estado actuais determinam tanto o estado seguinte como a saída.

$$\{\Theta, A, B, F(.,.), H(.,.)\} \quad (4.1)$$

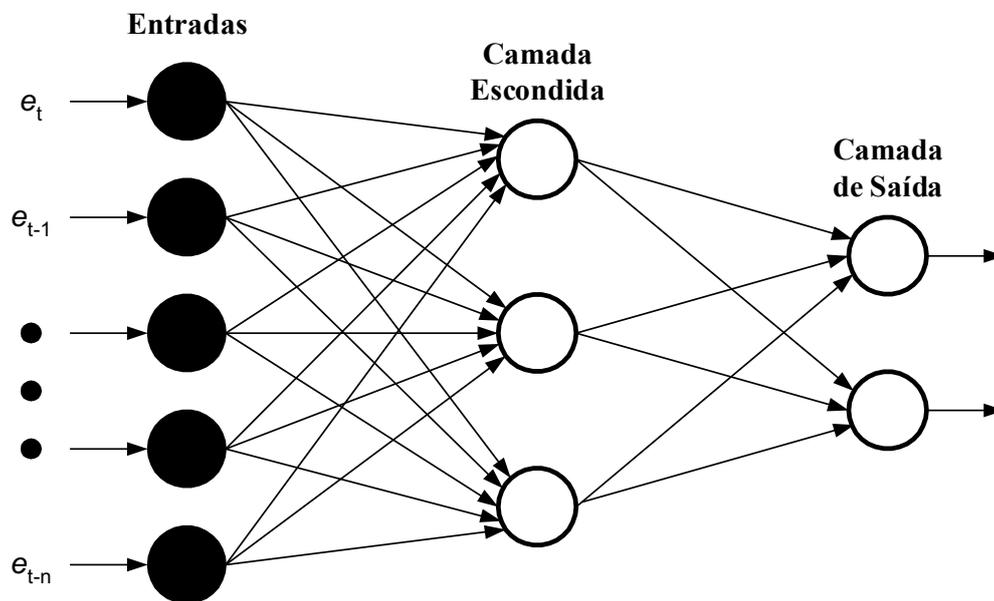
Nesta definição  $\Theta$  representa o conjunto de estados internos,  $A$  o conjunto de saídas,  $B$  o conjunto de entradas,  $F(.,.): \Theta \times B \rightarrow \Theta$  uma função que mapeia o estado e entradas actuais para o estado seguinte, e  $H(.,.): \Theta \times B \rightarrow A$  uma função que mapeia o estado e entradas actuais para a saída actual. Basicamente, o autômato aceita uma sequência de

entradas de modo a gerar uma sequência de saídas, sendo que o estado seguinte e saída actual são determinados pela entrada e pelo estado actual.

Nos modelos conexionistas o processo de aprendizagem é entendido como a determinação dos pesos das ligações sinápticas, entre as várias unidades constituintes do sistema, em função de um processo de treino. Estes pesos são ajustados até que o comportamento do sistema conexionista modelize as propriedades estatísticas das suas entradas. No limite as relações entre os vários estados do sistema conexionista, entendido como uma máquina de estados ou um autómato, replicam as relações entre os acontecimentos do sistema representado. Este conceito de aprendizagem está directamente relacionado com o conceito de *reinforcement learning*, na medida em que a força entre ‘ideias’ é uma função da frequência com que estas são experimentadas, e a força de uma conexão sináptica é função da frequência com que a resposta do sistema conexionista é recompensada em presença de uma dada entrada. O conceito de *reinforcement learning* baseia-se na ideia intuitiva de que se uma acção sobre um sistema resulta numa melhoria do seu estado, então a probabilidade de repetir essa acção deve ser aumentada.

Estes modelos conexionistas são também referidos como de ‘processamento paralelo distribuído’ (McClelland e Rumelhart, 1988) podendo ser utilizados para integrar o processamento simbólico e as redes neuronais, na medida em que providenciam o processamento elementar de informação, presente nas redes neuronais, servindo estas de blocos representativos de regras ou algoritmos simbólicos (Pinker e Prince, 1988). As redes de processamento paralelo distribuído individuais serão responsáveis por processar associações primitivas de símbolos, sendo que a saída de uma rede como entrada de outra funciona como a manipulação de símbolos presente nas regras do processamento simbólico. Uma qualquer entidade é representada como um padrão de activação de um conjunto de unidades e não de uma única unidade. Estas pequenas unidades apresentam conteúdo semântico, pois representam propriedades específicas da entidade (no sentido em que são activadas como resposta a essas propriedades) e a entidade é assim representada apenas em termos das suas propriedades. As ligações sinápticas descrevem a força de associação entre propriedades da entidade.

A maioria das descrições do comportamento de entidades – sejam elas sistemas electromecânicos ou linguagens – encontra-se dependente do conceito de tempo. Mesmo não aparecendo muitas vezes de uma forma explícita o comportamento dinâmico está intimamente associado à evolução temporal. Classicamente a noção de tempo tem sido introduzida em modelos conexionistas adoptando uma representação sequencial temporal das entradas (Haykin, 1999). Esta abordagem, representada na Figura 4.1, tem sido também utilizada na modelização de sistemas de accionamento electromecânicos por recurso a redes neurais, onde as entradas da rede se reportam aos valores assumidos pelas variáveis de interesse ao longo do tempo –  $e_t, e_{t-1}, \dots, e_{t-n}$  (Martins, 1996).



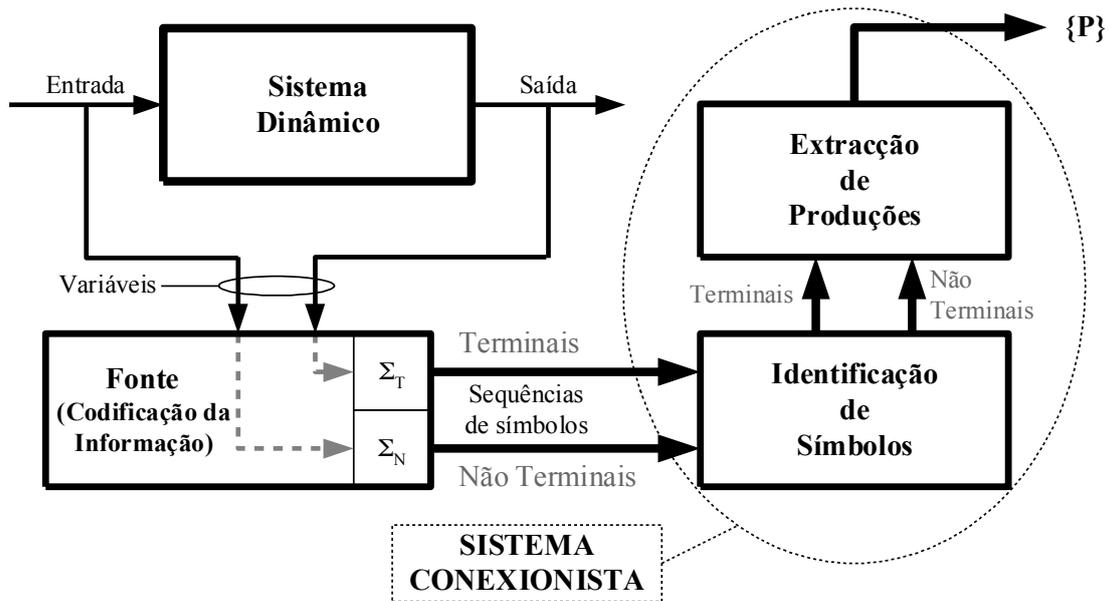
**Figura 4.1 – Uma rede neuronal feed-forward que representa o tempo através do espaço.**

Modelos mais recentes têm sugerido a representação do tempo, quer no processamento quer na aprendizagem, não de uma forma explícita nas entradas do sistema conexionista, mas através dos seus efeitos. Isto é, o estado da rede, denominada de recorrente, é uma função das entradas e do seu estado actual (Haykin, 1999).

Neste capítulo considerar-se-á a associação de um conjunto de autómatos em rede, tomando partido das vantagens de optimização combinatória oferecidas pelos sistemas conexionistas e pela dependência temporal presente nos autómatos. Por outro lado, a

opção por esquemas de *reinforcement learning* permite a utilização de algoritmos de aprendizagem mais robustos, no sentido em que somente as propriedades significativas da entidade descrita são realmente apreendidas.

Deste modo é possível considerar uma estrutura hierárquica de autómatos que sejam capazes de identificar, num primeiro nível, os símbolos dos alfabetos envolvidos, e extrair, num segundo nível, as produções da gramática que representa a linguagem do sistema em estudo. Esta estrutura encontra-se representada na Figura 4.2, e será detalhadamente explicada ao longo do presente capítulo. Considerou-se um tipo de autómato adaptado à estrutura das produções que se assumiu descreverem a linguagem representativa da dinâmica de um sistema dinâmico. Sem detrimento de outras possíveis soluções pretende-se implementar, partindo do nível mais baixo até ao mais elevado, uma estrutura baseada em autómatos que consiga extrair de uma forma automática as referidas produções.



**Figura 4.2 – Sistema conexionista de uma gramática representativa da linguagem gerada por um sistema dinâmico.**



## 4.2 AUTÓMATOS

Como unidades básicas do sistema conexionista optou-se pela utilização de autómatos de estados finitos com duas entradas. Seguindo a definição ( 4.1 ), os conjuntos de estados internos, entradas e saídas do autômato possuem dois elementos  $\Theta, B, A \in \{-1, +1\}$  cujas evoluções são dadas, respectivamente, por ( 4.2 ) e ( 4.3 ), onde  $B_i$  representa uma entrada no autômato,  $w_i$  o peso sináptico associado a cada entrada e  $\rho(\cdot)$  a função escalão.

$$\Theta(t+1) = \rho\left(\sum_{i=1}^2 w_i B_i(t) - 1\right) \quad (4.2)$$

$$A(t) = \Theta(t) \quad (4.3)$$

Estas unidades processam os seus sinais de saída (no caso presente iguais ao seu estado interno) multiplicando cada sinal de entrada pelo respectivo peso sináptico, somando estes valores ponderados, e utilizando o resultado como argumento da função escalão.



### 4.3 PRIMEIRO NÍVEL DO SISTEMA CONEXIONISTA – IDENTIFICAÇÃO DE SÍMBOLOS

O objectivo do primeiro nível do sistema conexionista é proceder a uma identificação dos símbolos terminais e não terminais, presentes nos respectivos alfabetos, constituintes da gramática descritiva da linguagem associada ao sistema dinâmico considerado. A evolução dinâmica da linguagem é, como já visto anteriormente, determinada pela sequência de símbolos não terminais, imposta exteriormente, e pela evolução passada das palavras terminais.

Das várias formas de proceder a uma identificação de símbolos, optou-se por desenvolver uma estrutura conexionista baseada em unidades básicas simples, os autómatos anteriormente descritos, desde o nível mais baixo ao mais elevado. Neste contexto, torna-se necessário considerar os símbolos dos alfabetos terminal e não terminal codificados em  $n$  códigos binários ( $\pm 1$ ), o que se traduz numa existência máxima de  $2^n$  símbolos no respectivo alfabeto. Estabeleceu-se uma rede de autómatos, representada na Figura 4.3, do tipo *feedforward*, com  $n$  camadas, e  $2^i$  autómatos por camada, onde  $i$  denota a camada em causa. Isto perfaz um total inicial de  $\sum_{i=1}^n 2^i$  autómatos, mas cujo número diminuirá após um necessário processo de aprendizagem.

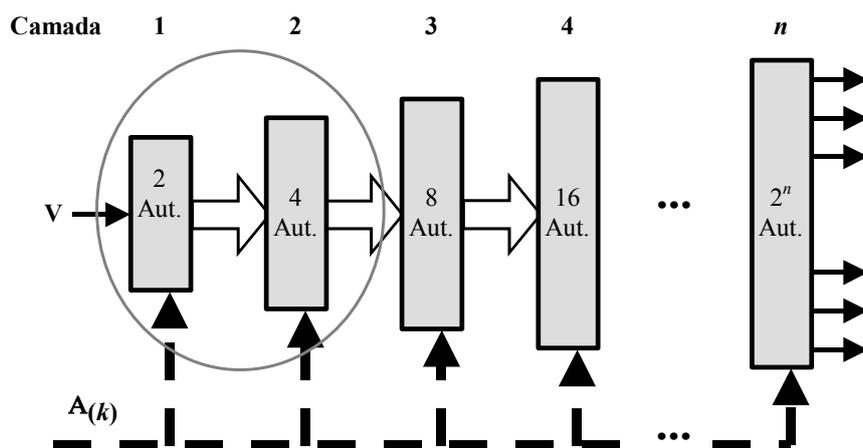


Figura 4.3 – Rede de autómatos para identificação de símbolos.

Os símbolos codificados, representados por  $A(k)$  na Figura 4.3, entram de forma sequencial em todas as camadas da rede. Notar que  $k$  representa o tempo na base dos códigos, isto é, uma palavra de  $m$  símbolos produz uma sequência de  $m \times n$  códigos, após codificação. O sinal  $V$  é um sinal de controlo auxiliar utilizado para indicar à rede que foi iniciada a introdução de uma palavra.

Sempre que o valor de  $k$  for múltiplo de  $n$ , isto é, sempre que toda a codificação de um único símbolo tenha sido fornecida à rede, uma das saídas da rede será activada indicando qual o símbolo em causa. A ausência de saída na última camada da rede traduz o facto de o símbolo não ter sido identificado.

Na Figura 4.4 apresenta-se um diagrama ampliado da zona limitada pelo círculo na Figura 4.3. Todos os pesos sinápticos serão estabelecidos após um processo de aprendizagem, admitindo como limites de variação os estabelecidos em ( 4.4 ).

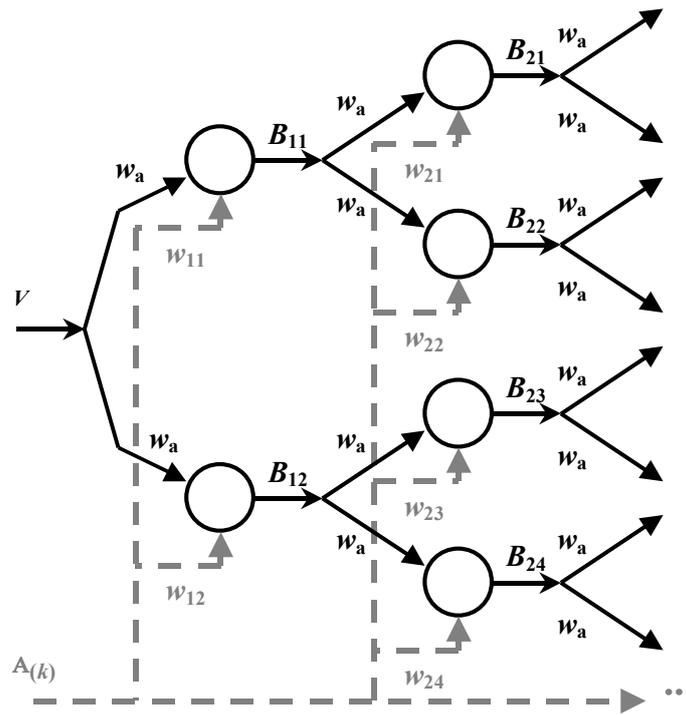


Figura 4.4 - Rede de autómatos para identificação de símbolos – duas primeiras camadas.

$$\begin{cases} w_{ij} \in \left[0, +\frac{1}{2}\right] , & \text{com } i \text{ ímpar} \\ w_{ij} \in \left[-\frac{1}{2}, 0\right] , & \text{com } i \text{ par} \\ w_a = +\frac{1}{2} \end{cases} \quad (4.4)$$

Sendo cada autómato do tipo descrito em ( 4.2 ) e ( 4.3 ), a evolução do seu estado é determinada pelo seu próprio estado e pelas suas entradas. A construção da rede implica que cada autómato só responderá positivamente a um determinado código se o autómato imediatamente anterior providenciou uma resposta também positiva, no instante de tempo anterior. Assim o processo de reconhecimento vai avançando como uma ‘frente de onda’ ao longo da rede, à medida que os códigos referentes a cada símbolo vão sendo fornecidos à rede.

Como facilmente se pode verificar, considerando todas as ligações sinápticas e autómatos possíveis na rede da Figura 4.3, todos os  $2^n$  símbolos possíveis podem ser identificados. No entanto, não é obrigatório que a gramática descritiva da linguagem gerada pelo sistema dinâmico utilize todos símbolos do alfabeto, pelo que será de todo o interesse que a rede de autómatos seja sujeita a um processo de aprendizagem por forma a identificar quais os símbolos efectivamente presentes nas palavras.

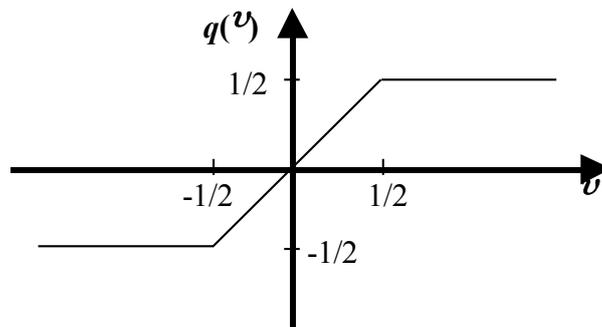
A principal motivação presente no estabelecimento do algoritmo de aprendizagem de símbolos traduz-se no facto de que um determinado autómato só deve aprender quando o anterior tiver efectivamente aprendido no instante anterior. Subjacente a esta motivação está o facto de que para identificar os símbolos presentes se devem aprender sequências de códigos, embebendo desta forma a noção de tempo no conceito de aprendizagem.

As leis de aprendizagem consideradas foram as apresentadas em ( 4.5 ), onde  $A(k)$  representa a entrada em cada autômato correspondente ao símbolo codificado. Consideram-se taxas de aprendizagem  $\gamma_{ap}$  e de esquecimento  $\gamma_{esq}$  distintas bem como a função  $q(\cdot)$ , dada por ( 4.6 ) e representada na Figura 4.5.

$$\begin{cases} w_{ij}(k+1) = q(w_{ij}(k) + \gamma_{ap}A(k)), & \text{se } w_{ij}(k)A(k) > 0 \\ w_{ij}(k+1) = q(w_{ij}(k) - \gamma_{esq}A(k)), & \text{se } w_{ij}(k)A(k) < 0 \end{cases} \quad \text{somente se } B_{i-1}(k) > 0$$

$$\begin{cases} w_{ij}(k+1) = q(w_{ij}(k) + \gamma_{ap}A(k)), & \text{se } w_{ij}(k)A(k) > 0 \\ w_{ij}(k+1) = q(w_{ij}(k) - \gamma_{esq}A(k)), & \text{se } w_{ij}(k)A(k) < 0 \end{cases} \quad \text{somente se } V > 0$$
( 4.5 )

$$q(v) = \begin{cases} -1/2 & ; v \leq -1/2 \\ v & ; -1/2 < v < 1/2 \\ 1/2 & ; v \geq 1/2 \end{cases} \quad ( 4.6 )$$



**Figura 4.5 – Função utilizada na aprendizagem.**

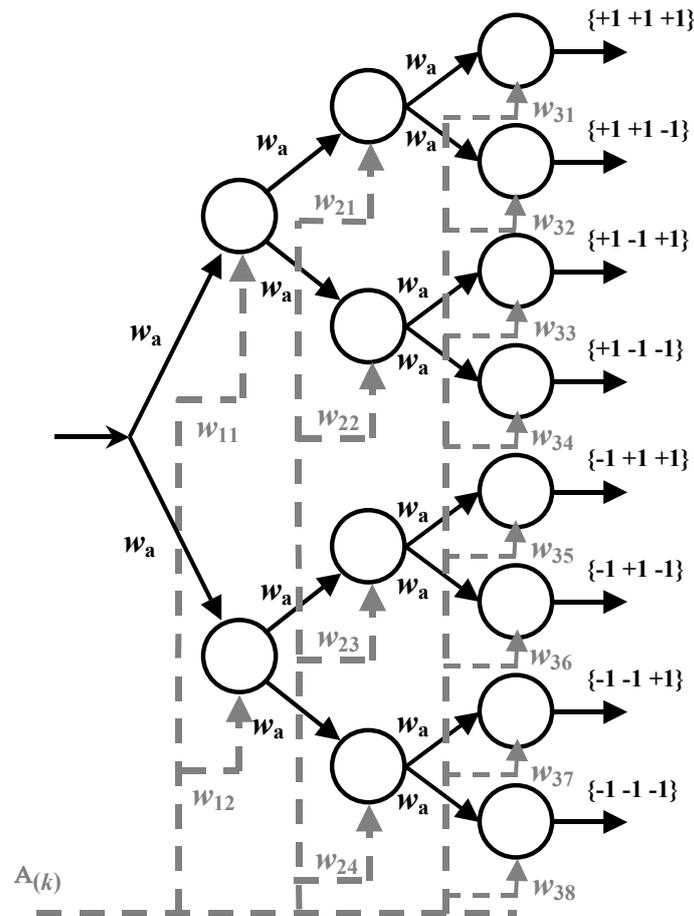
Basicamente, o funcionamento do processo de aprendizagem, que se inicia com uma atribuição nula aos pesos  $w_{ij}$ , pode resumir-se no seguinte: Quando o autômato anterior tiver identificado um código, no instante anterior, possibilita-se a aprendizagem

(evolução de  $w_{ij}$  no sentido de  $\pm 1/2$ ) somente se a entrada actual tiver o mesmo sinal do peso  $w_{ij}$  do respectivo autómato. Por outro lado, considera-se um esquecimento (evolução de  $w_{ij}$  no sentido de 0) se a entrada actual apresentar um sinal diferente de  $w_{ij}$  (indicando que se trata de um código errado e que não deve ser aprendido por esse autómato).

São consideradas duas taxas de evolução dos pesos sinápticos, uma para aprendizagem e outra para esquecimento, necessariamente inferior à primeira. No presente estudo consideraram-se  $\gamma_{ap}=0,004$  e  $\gamma_{esq}=0,002$ . A existência destas duas taxas surge da necessidade de evitar que uma codificação ortogonal, menos frequente, não seja aprendida.

No final do processo de aprendizagem a rede será sujeita a um processo de *prunning*, eliminando todas as ligações sinápticas de todos os autómatos cujos antecedentes não identificaram nenhum código ao longo de todo o processo de aprendizagem, isto é, apresentam pesos sinápticos na banda  $-0,4 < w_{ij} < +0,4$ .

Para ilustrar, de uma forma simples, o processo de aprendizagem desta rede de autómatos, considere-se que uma sequência de 1666 símbolos codificados com apenas três códigos binários ( $n=3$ ). Com esta codificação a rede de autómatos será inicialmente constituída por um total de 14 autómatos, conforme se apresenta na Figura 4.6. Na anterior sequência de símbolos os primeiros 333 encontram-se totalmente corrompidos por ruído e, nos restantes somente 25% destes se encontram corrompidos (dos 75% símbolos não sujeitos a ruído, 50% representam um símbolo codificado por '+1 -1 -1' e 25% outro símbolo codificado por '+1 -1 +1').



**Figura 4.6 – Rede de Autômatos para uma codificação binária de três bits ( $n=3$ ).**

Na Figura 4.7, onde o eixo das abcissas está graduado em códigos binários, pode observar-se a evolução dos pesos sinápticos  $w_{ij}$  ao longo do processo de aprendizagem. A influência do factor tempo é determinante, pois os pesos sinápticos de uma camada  $i+1$  só começam a evoluir após os correspondentes da camada  $i$  apresentarem sinais de efectiva aprendizagem. No final do processo de aprendizagem somente os pesos  $w_{11}$  na primeira camada,  $w_{22}$  na segunda camada e  $w_{33}$  e  $w_{34}$  na última camada evoluíram para  $\pm 1/2$ . Todas as restantes conexões sinápticas são eliminadas por um processo de *prunning*, resultando na rede apresentada na Figura 4.8.

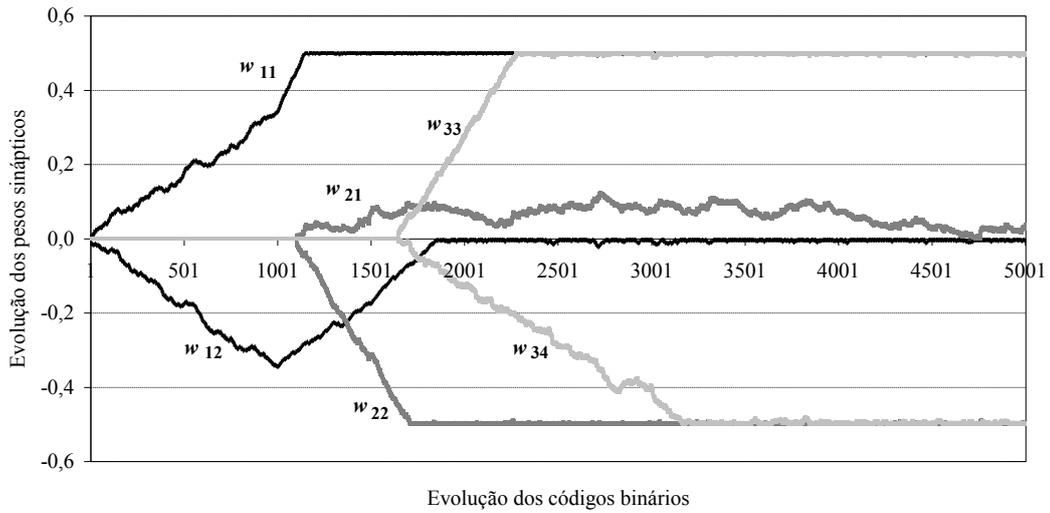


Figura 4.7 – Evolução da aprendizagem para a rede de autômatos representada na Figura 4.6.

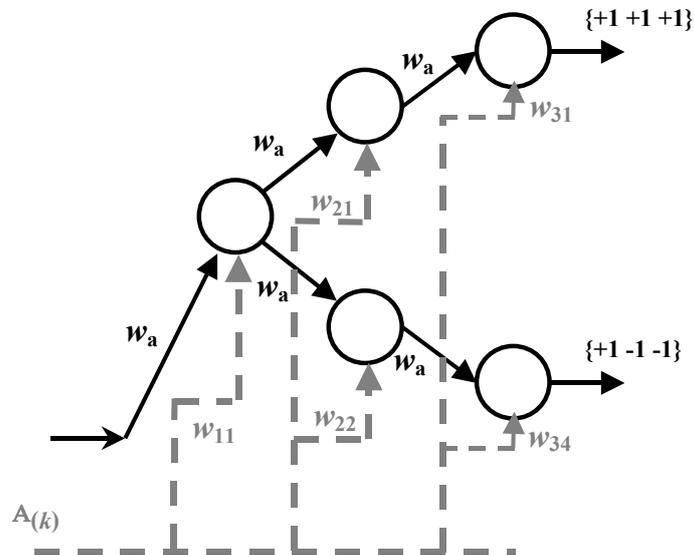


Figura 4.8 – Rede da Figura 4.6 após decorrer o processo de aprendizagem e de *prunning*.

Após o processo de aprendizagem, e considerando ambas as saídas do sistema conexionista apresentado na Figura 4.3, é possível assumir esta rede de autómatos como um detector de símbolos. A rede da Figura 4.8 pode ser utilizada como um detector dos dois símbolos codificados por  $\{+1 +1 +1\}$  e  $\{+1 -1 -1\}$ . Estas redes – detectores de símbolos – serão utilizadas como unidades básicas do sistema conexionista hierarquicamente superior e cujo o objectivo é descrever as produções que estabelecem as relações entre os vários símbolos da linguagem gerada pelo sistema dinâmico comandado.

## 4.4 SEGUNDO NÍVEL DO SISTEMA CONEXIONISTA – APRENDIZAGEM DAS PRODUÇÕES

Mantendo a filosofia subjacente à estrutura conexionista adoptada, este segundo nível hierárquico será baseado numa estrutura semelhante à do primeiro nível. Adicionalmente deverá, além de implementar as produções do sistema de rescrita da gramática representativa da linguagem gerada pelo sistema dinâmico, possibilitar a aprendizagem automática dessas mesmas produções.

A rede de autómatos desenvolvida para o primeiro nível hierárquico será aqui utilizada como estrutura de identificação de símbolos (terminais e não terminais), que designaremos por ‘autómatos de identificação’. Um ‘autômato de identificação’ é assim uma unidade de processamento com dois estados, que gerará uma saída unitária somente quando o símbolo de entrada for correctamente identificado. Será considerado um ‘autômato de identificação’ distinto para cada símbolo dos alfabetos utilizados pela gramática. Em ( 4.7 ) especifica-se um ‘autômato de identificação’ para um símbolo do alfabeto terminal, sendo que para o alfabeto não terminal a especificação será em tudo semelhante.

$$\begin{aligned} A(t) &\in \Sigma_T \\ B(t) &\in \{0,1\} \\ \Theta(t+1) &= \begin{cases} 1 & , \text{ se o símbolo foi bem identificado} \\ 0 & , \text{ se o símbolo não foi bem identificado} \end{cases} \\ B(t) &= \Theta(t) \end{aligned} \quad (4.7)$$

Como já estudado em capítulos anteriores, as produções do sistema de rescrita podem ser de vários tipos especificando o comportamento dinâmico de um qualquer sistema em distintas zonas do seu domínio de funcionamento. Existem descritas na literatura (Kain, 1972) aplicações com redes de autómatos para identificar sequências de símbolos. A rede apresentada na Figura 4.9 identifica a sequência ‘ab’, sendo que o segundo autômato só gerará uma saída positiva caso o primeiro o tenha feito anteriormente. Mais uma vez a noção de evolução temporal é aqui fundamental.



**Figura 4.9 – Rede de autômatos para identificar a sequência ‘ab’.**

Partindo desta utilização, construir-se-á uma rede de ‘autômatos de identificação’ que represente as várias produções presentes na gramática inferida a partir do sistema dinâmico. Esta rede será constituída por  $m+2$  camadas, onde  $m$  denota a máxima ordem permitida ( $p_{max}$ ) ou atingida para as produções. Uma gramática que apresenta, no máximo, *produções de tipo 2* terá assim uma rede com 4 camadas. Três destas camadas representarão as várias produções (tipo0, tipo1 e tipo2) e a quarta camada funciona como uma camada de saída. A rede recebe como entradas as sequências de símbolos terminais e não terminais, relativos à evolução do sistema dinâmico, e a sua saída será, em cada unidade de tempo, um símbolo terminal que será coerente com a evolução da palavra terminal proveniente do sistema dinâmico.

Como exemplo considere-se  $abA \rightarrow abb\delta$  como uma das produções da gramática em causa. Esta *produção de tipo 2* será representada pela rede apresentada na Figura 4.10. Os três primeiros nós serão ‘autômatos de identificação’ e o último um autômato de saída. O primeiro ‘autômato de identificação’ reconhecerá o símbolo não terminal ‘A’ e os dois seguintes os símbolos terminais ‘b’ e ‘a’. Estes serão responsáveis pela identificação correcta da sequência de símbolos terminais e pelo símbolo não terminal que constituem o antecedente da produção. O autômato de saída gerará o símbolo, presente no conseqüente da produção, que determina a correcta construção da palavra. A rede é do tipo *feedforward*, com todos os pesos sinápticos, interligando os vários nodos, unitários. A palavra terminal é entrada de forma sequencial, nas várias camadas, em função da sua evolução temporal.

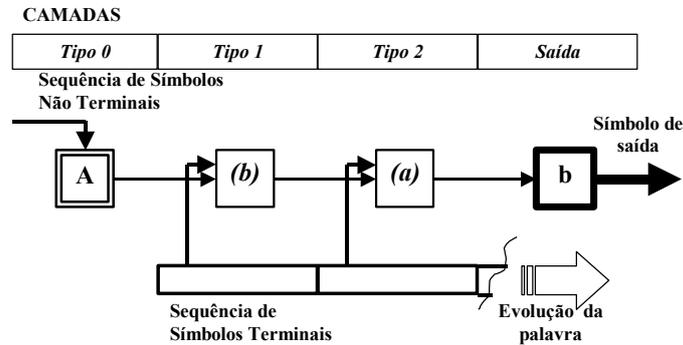


Figura 4.10 – Rede de autómatos para representar a produção  $abA \rightarrow abb\delta$ .

Como já estudado anteriormente na secção 3.5.2 a existência de uma *produção de tipo p* válida elimina um número de *produções de tipo p+1* dado por ( 3.26 ), o que vai diminuir o número de ligações sinápticas possíveis entre camadas. Se, no exemplo anterior, fosse válida a produção  $A \rightarrow b\delta$ , tal implicaria a existência de uma única ligação sináptica entre a primeira camada e a camada de saída.

$$\sum_{i=1}^{p_{\max}-p} (\text{num}(\Sigma_T))^i \quad ( 3.26 )$$

Por forma a melhor explicitar o funcionamento da rede de autómatos vamos considerar uma gramática com o conjunto de produções dado por ( 4.8 ).

$$P = \{A \rightarrow a\delta, aB \rightarrow ab\delta, cB \rightarrow ca\delta, abB \rightarrow abb\delta, bbB \rightarrow bbc\delta\} \quad ( 4.8 )$$

A rede de autómatos que representa o anterior conjunto encontra-se representada na Figura 4.11.

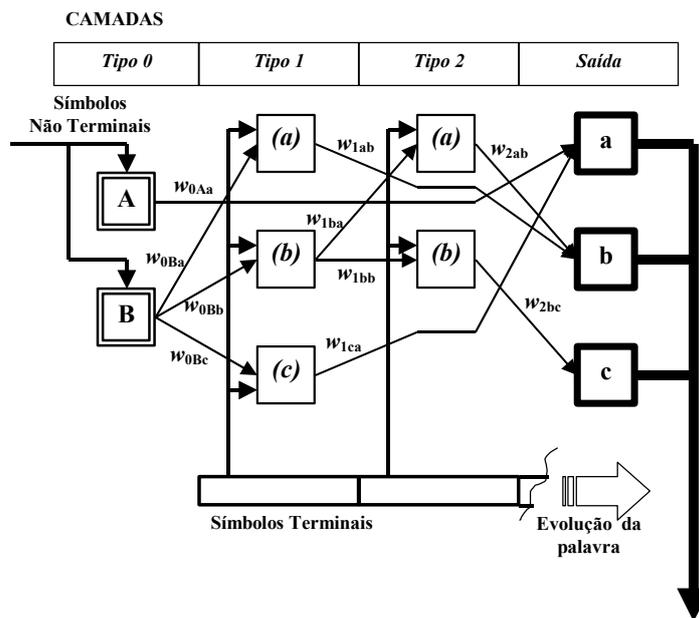


Figura 4.11 – Rede de autômatos para o conjunto de produções  $P$ .

Em cada unidade de tempo apenas um dos autômatos da camada de saída se encontra activo. Deste modo a sequência de símbolos gerados na saída da rede podem ser comparados à evolução da palavra gerada pelo sistema dinâmico, funcionando assim a rede de autômatos como um modelo do sistema dinâmico.

Será de todo o interesse que a anterior rede seja construída de uma forma automática, a partir apenas da análise da informação de entrada/saída referente ao sistema dinâmico. A rede será assim construída por um processo de inferência semelhante ao processo de inferência gramatical utilizado na obtenção das produções.

#### 4.4.1 Inferência gramatical da rede das produções

O objectivo é estabelecer a rede de produções a partir exclusivamente das palavras provenientes do sistema dinâmico. A única informação disponível, à partida, são os alfabetos (terminal e não terminal) e a ordem máxima desejada para as produções. Esta situação é semelhante aquela que foi considerada aquando do estabelecimento das produções por um processo de inferência gramatical.

O processo de estabelecimento da rede das produções baseia-se em dois conceitos fundamentais. Primeiro, o conceito de *reinforcement learning* segundo o qual as ligações sinápticas evoluirão para a unidade sempre que a palavra presente à rede potenciar a produção traduzida por essas mesmas ligações. Segundo, o facto de as produções do tipo  $p+1$  só serem aprendidas quando já não existe possibilidade de aprender produções do tipo  $p$ .

Para exemplificar o estabelecimento da rede considera-se um exemplo simples, onde a gramática é constituída pelos alfabetos  $\Sigma_N=\{A,B\}$  e  $\Sigma_T=\{a,b,c\}$  e pelo conjunto de produções apresentado anteriormente em ( 4.8 ).

Como informação disponível, para o processo de inferência, existe uma sequência de símbolos (organizados em palavras) terminais e não terminais ( 4.9 ). Estes símbolos podem sempre ser obtidos a partir da codificação da evolução das variáveis de comando e saída de um qualquer sistema dinâmico. Adicionalmente considerar-se-á  $p_{max}=2$  como o tipo máximo das produções.

$$\left\{ \begin{array}{l} B B B B B B B B B B B B A A A A B B B B B B B B A A \dots \\ a b b c a b b c a b b c a a a a b b c a b b c a a a \dots \end{array} \right\} \quad (4.9)$$

Em primeiro lugar são estabelecidas as ligações sinápticas da primeira camada, correspondentes às produções do tipo 0.

Inicialmente estão disponíveis para aprendizagem todas as ligações sinápticas entre esta primeira camada e a camada de saída, tal como apresentado na Figura 4.12, o que corresponde a considerar como potencialmente válidas todas as seis possíveis produções do tipo 0 ( 4.10 ). Todos os pesos sinápticos são inicializados aleatoriamente evoluindo segundo uma regra de *reinforcement learning* ( 4.11 ) com uma taxa de aprendizagem  $\gamma=0,01$ , onde a evolução para a unidade é função da produção correspondente existir ou não.

$$\{A \rightarrow a\delta, A \rightarrow b\delta, A \rightarrow c\delta, B \rightarrow a\delta, B \rightarrow b\delta, B \rightarrow c\delta\} \quad (4.10)$$

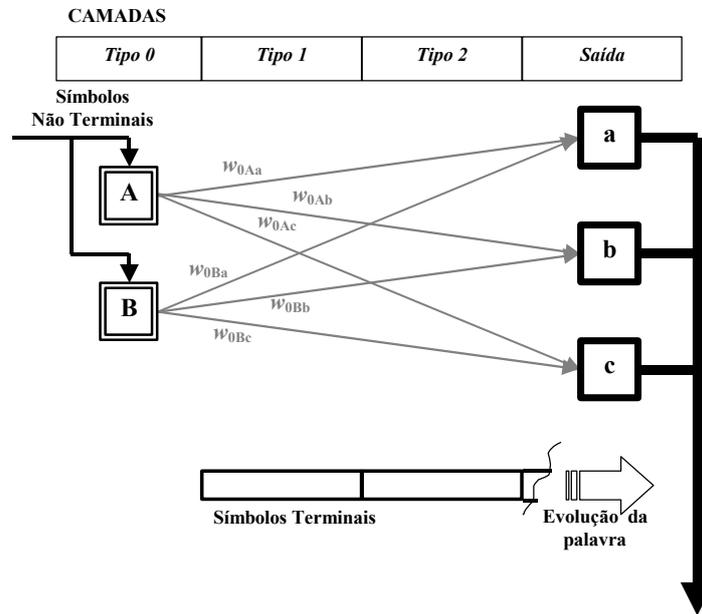


Figura 4.12 – Aprendizagem das produções de tipo 0.

$$\begin{cases} w_{pij}(t+1) = \gamma + (1 - \gamma)w_{pij}(t) & \text{se símbolo terminal} = \text{autômato } j \\ w_{pij}(t+1) = (1 - \gamma)w_{pij}(t) & \text{se símbolo terminal} \neq \text{autômato } j \end{cases} \quad (4.11)$$

Durante o processo de aprendizagem somente um dos pesos –  $w_{0Aa}$  – evoluiu para a unidade, conforme se pode verificar na Figura 4.13, indicando a existência de apenas uma produção do tipo 0 válida, a produção  $A \rightarrow a\delta$ .

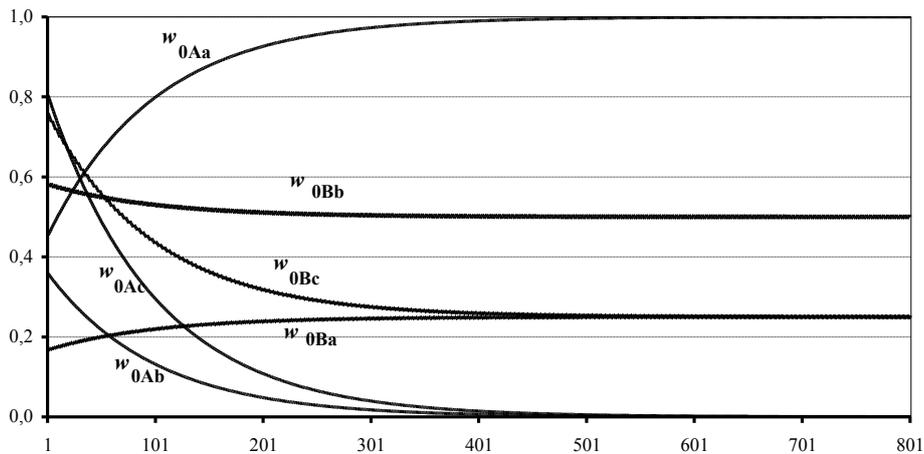


Figura 4.13 – Evolução dos pesos sinápticos na aprendizagem das produções de tipo 0.

Deste modo, e após o processo de aprendizagem das produções do tipo 0, a rede fica com a topologia apresentada na Figura 4.14.

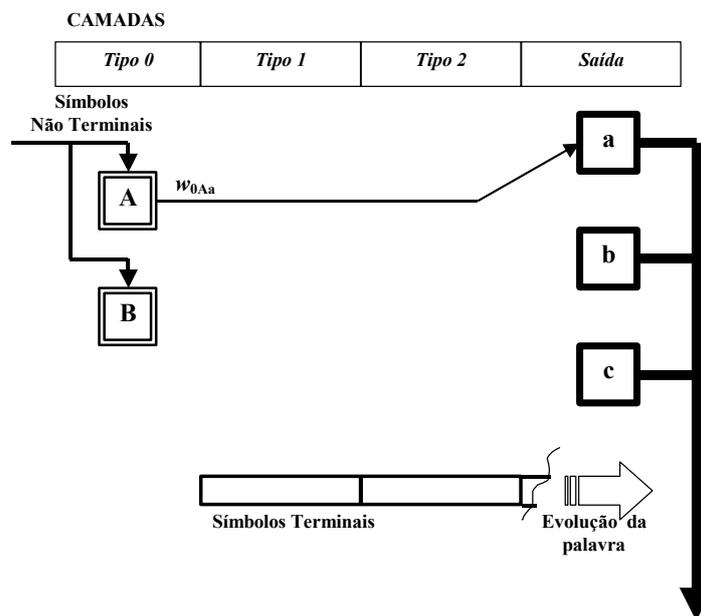


Figura 4.14 – Produções de tipo 0.

De seguida são estabelecidas as ligações sinápticas da segunda camada, correspondentes às produções do tipo 1. Uma vez que já foi estabelecida uma produção do tipo 0 encontram-se assim indisponíveis nove das dezoito possíveis produções do tipo 1. Na

Figura 4.15 apresentam-se, em traço contínuo cinzento, as ligações sinápticas referentes às produções do tipo 1 com possibilidade de existência no sistema de rescrita.

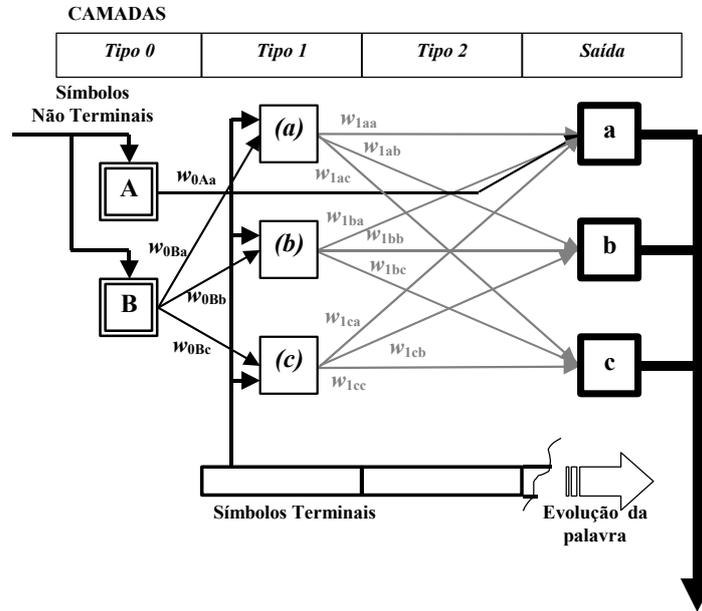


Figura 4.15 – Aprendizagem das produções de tipo 1.

Por aplicação da anterior regra de aprendizagem ( 4.11 ) verifica-se, na Figura 4.16, a evolução de dois pesos sinápticos para a unidade –  $w_{1ca}$  e  $w_{1ab}$  – que correspondem às produções do tipo 1  $aB \rightarrow abd$  e  $cB \rightarrow ca\delta$ .

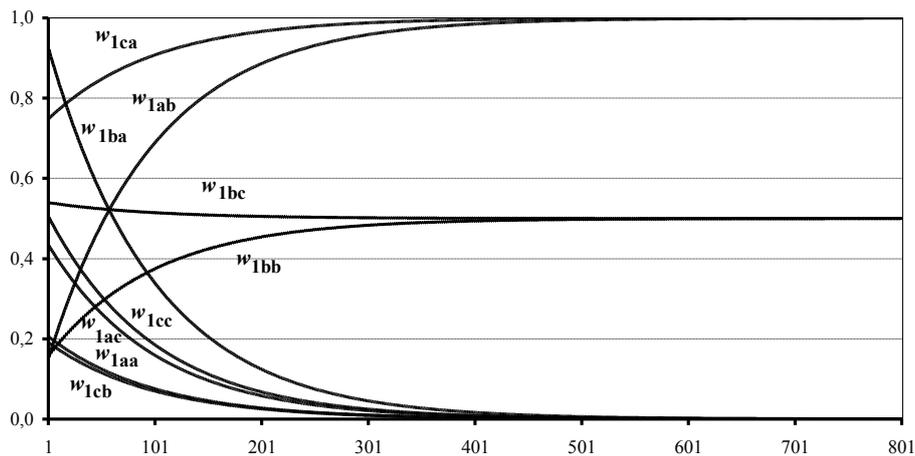


Figura 4.16 – Evolução dos pesos na aprendizagem das produções de tipo 1.

Após esta etapa do processo de aprendizagem a rede evolui da topologia apresentada na Figura 4.14 para a apresentada na Figura 4.17, onde já se encontram estabelecidas as produções do tipo 0 e as produções do tipo 1.

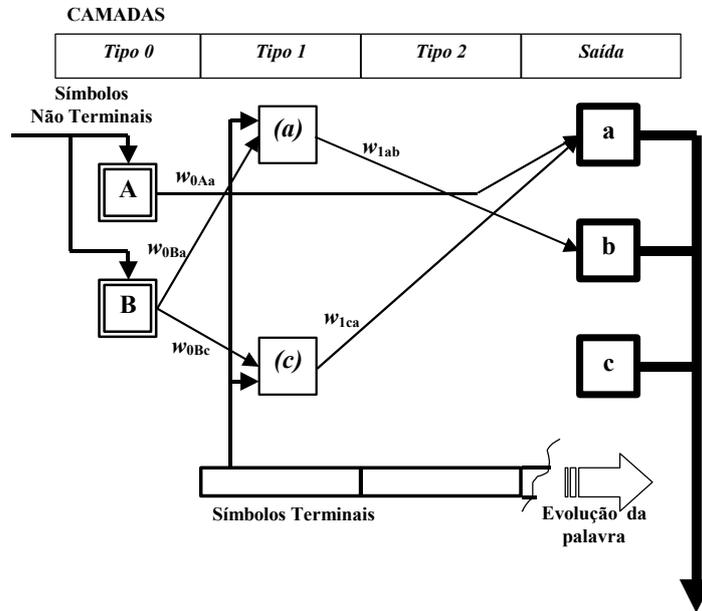


Figura 4.17 – Produções de tipo 0 e 1.

Por fim, e atendendo ao facto de se ter considerado  $p_{\max}=2$  como o tipo de produção máximo, estabelecer-se-ão as ligações sinápticas da terceira e última camada de ‘autómatos de identificação’, correspondente às produções do tipo 2. Pelo facto de já existirem produções válidas de tipo inferior, apenas estão disponíveis nove das vinte e sete produções do tipo 2, representadas a traço contínuo cinzento na Figura 4.18.

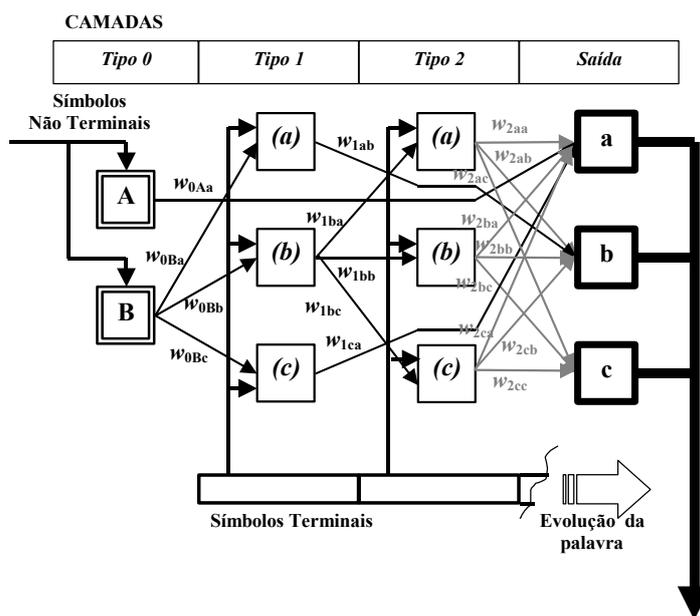


Figura 4.18 – Aprendizagem das produções de tipo 2.

Por aplicação da regra de aprendizagem ( 4.11 ) verifica-se, na Figura 4.19, a evolução de dois pesos sinápticos para a unidade –  $w_{2ab}$  e  $w_{2bc}$  – que correspondem às produções do tipo 2  $abB \rightarrow abb\delta$  e  $bbB \rightarrow bbc\delta$ .

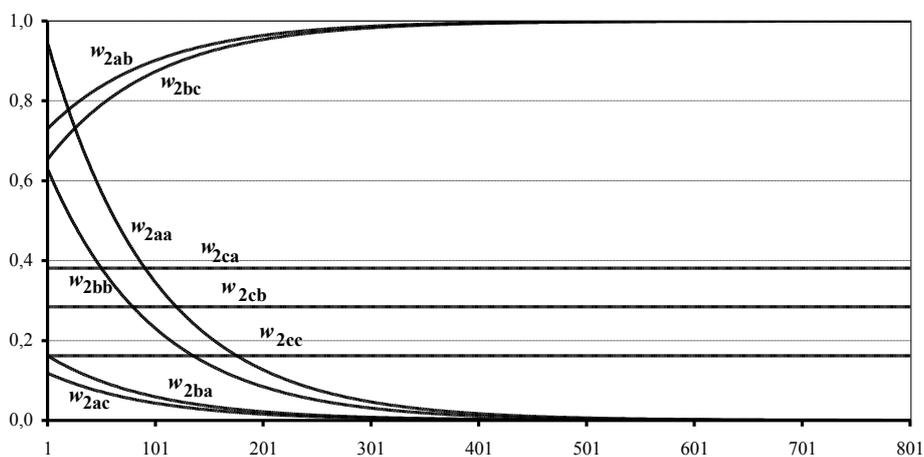


Figura 4.19 – Evolução dos pesos na aprendizagem das produções de tipo 2.

Finda esta última etapa do processo de aprendizagem a rede evolui da Figura 4.18 para a Figura 4.20, necessariamente igual à Figura 4.11.

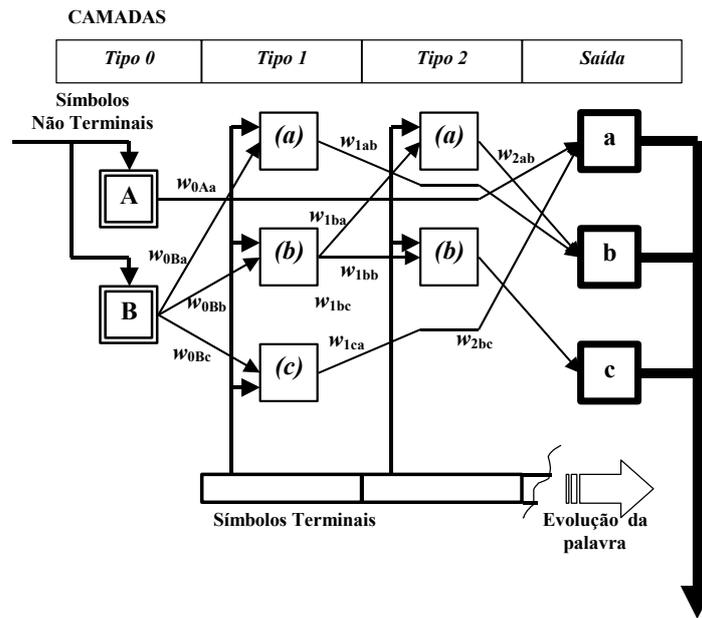


Figura 4.20 – Produções de tipo 0, 1 e 2.

Este exemplo ilustra o processo de inferência de uma rede de autómatos que representa o conjunto de produções do sistema de rescrita de uma gramática. Como informação inicial apenas se torna necessário conhecer os alfabetos da gramática – por forma a utilizar os ‘autómatos de identificação’ respectivos – e o tipo de produção máximo a considerar. Durante o processo de aprendizagem as ligações sinápticas são estabelecidas – por utilização de uma regra de *reinforcement learning* – à medida que a sequência de símbolos terminais e não terminais – provenientes da codificação das variáveis de comando e saída do sistema dinâmico – é analisada.



## 4.5 CONCLUSÕES

Neste capítulo desenvolveu-se uma forma conexionista de representação da gramática inferida a partir da análise da linguagem gerada por um sistema dinâmico comandado.

A estrutura baseia-se no estabelecimento de redes de autómatos, com todas as vantagens de optimização combinatória e de dependência temporal que advém da sua utilização. Esta estrutura está organizada em dois níveis hierárquicos, sendo um responsável pelos símbolos (alfabetos) e outro pelas produções.

Ambos os níveis são estabelecidos por um processo de aprendizagem através de exemplos. O primeiro nível identifica os símbolos, presentes nos alfabetos terminal e não terminal, provenientes da evolução quer da dinâmica do sistema quer do seu comando. O segundo nível identifica as produções da gramática em função da referida evolução.

O processo de aprendizagem desenvolve-se assim em duas fases, uma primeira fase para os símbolos e uma segunda para as produções. O facto de os algoritmos de aprendizagem recorrerem ao conceito de *reinforcement learning* providencia boas características de robustez, quer na identificação de símbolos quer na extracção de produções.

Finda a aprendizagem a estrutura conexionista resultante representa a gramática que descreve a linguagem gerada pelo sistema dinâmico, podendo ser utilizada como um modelo do mesmo.



## CAPÍTULO 5 - APLICAÇÕES

**Resumo:** Neste capítulo é apresentada uma verificação experimental da inferência gramatical como algoritmo de aprendizagem através de exemplos. A gramática é inferida com o objectivo de modelizar sistemas de accionamento electromecânico através de uma relação funcional representativa do funcionamento dos mesmos. São verificadas propriedades e aspectos particulares necessários a um bom desempenho. Além dos aspectos de modelização é também analisada a aplicação dos algoritmos gramaticais no comportamento do accionamento. Designadamente são aplicados no controlo, gerando o comando por forma a que a saída do accionamento siga uma determinada referência, e na detecção de anomalias.



## 5.1 INTRODUÇÃO

Neste capítulo efectua-se uma verificação experimental do algoritmo de inferência gramatical quando aplicado à modelização, detecção de anomalias e controlo de sistemas de accionamento electromecânico.

O sistema de accionamento electromecânico em análise é constituído por uma máquina assíncrona alimentada por um conversor electrónico de potência. A complexidade resultante do elevado número de variáveis envolvidas torna a modelização destes sistemas uma tarefa complexa. Por forma a simplificar o processo de modelização é usual reduzir este número, fazendo uso do conhecimento físico do sistema e de algumas técnicas de controlo complementares. A simplificação introduzida permite a aplicação das técnicas de inferência gramatical sobre um menor número de variáveis, com os consequentes ganhos ao nível do processamento sem invalidar as conclusões alcançadas.

Após a caracterização e análise do sistema de accionamento electromecânico considerado, o processo de inferência gramatical será aplicado quando se pretende modelizar de uma forma automática a sua velocidade, extraindo uma relação funcional entre as suas variáveis.

Neste contexto, características como a especificação da gramática, o preenchimento de lacunas gramaticais através do processo de interpolação gramatical, e a capacidade de generalização serão verificadas para o caso concreto do sistema de accionamento electromecânico.

Sendo a gramática inferida a partir dos dados experimentais do sistema, apresenta-se como uma descrição do comportamento mesmo. Sempre que este comportamento se afaste do normal, por exemplo em presença de uma anomalia, as palavras codificadas a partir da sua evolução dinâmica deixam de ser representadas pela gramática anteriormente inferida. Deste modo, é possível considerar a utilização desta mesma gramática para detectar este facto, bem como o afastamento da nova linguagem em

relação à original. Será considerada uma simples aplicação para ilustrar a utilização da análise gramatical na detecção de anomalias em sistemas dinâmicos.

Os bons desempenhos na obtenção, de uma forma automática, de uma relação funcional representativa do comportamento de um sistema dinâmico comandado potenciam a utilização desta informação em estratégias de controlo de sistemas de accionamento electromecânico. O objectivo será o estabelecer a variável de comando por forma a que a variável de saída do accionamento siga uma dada referência.

Os sistemas de controlo ‘com aprendizagem’ potenciam uma melhoria no seu desempenho com base no conhecimento que vão inferindo do meio com o qual interactuam. A utilização das capacidades de memória, adaptação e generalização dos algoritmos de aprendizagem através de exemplos permitem a adaptação contínua da relação funcional inferida a partir da análise dos vários dados experimentais permanentemente disponíveis. Por outras palavras, será importante manter a gramática representativa da linguagem gerada pelo sistema de accionamento electromecânico o mais actualizada possível. O algoritmo de aprendizagem deverá ser capaz de efectuar as necessárias adaptações na gramática em face de alterações de parâmetros ou do próprio domínio de funcionamento do sistema.

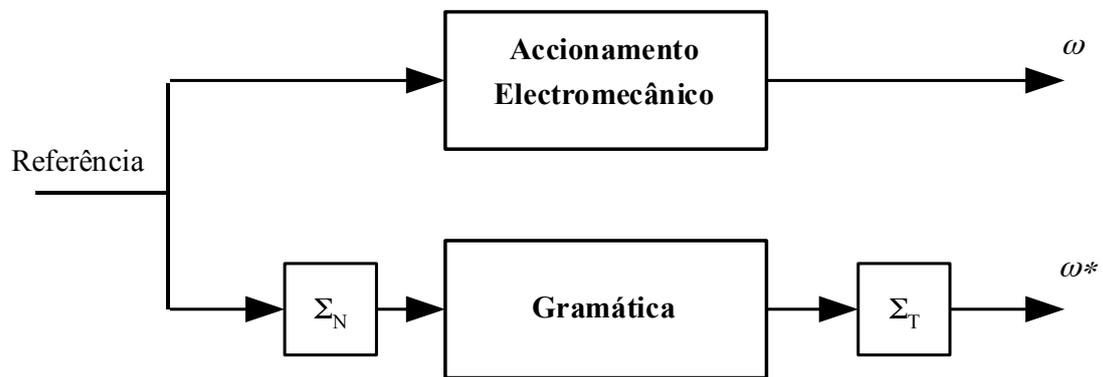
## 5.2 MODELIZAÇÃO

O conceito de modelização, utilizado na perspectiva dos algoritmos de aprendizagem através de exemplos, não se apresenta em tudo idêntico ao conceito de identificação de sistemas, tão utilizado na análise e controlo de sistemas dinâmicos.

Segundo alguns autores (Sage e Melsa, 1971) a identificação de sistemas consiste no processo de determinação de equações diferenciais ou às diferenças, ou dos parâmetros a estas associados, que descrevem um fenómeno físico de acordo com um critério pré-determinado. Deste modo, no problema da identificação de sistemas a estrutura e ordem do modelo estão estabelecidas *à priori*. Em sistemas de accionamento electromecânico, técnicas associadas à identificação de sistemas como o filtro de Kalman (Dhaouadi, Bolognami e Zigliotto, 1992) ou algoritmos baseados em observadores (Bodson, Chaisson e Novotnak, 1995) têm sido utilizados com algum sucesso, quando integrados em sistemas de controlo.

No entanto, o conceito de modelização considerado neste trabalho está mais próximo dos sistemas de classificação ou de reconhecimento de padrões, onde grupos de padrões são classificados em categorias (Kanal, 1974). Assim, entende-se por modelização a identificação de relações funcionais entre as condições do comando do sistema e o estado e saída observados.

Para verificar experimentalmente o desempenho da modelização gramatical, efectua-se a comparação entre a velocidade do accionamento electromecânico (velocidade angular do rotor da máquina eléctrica –  $\omega$ ) e a obtida a partir da utilização da gramática –  $\omega^*$ , tal como se apresenta na Figura 5.1. Este processo de modelização compreende duas fases: fase de treino e fase de teste.



**Figura 5.1. Verificação experimental da modelização gramatical.**

A fase de treino caracteriza-se pela recolha de informação acerca do sistema a modelizar. Para obter conjuntos de treino relevantes, dois tipos de informação devem ser considerados, qualitativa e quantitativa. A informação qualitativa é obtida essencialmente a partir da análise dos modelos matemáticos do accionamento. Isto leva à escolha das variáveis relevantes que descrevem o comportamento do sistema. Esta escolha é muito importante para assegurar a existência de uma relação funcional representativa do comportamento dinâmico do accionamento (Martins, Pires e Dente, 1998). Caso contrário, quer a aprendizagem quer o reconhecimento revelam-se muito difíceis. A informação quantitativa é obtida a partir de um conjunto de dados experimentais do sistema. De modo a obter estes dados torna-se necessário escolher um sinal de excitação que deve ser aplicado ao comando do sistema. Uma possível aproximação seria a utilização de um sinal binário pseudo-randómico (Low, Lee e Lim, 1993). No entanto, este tipo de sinal não se apresenta como a melhor escolha para sistemas de accionamento electromecânico, pois é filtrado pelas constantes de tempo mecânicas do sistema. Deste modo, é preferível utilizar sinais sinusoidais de diferentes frequências e amplitudes, ajustadas dentro dos limites de resposta do accionamento, evitando o problema da filtragem e recolhendo uma quantidade de informação que se estende de uma forma adequada a todo o domínio de funcionamento. O conjunto de dados obtido, designado por ‘conjunto de treino’, será utilizado pelo algoritmo de aprendizagem para inferir a gramática que descreve a linguagem associada ao sistema.

A fase de teste caracteriza-se por considerar um conjunto de situações não verificadas aquando da fase de treino. O novo conjunto de dados, designado por ‘conjunto de teste’ e necessariamente disjunto do conjunto de treino, servirá para testar as capacidades de representação da gramática anteriormente inferida.

## 5.2.1 Caracterização do accionamento electromecânico

O estudo de um sistema de accionamento electromecânico, como o que se apresenta na Figura 5.2, compreende não apenas a máquina eléctrica mas também o conversor electrónico de potência; havendo ainda a considerar todos os sensores eléctricos e mecânicos, bem como os processadores digitais indispensáveis ao comando e controlo do accionamento. É importante considerar as características de todos estes elementos, e as suas inter-relações, quando se pretende efectuar uma modelização do accionamento electromecânico.

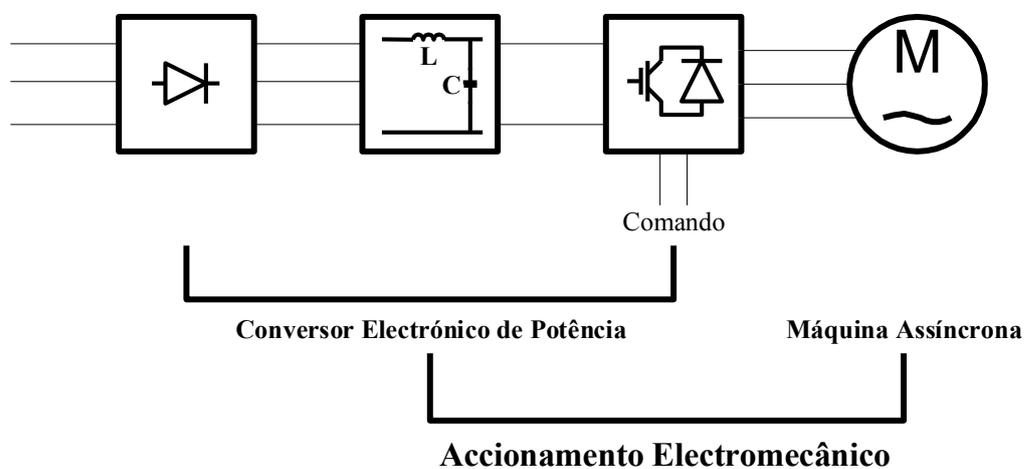


Figura 5.2. Esquema de princípio do accionamento electromecânico.

### 5.2.1.1 Máquina assíncrona

O accionamento electromecânico em estudo baseia-se na utilização de uma máquina assíncrona. Um sistema trifásico de tensões aplicado aos terminais desta máquina origina um campo girante no entreferro, cuja velocidade angular depende do número de pólos da máquina e da frequência estatórica.

Sempre que o rotor rodar a uma velocidade diferente da velocidade do campo girante criado pelas correntes estatóricas (velocidade de sincronismo) é induzida uma força electromotriz nos circuitos do rotor, responsável pela circulação de corrente nestes circuitos. Da interação entre estas correntes rotóricas e o campo girante resulta um binário electromagnético responsável pelo movimento de rotação da máquina. A

situação de sincronismo é caracterizada pela ausência de binário, pois os enrolamentos rotóricos rodam à mesma velocidade do campo girante estático impedindo a indução de força electromotriz nos referidos enrolamentos (Fitzgerald, Kingsley e Umans, 1985). O binário produzido é função dos parâmetros da máquina e das correntes que percorrem os enrolamentos do estator e do rotor.

A modelização deste tipo de máquina assíncrona encontra-se sujeita a dificuldades de vária ordem. Além das dificuldades em obter algumas das variáveis presentes, existem outras como sejam a presença de não linearidades, a dificuldade em conhecer o valor exacto de alguns parâmetros, a variação a que estes se encontram sujeitos em função do regime de funcionamento, a existência de perdas por correntes de Foucault ou por histerese, a variação da resistência dos condutores por efeito pelicular e ainda a modelização de fenómenos térmicos.

A máquina assíncrona apresenta-se assim como um sistema físico multivariável, não linear, de dinâmica relativamente complexa. A modelização mais frequente é baseada na teoria de circuitos, considerando os vários circuitos como filiformes. Esta modelização apresenta, à partida, algumas limitações pois não considera certos fenómenos térmicos, como a dependência de alguns parâmetros da temperatura, ou eléctricos como a presença de efeito pelicular. As relações estabelecidas entre as várias variáveis (estado, comando e saída) resultam no modelo da máquina.

Um circuito indutivo trifásico equilibrado com acoplamento magnético entre fases é regido pelo sistema de equações diferenciais ( 5.1 ). Nesta equação matricial  $\mathbf{u}_{abc}$ ,  $\mathbf{i}_{abc}$  e  $\boldsymbol{\psi}_{abc}$  representam, respectivamente, os vectores das tensões, correntes e fluxos ligados com os vários enrolamentos estáticos e rotóricos.  $\mathbf{R}_{abc}$  é uma matriz quadrada diagonal com as resistências, por fase, dos enrolamentos estáticos e rotóricos. Considerando um sistema trifásico de tensões e um conjunto de três enrolamentos no estator e no rotor da máquina assíncrona, a matriz e os vectores existentes na equação ( 5.1 ) apresentam dimensão 6.

$$\mathbf{u}_{abc} = \mathbf{R}_{abc} \mathbf{i}_{abc} + \frac{d\boldsymbol{\psi}_{abc}}{dt} \quad ( 5.1 )$$

A equação mecânica ( 5.2 ) traduz o equilíbrio dos binários presentes, onde  $J$  representa o momento de inércia. O termo  $T_{em}$  designa o binário electromagnético desenvolvido

pela máquina;  $B\omega$  o binário de atrito proporcional à velocidade angular do rotor  $\omega$ , com constante de proporcionalidade  $B$ ; e  $T_{\text{ext}}$  o conjunto dos binários exteriores, onde é possível englobar todos os binários considerados parasitas como o de ventilação, o de vibração e os criados por harmónicas superiores.

$$J \frac{d\omega}{dt} = T_{\text{em}} - B\omega - T_{\text{ext}} \quad (5.2)$$

Os fluxos magnéticos ligados ( 5.3 ) são uma função não linear das correntes que percorrem os vários enrolamentos do estator e do rotor –  $\mathbf{i}_{\text{abc}}$  – e da posição do rotor –  $\theta$ .

$$\Psi_{\text{abc}} = \psi(\mathbf{i}_{\text{abc}}, \theta) \quad (5.3)$$

O binário electromagnético ( 5.4 ), entendido como a derivada parcial da co-energia magnética –  $W_c$  – em ordem à posição do rotor, depende igualmente das correntes e da posição do rotor.

$$T_{\text{em}} = \frac{\partial W_c}{\partial \theta} = \frac{\partial}{\partial \theta} \sum_j \int_0^j \psi(\sigma_j, \theta) d\sigma_j \quad (5.4)$$

O modelo matemático resultante das expressões anteriores apresenta-se extremamente complexo, difícil de manipular e utilizar em aplicações onde se exige o funcionamento em tempo real. Sobre este modelo são habitualmente consideradas algumas hipóteses simplificativas, para que o problema da modelização seja simplificado.

Não considerando a saturação magnética do ferro, o fluxo magnético apresenta uma dependência linear das correntes ( 5.5 ), onde  $\mathbf{L}_{\text{abc}}(\theta)$  representa a matriz dos coeficientes de indução. Esta hipótese vai introduzir simplificações tanto na equação eléctrica ( 5.1 ) como na expressão do binário electromagnético ( 5.4 ), que passam a assumir a forma indicada respectivamente por ( 5.6 ) e ( 5.7 ).

$$\Psi_{\text{abc}} = \mathbf{L}_{\text{abc}}(\theta) \mathbf{i}_{\text{abc}} \quad (5.5)$$

$$\mathbf{u}_{\text{abc}} = \mathbf{R}_{\text{abc}} \mathbf{i}_{\text{abc}} + \mathbf{L}_{\text{abc}}(\theta) \frac{d\mathbf{i}_{\text{abc}}}{dt} + \omega \frac{\partial \mathbf{L}_{\text{abc}}(\theta)}{\partial \theta} \mathbf{i}_{\text{abc}} \quad (5.6)$$

$$T_{em} = \frac{1}{2} \sum_{k,j} \frac{\partial L_{abc}(\theta)[j,k]}{\partial \theta} i_j i_k \quad (5.7)$$

Admitindo circuitos com espaçamento angular de  $120^\circ$ , estrutura sinusoidal por fase, simetria magnética, enrolamentos estatóricos iguais, entre-ferro uniforme, a não ligação do condutor de neutro e desprezando a existência de harmónicas diferentes da fundamental, a matriz dos coeficientes de indução é passível de simplificação. Mais importante do que isso é a possibilidade de efectuar transformações de variáveis que levam à simplificação do modelo. É assim possível obter um modelo matemático para a dinâmica eléctrica ( 5.8 ), representado num referencial genérico dq a rodar a uma velocidade genérica  $\omega_R$ , e para a expressão do binário electromagnético da máquina ( 5.9 ), onde todos os coeficientes se apresentam constantes.

$$\mathbf{u}_{dq} = \mathbf{R}_{dq} \mathbf{i}_{dq} + \mathbf{L}_{dq} \frac{d\mathbf{i}_{dq}}{dt} + \omega \mathbf{G}_{\theta dq} \mathbf{i}_{dq} + \omega_R \mathbf{G}_{\gamma dq} \mathbf{i}_{dq} \quad (5.8)$$

$$T_{em} = \sum_{k,j} \mathbf{G}_{\theta dq}[j,k] i_j i_k \quad (5.9)$$

O modelo resultante das hipóteses e aproximações referidas, onde se consideram como variáveis de estado eléctricas as correntes estatóricas e rotóricas e como variável de estado mecânica a velocidade, é bastante aplicado no estudo de accionamentos electromecânicos.

Relativamente ao modelo resultante das equações ( 5.8 ) e ( 5.9 ) é importante salientar, além das não linearidades presentes na equação eléctrica, a não medição das correntes que circulam no rotor e as incertezas a que os parâmetros que definem as relações dos fluxos de ligação podem estar sujeitos ( 5.5 ). Por outro lado tanto as resistências dos enrolamentos do estator como do rotor encontram-se sujeitas a variações ao longo do tempo. Na equação mecânica é difícil conhecer com exactidão os seus parâmetros, nomeadamente o momento de inércia e o coeficiente de atrito. Além da não linearidade do binário electromagnético, também o próprio binário exterior se apresenta muitas vezes como não linear.

Na implementação de técnicas de controlo vectorial, como o controlo por orientação de campo (Palma, 1993) ou a metodologia de controlo apresentada em (Pires e Dente, 1991a) e (Pires, e Dente, 1991b), onde se torna importante o conhecimento dos fluxos

magnéticos ligados, interessa considerar um modelo dinâmico onde estas variáveis se apresentem como variáveis de estado. Assim, indica-se em ( 5.10 ) uma variante do modelo de estado anterior, também bastante utilizada, onde se consideram como variáveis de estado eléctricas as componentes das correntes que percorrem os enrolamentos do estator ( $i_{ds}$  e  $i_{qs}$ ) e as componentes do fluxo magnético rotórico ( $\psi_{dr}$  e  $\psi_{qr}$ ), expressas num referencial dq genérico a rodar a uma velocidade  $\omega_R$ , e como variável de estado mecânica a velocidade angular do veio ( $\omega$ ). Como variáveis de comando consideram-se as componentes das tensões aplicadas ao estator ( $u_{ds}$  e  $u_{qs}$ ), também representadas no referido referencial.  $L_s$  representa o coeficiente de auto-indução dos circuitos estatóricos da máquina assíncrona,  $L_r$  representa o coeficiente de auto-indução dos circuitos rotóricos,  $M$  representa o coeficiente de indução mútua entre estator e rotor,  $\tau_s$  representa a constante de tempo associada aos circuitos estatóricos da máquina assíncrona,  $\tau_r$  representa a constante de tempo associada aos circuitos rotóricos da máquina assíncrona e  $\sigma$  representa o coeficiente de dispersão magnética.

$$\left\{ \begin{array}{l} \frac{di_{ds}}{dt} = -\left(\frac{1}{\sigma\tau_s} + \frac{1-\sigma}{\sigma\tau_r}\right)i_{ds} + \omega_R i_{qs} + \frac{1-\sigma}{\sigma\tau_r M} \psi_{dr} + \frac{1-\sigma}{\sigma M} \omega \psi_{qr} + \frac{1}{\sigma L_s} u_{ds} \\ \frac{di_{qs}}{dt} = -\omega_R i_{ds} - \left(\frac{1}{\sigma\tau_s} + \frac{1-\sigma}{\sigma\tau_r}\right)i_{qs} - \frac{1-\sigma}{\sigma M} \omega \psi_{dr} + \frac{1-\sigma}{\sigma\tau_r M} \psi_{qr} + \frac{1}{\sigma L_s} u_{qs} \\ \frac{d\psi_{dr}}{dt} = -\frac{1}{\tau_r} \psi_{dr} + (\omega_R - \omega) \psi_{qr} + \frac{M}{\tau_r} i_{ds} \\ \frac{d\psi_{qr}}{dt} = -(\omega_R - \omega) \psi_{dr} - \frac{1}{\tau_r} \psi_{qr} + \frac{M}{\tau_r} i_{qs} \\ \frac{d\omega}{dt} = \frac{M}{JL_r} (-i_{ds} \psi_{qr} + i_{qs} \psi_{dr}) - \frac{B}{J} \omega - \frac{1}{J} T_{ext} \end{array} \right. \quad ( 5.10 )$$

A última equação do modelo anterior é designada por equação mecânica, sendo o binário electromagnético resultado do produto cruzado de ambas as componentes das correntes estatórica e do fluxo magnético rotórico ( 5.11 ).

$$T_{em} = \frac{M}{L_r} (-i_{ds} \psi_{qr} + i_{qs} \psi_{dr}) \quad ( 5.11 )$$

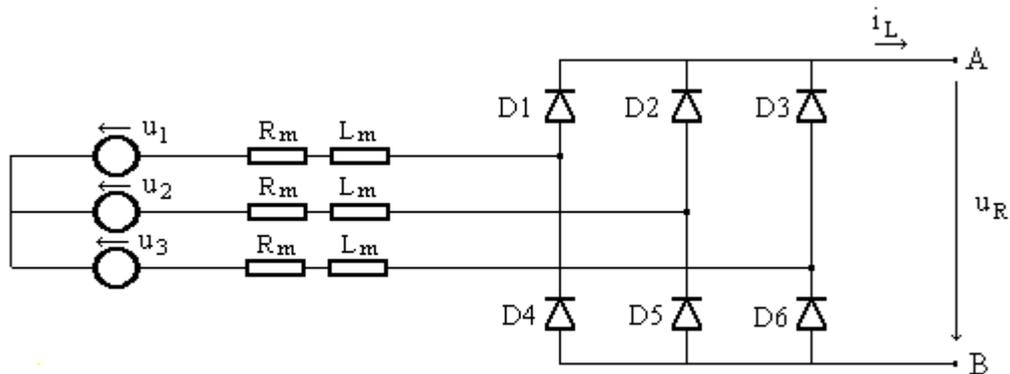
### 5.2.1.2 Conversor electrónico de potência

A velocidade de uma máquina assíncrona é condicionada, aparte dos aspectos construtivos, pela amplitude e frequência do sistema trifásico aplicado aos seus terminais. Em aplicações de accionamentos electromecânicos de velocidade variável é de todo o interesse possuir uma fonte trifásica que permita variar estas grandezas.

O uso de um ondulator de tensão autónomo possibilita a obtenção de um sistema trifásico de tensões, de amplitude e frequência variável, a partir de uma fonte de tensão contínua. O conversor electrónico de potência representado na Figura 5.2, é ainda constituído por um rectificador, por forma a obter uma tensão contínua, e por um filtro passa-baixo do tipo LC.

Associados a estes circuitos de potência, encontram-se os respectivos circuitos electrónicos de comando e controlo, que garantem também a necessária protecção do sistema. Estas protecções envolvem tanto a máquina assíncrona como o próprio conversor electrónico de potência. Na máquina assíncrona é necessário proteger os circuitos estatóricos de sobreintensidades, limitando a corrente que os percorre. No conversor electrónico de potência os dispositivos semicondutores apresentam-se bastante sensíveis a condições extremas de funcionamento, como sejam a presença de correntes ou frequências de comutação elevadas. Em presença de tais situações é fundamental inibir o seu funcionamento.

Como fonte de tensão contínua utilizou-se um rectificador trifásico em ponte, não comandado, representado na Figura 5.3. Este tipo de rectificador, bastante utilizado em aplicações industriais, proporciona uma tensão contínua de fraca ondulação. É composto por dois conjuntos de díodos, um de ânodo comum e outro de cátodo comum, permanentemente percorridos pela corrente de saída -  $i_L$ .



**Figura 5.3. Esquema de potência do rectificador trifásico em ponte.**

No conjunto superior conduz, em cada instante, o diódo que apresenta um maior potencial de ânodo e no conjunto inferior o que apresenta um menor potencial de cátodo. A tensão na carga é assim definida pela diferença instantânea entre os potenciais presentes nos terminais A e B. Admitindo como ideais tanto a fonte de tensão como os dispositivos utilizados, a tensão à saída do rectificador será dada por ( 5.12 ), onde  $u_k$  representa o sistema de tensões trifásico disponibilizado pela fonte.

$$u_R = \max\{u_k\} - \min\{u_k\} \quad , \quad k = 1,2,3 \quad (5.12)$$

A idealidade da fonte de tensão advém do facto de se considerar nula a sua impedância interna ( $R_m=L_m=0$ ). Deste modo as tensões disponibilizadas –  $u_k$  – encontram-se presentes aos terminais dos diódos e a corrente fornecida pode sofrer variações instantâneas. Para considerar os diódos ideais será necessário admitir que tanto a sua tensão de condução como a sua corrente de corte são nulas. Assim a corrente que percorre cada diódo será a corrente de saída, e não provocará neste qualquer queda de tensão.

A não consideração da idealidade da fonte de tensão trifásica, atendendo à sua indutância interna não nula –  $L_m$  –, ou ainda o facto de poderem existir transformadores de potência (com indutância elevada) a montante, implica que a corrente em qualquer

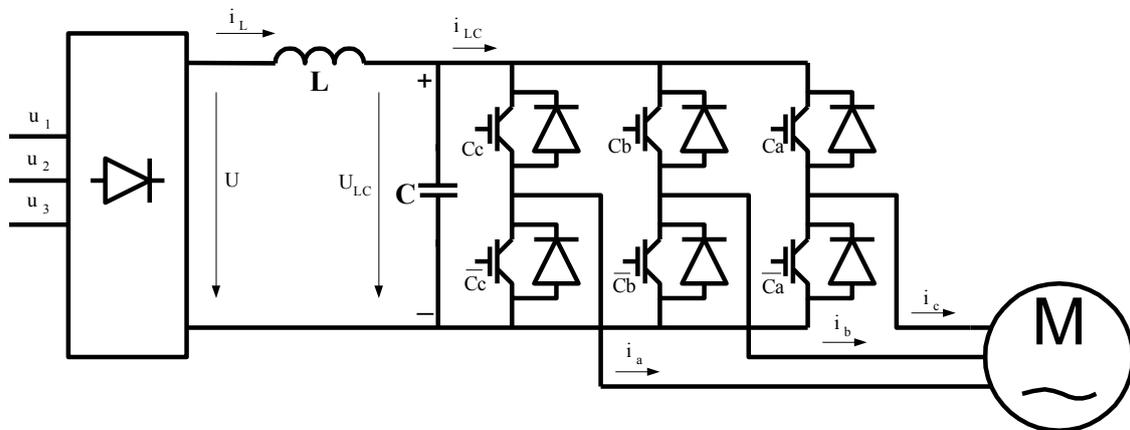
das fases não pode variar instantaneamente. Assim a comutação entre díodos não será instantânea originando o aparecimento do fenómeno de condução simultânea, que se traduz pela diminuição do valor da tensão  $u_R$ . Esta diminuição será tanto maior quanto maior for a corrente de saída, as indutâncias presentes ou a sua frequência (Palma, 1984). No entanto, e devido à elevada potência de curto-circuito normalmente apresentada pela rede de distribuição face à carga do rectificador e à não existência de transformadores de potência, a indutância a montante apresenta-se muito pequena, pelo que, para a gama de correntes em jogo – da ordem do ampere –, a queda de tensão originada pelo fenómeno da condução simultânea é geralmente desprezável.

Também a consideração da resistência interna da fonte origina uma diminuição na tensão de saída, que será tanto maior quanto maior for a corrente de saída e a própria resistência interna. Este valor é habitualmente desprezado pois a impedância da fonte de tensão trifásica apresenta-se essencialmente indutiva.

A consideração da queda de tensão limiar e da resistência incremental dos díodos provoca também uma diminuição no valor da tensão de saída. Contudo como a resistência incremental é muito pequena – da ordem da milésima de ohm –, para a gama das correntes em jogo é suficiente considerar que a queda de tensão directa se limita à queda de tensão limiar. Esta apresenta valores típicos da ordem do volt, pelo que a consideração desta parcela é perfeitamente desprezável face às centenas de volt presentes à saída do rectificador.

A utilização de um filtro LC, representado na Figura 5.4, possibilita a diminuição da ondulação da tensão de saída do rectificador, o alisar da corrente de saída –  $i_{LC}$  – e a recuperação de energia numa eventual situação de travagem do accionamento electromecânico. Este filtro passa-baixo reduz as frequências parasitas elevadas presentes na tensão de saída, por forma a que a tensão  $u_{LC}$  se apresente ‘mais lisa’ que a tensão  $u_R$ . As equações de funcionamento do filtro são facilmente obtidas a partir da teoria de circuitos ( 5.13 ).

$$\begin{cases} \frac{du_{LC}}{dt} = \frac{1}{C} i_C \\ i_{LC} = i_L - i_C \\ \frac{di_L}{dt} = -\frac{1}{L} (u_R + u_{LC}) \end{cases} \quad ( 5.13 )$$



**Figura 5.4. Esquema de potência do filtro LC e do ondulator autónomo de tensão.**

Como conversor electrónico para alimentar a máquina eléctrica foi utilizado um ondulator autónomo de tensão, também representado na Figura 5.4, onde foram utilizados transístores IGBT (*Isolated Gate Bipolar Transistor*) como interruptores electrónicos. Estes dispositivos electrónicos devem obrigatoriamente permanecer nos estados de corte e condução durante tempos mínimos característicos de cada dispositivo. A consideração destes factores, e de outros de carácter tecnológico, obrigam ao estabelecimento de tempos mínimos e tempos mortos no comando de cada interruptor. Assim, o sinal de comando para cada braço do ondulator não poderá ser complementar para cada interruptor, limitando superiormente a sua frequência de comutação,

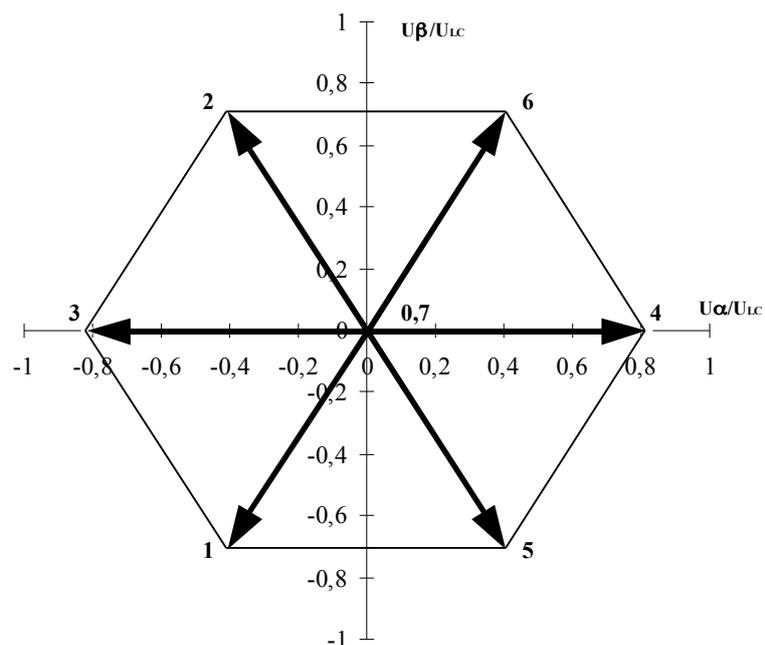
Este conversor apresenta oito possíveis combinações dos interruptores, a que correspondem sete estados distintos de tensão, traduzidos por seis vectores de igual amplitude e diferente orientação no sistema de coordenadas abc. Os estados resultantes de comandar simultaneamente todos os interruptores superiores ou inferiores traduzem-se num vector de amplitude nula.

Para caracterizar o estado de condução de cada braço definem-se as funções de comando  $c_k$ , com  $k=1,2,3$ , que assumem os valores 0 ou 1 conforme o interruptor que se pretende comandar em cada braço. O valor 1 associa-se à condução do interruptor superior e inibição do inferior, e o valor 0 à situação dual. Na Tabela 5.I encontram-se definidas as tensões referentes a cada combinação de interruptores e na Figura 5.5 os vectores de tensão correspondentes, definidos num sistema de coordenadas  $\alpha\beta$ .

**Tabela 5.I – Repartição das combinações dos interruptores do ondulator autónomo e respectivas tensões.**

Estado	$c_1$	$c_2$	$c_3$	$u_a$	$u_b$	$u_c$	$u_\alpha$	$u_\beta$
0	0	0	0	0	0	0	0	0
1	0	0	1	$-1/3 u_{LC}$	$-1/3 u_{LC}$	$2/3 u_{LC}$	$-1/\sqrt{6} u_{LC}$	$-1/\sqrt{2} u_{LC}$
2	0	1	0	$-1/3 u_{LC}$	$2/3 u_{LC}$	$-1/3 u_{LC}$	$-1/\sqrt{6} u_{LC}$	$1/\sqrt{2} u_{LC}$
3	0	1	1	$-2/3 u_{LC}$	$1/3 u_{LC}$	$1/3 u_{LC}$	$-\sqrt{(2/3)} u_{LC}$	0
4	1	0	0	$2/3 u_{LC}$	$-1/3 u_{LC}$	$-1/3 u_{LC}$	$\sqrt{(2/3)} u_{LC}$	0
5	1	0	1	$1/3 u_{LC}$	$-2/3 u_{LC}$	$1/3 u_{LC}$	$1/\sqrt{6} u_{LC}$	$-1/\sqrt{2} u_{LC}$
6	1	1	0	$1/3 u_{LC}$	$1/3 u_{LC}$	$-2/3 u_{LC}$	$1/\sqrt{6} u_{LC}$	$1/\sqrt{2} u_{LC}$
7	1	1	1	0	0	0	0	0

$u_{LC}$  - Valor da fonte de tensão contínua, após o filtro LC.



**Figura 5.5. Vectores de tensão, representados num sistema de coordenadas  $\alpha\beta$ , gerados pelo ondulator autónomo.**

Admitindo carácter ideal para os interruptores as tensões compostas à saída do ondulator autónomo são dadas por ( 5.14 ) (Santana e Labrique, 1991). Este carácter ideal traduz-se pela possibilidade de comutação instantânea e por tensão de condução e corrente de corte nulas, associadas aos interruptores electrónicos.

$$\begin{cases} u_{ab} = (c_1 - c_2)u_{LC} \\ u_{bc} = (c_2 - c_3)u_{LC} \\ u_{ca} = (c_3 - c_1)u_{LC} \end{cases} \quad (5.14)$$

Num referencial dq genérico, a rodar a uma velocidade  $\omega_R$ , as tensões anteriores de alimentação da máquina assíncrona são dadas por ( 5.15 ), onde  $\theta_R = \omega_R t + \theta_{R0}$  representa a posição do referido referencial, com posição inicial  $\theta_{R0}$ .

$$\begin{cases} u_{ds} = \left( \sqrt{\frac{2}{3}} \cos \theta_R c_1 + \left( -\frac{1}{\sqrt{6}} \cos \theta_R + \frac{1}{\sqrt{2}} \operatorname{sen} \theta_R \right) c_2 + \left( -\frac{1}{\sqrt{6}} \cos \theta_R - \frac{1}{\sqrt{2}} \operatorname{sen} \theta_R \right) c_3 \right) u_{LC} \\ u_{qs} = \left( -\sqrt{\frac{2}{3}} \operatorname{sen} \theta_R c_1 + \left( \frac{1}{\sqrt{2}} \cos \theta_R + \frac{1}{\sqrt{6}} \operatorname{sen} \theta_R \right) c_2 + \left( -\frac{1}{\sqrt{2}} \cos \theta_R + \frac{1}{\sqrt{6}} \operatorname{sen} \theta_R \right) c_3 \right) u_{LC} \end{cases} \quad (5.15)$$

A corrente proveniente do filtro LC pode ser expressa como uma combinação linear das correntes em cada braço do ondulator autónomo ( 5.16 ).

$$i_{LC} = \sum_{k=1}^3 c_k i_k \quad (5.16)$$

As equações apresentadas nesta secção descrevem o comportamento dinâmico do conversor electrónico de potência assumindo como válidos os pressupostos considerados.

### 5.2.1.3 Modelo de estado do accionamento electromecânico

A consideração da modelização apresentada para a máquina assíncrona e para o conversor electrónico de potência origina o modelo de estado ( 5.17 ), susceptível de representar o comportamento dinâmico do accionamento electromecânico de velocidade variável, utilizado na montagem laboratorial.

$$\left\{ \begin{array}{l}
 \frac{di_{ds}}{dt} = -\left(\frac{1}{\sigma\tau_s} + \frac{1-\sigma}{\sigma\tau_r}\right)i_{ds} + \omega_R i_{qs} + \frac{1-\sigma}{\sigma\tau_r M} \psi_{dr} + \frac{1-\sigma}{\sigma M} \omega \psi_{qr} + \frac{1}{\sigma L_s} u_{ds} \\
 \frac{di_{qs}}{dt} = -\omega_R i_{ds} - \left(\frac{1}{\sigma\tau_s} + \frac{1-\sigma}{\sigma\tau_r}\right)i_{qs} - \frac{1-\sigma}{\sigma M} \omega \psi_{dr} + \frac{1-\sigma}{\sigma\tau_r M} \psi_{qr} + \frac{1}{\sigma L_s} u_{qs} \\
 \frac{d\psi_{dr}}{dt} = -\frac{1}{\tau_r} \psi_{dr} + (\omega_R - \omega) \psi_{qr} + \frac{M}{\tau_r} i_{ds} \\
 \frac{d\psi_{qr}}{dt} = -(\omega_R - \omega) \psi_{dr} - \frac{1}{\tau_r} \psi_{qr} + \frac{M}{\tau_r} i_{qs} \\
 \frac{d\omega}{dt} = \frac{M}{JL_r} (-i_{ds} \psi_{qr} + i_{qs} \psi_{dr}) - \frac{B}{J} \omega - \frac{1}{J} T_{ext} \\
 u_{ds} = \left( \sqrt{\frac{2}{3}} \cos \theta_R c_1 + \left( -\frac{1}{\sqrt{6}} \cos \theta_R + \frac{1}{\sqrt{2}} \sin \theta_R \right) c_2 + \left( -\frac{1}{\sqrt{6}} \cos \theta_R - \frac{1}{\sqrt{2}} \sin \theta_R \right) c_3 \right) u_{LC} \\
 u_{qs} = \left( -\sqrt{\frac{2}{3}} \sin \theta_R c_1 + \left( \frac{1}{\sqrt{2}} \cos \theta_R + \frac{1}{\sqrt{6}} \sin \theta_R \right) c_2 + \left( -\frac{1}{\sqrt{2}} \cos \theta_R + \frac{1}{\sqrt{6}} \sin \theta_R \right) c_3 \right) u_{LC} \\
 i_{LC} = \sqrt{\frac{2}{3}} (i_{ds} \cos \theta_R - i_{qs} \sin \theta_R) c_1 + \frac{1}{\sqrt{6}} (i_{ds} (-\cos \theta_R + \sqrt{3} \sin \theta_R) + i_{qs} (-\sqrt{3} \cos \theta_R + \sin \theta_R)) c_2 + \\
 + \frac{1}{\sqrt{6}} (i_{ds} (-\cos \theta_R - \sqrt{3} \sin \theta_R) + i_{qs} (-\sqrt{3} \cos \theta_R + \sin \theta_R)) c_3 \\
 i_L = i_{LC} - i_C \\
 \frac{du_{LC}}{dt} = \frac{1}{C} i_C \\
 \frac{di_L}{dt} = -\frac{1}{L} (u_R + u_{LC}) \\
 u_R = \max\{u_1, u_2, u_3\} - \min\{u_1, u_2, u_3\}
 \end{array} \right. \quad (5.17)$$

É um modelo de estado não linear composto por um conjunto de 12 equações. Consideram-se como comandos do sistema dinâmico as funções de comando dos interruptores do ondulator autónomo de tensão –  $c_k$  –, e como variável de saída a velocidade angular da máquina assíncrona –  $\omega$ .

Este modelo foi obtido considerando como filiformes os circuitos da máquina assíncrona e como ideais os dispositivos semicondutores do conversor electrónico de potência. Estas considerações levam a ignorar certos fenómenos, dependentes da temperatura ou relacionados com a frequência de comutação máxima dos dispositivos semicondutores, que podem comprometer a precisão do modelo, quer por variação dos seus parâmetros quer por alterações nas suas equações. Este modelo apresenta ainda dificuldades de utilização devidas ao elevado número de parâmetros e de equações (e consequentemente de variáveis) presentes, bem como devido às variações a que todos eles podem estar sujeitos.

## 5.2.1.4 Simplificação do modelo de estado

Atendendo ao elevado número de variáveis consideradas na relação funcional o processo de aprendizagem apresentar-se-ia extremamente complexo, inviabilizando qualquer tentativa de aprendizagem em tempo real. Irá assim efectuar-se um estudo mais aprofundado das características físicas do accionamento para verificar a possibilidade de diminuir o número de argumentos presentes na referida relação funcional.

### 5.2.1.4.1 Controlo de corrente

Por razões de ordem prática, como sejam a protecção dos circuitos da máquina e a simplificação da sua dinâmica, optou-se por efectuar o controlo das correntes fornecidas pelo conversor electrónico de potência. Devido à sua eficácia e fácil implementação prática, escolheu-se o controlo por modo de deslizamento (Utkin, 1977) (Utkin, 1987). Este tipo de controlo pode exibir respostas rápidas, pequeno *ripple*, e elevada robustez contra perturbações e variações de parâmetros. A estabilidade não depende dos parâmetros da carga desde que o valor da tensão contínua do conversor seja suficientemente elevado para impor as derivadas de corrente adequadas em todas as condições de funcionamento.

Este controlo, particularmente bem adaptado a sistemas de estrutura variável, como o conversor electrónico de potência utilizado, caracteriza-se por admitir descontinuidades no controlo e traduz-se na evolução do estado do sistema ao longo de uma trajectória – equação de comutação ( 5.18 ) – que representa uma relação entre as variáveis de estado.

$$S(\mathbf{x}, t) = 0 \quad ( 5.18 )$$

O comando gerado, dado por ( 5.19 ), deverá contrariar a tendência apresentada pela equação de comutação de modo a manter o estado do sistema na trajectória desejada ( 5.18 ).

$$u = \begin{cases} U^+ & \text{se } S(\mathbf{x}, t) > 0 \\ U^- & \text{se } S(\mathbf{x}, t) < 0 \end{cases} \quad (5.19)$$

Uma vez que o sistema trifásico que alimenta o accionamento electromecânico não possui neutro ligado é possível representar as correntes que alimentam a máquina assíncrona num referencial  $\alpha\beta$ , sem componente homopolar ( 5.20 ).

$$\begin{aligned} i_a + i_b + i_c &= 0 \\ i_\alpha &= \sqrt{\frac{3}{2}} i_a \\ i_\beta &= \frac{1}{\sqrt{2}} (i_a + 2i_b) \end{aligned} \quad (5.20)$$

Assim, o comando por modo de deslizamento deverá ser tal que ambas as correntes  $-i_\alpha$  e  $i_\beta$  sigam as respectivas referências ( 5.21 ), onde  $\varepsilon_{\alpha\beta}$  representa o erro.

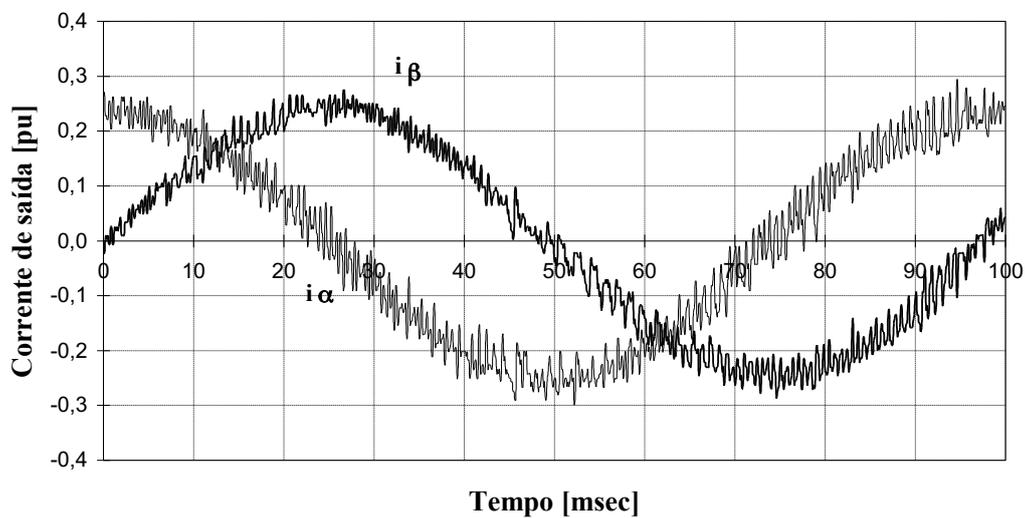
$$\varepsilon_{\alpha\beta} = i_{\alpha\beta\text{ref}} - i_{\alpha\beta} \rightarrow 0 \quad (5.21)$$

Foi desenvolvida uma nova técnica de comando que, através de uma escolha cuidada de todos os vectores de tensão disponíveis, minimiza o erro de corrente (Martins, Pires e Silva, 1998). Esta técnica utiliza comparadores de três níveis, que analisam os erros de corrente ( 5.21 ) produzindo os sinais  $\delta_{\alpha\beta}$ , e duas condições adicionais (‘ $|\varepsilon_{ib}| > |\varepsilon_{ic}|$ ’ e ‘Dois interruptores superiores ligados’, onde  $\varepsilon_{ib}$  e  $\varepsilon_{ic}$  denotam os erros de corrente respectivamente nas fases ‘b’ e ‘c’ do ondulator autónomo de tensão) que são utilizadas como uma palavra digital que, através de uma tabela implementada em EPROM (*Erasable Programmable Read Only Memory*), gera os sinais de comando dos interruptores, tal como se descreve na Tabela 5.II.

**Tabela 5.II – Resultados dos comparadores de três níveis, condições adicionais e respectivos vectores de tensão.**

$\delta_a$	$\delta_b$	Dois interruptores superiores ligados	$ \epsilon i_b  >  \epsilon i_c $	$U_k$
0	0	não , sim	x	$U_0, U_7$
0	+1	x	sim , não	$U_2, U_6$
0	-1	x	não , sim	$U_1, U_5$
+1	0	x	x	$U_4$
+1	+1	x	x	$U_6$
+1	-1	x	x	$U_5$
-1	0	x	x	$U_3$
-1	+1	x	x	$U_2$
-1	-1	x	x	$U_1$

Apresenta-se na Figura 5.6, um registo experimental referente ao controlo de corrente efectuado no ondulador autónomo, onde se visualiza a evolução das componentes  $\alpha\beta$  da corrente que percorre os enrolamentos estatóricos da máquina assíncrona.



**Figura 5.6. Evolução das componentes  $\alpha\beta$  das correntes reais que percorrem os enrolamentos estatóricos da máquina assíncrona.**

O controlo considerado para o conversor electrónico de potência apresenta bons resultados, nas condições da sua existência, quando aplicado ao controlo das correntes

que alimentam a máquina assíncrona, pelo que é possível assumir um efectivo controlo das mesmas. A opção por este controlo simplifica o modelo anteriormente considerado para o accionamento electromecânico. Desprezando a dinâmica associada ao conversor electrónico de potência o modelo do accionamento resume-se à dinâmica associada à máquina assíncrona e à carga mecânica ( 5.22 ), onde se consideram como comandos as referências das componentes das correntes que percorrem o estator da máquina assíncrona e como variáveis de estado as componentes dos fluxos de ligação magnéticos rotóricos e a velocidade angular do rotor. Continua a considerar-se como variável de saída do accionamento electromecânico a velocidade angular da máquina assíncrona.

$$\begin{cases} \frac{d\psi_{dr}}{dt} = -\frac{1}{\tau_r}\psi_{dr} + (\omega_R - \omega)\psi_{qr} + \frac{M}{\tau_r}i_{dsref} \\ \frac{d\psi_{qr}}{dt} = -(\omega_R - \omega)\psi_{dr} - \frac{1}{\tau_r}\psi_{qr} + \frac{M}{\tau_r}i_{qsref} \\ \frac{d\omega}{dt} = \frac{M}{JL_r}(-i_{dsref}\psi_{qr} + i_{qsref}\psi_{dr}) - \frac{B}{J}\omega - T_{ext} \end{cases} \quad ( 5.22 )$$

Este modelo apresenta-se válido se, além da validade dos vários pressupostos enunciados aquando do seu estabelecimento, as componentes das correntes estatóricas se encontrarem controladas dentro de uma banda de erro suficientemente pequena, quando comparada com as correntes que circulam na máquina.

#### 5.2.1.4.2 Orientação de campo

No início da década de 70, Blaschke introduziu os princípios do controlo da máquina de indução por orientação de campo (Blaschke, 1972). Este trabalho foi inspirado no comportamento dinâmico da máquina de corrente contínua de excitação independente, onde existe, para a máquina ideal, um desacoplamento magnético entre as correntes do circuito indutor e induzido.

Assumindo que o referencial genérico dq a rodar a uma velocidade genérica  $\omega_R$  coincide com o referencial do fluxo rotórico ( $\psi_{dr}=\psi_r$  e  $\psi_{qr}=0$ ), o conjunto de equações anteriores transformam-se em ( 5.23 ).

$$\begin{cases} \frac{d\psi_r}{dt} = -\frac{1}{\tau_r}\psi_r + \frac{M}{\tau_r}i_{dsref} \\ 0 = -(\omega_R - \omega)\psi_r + \frac{M}{\tau_r}i_{qsref} \\ \frac{d\omega}{dt} = \frac{M}{JL_r}i_{qsref}\psi_r - \frac{B}{J}\omega - T_{ext} \end{cases} \quad (5.23)$$

Neste referencial particular, a dinâmica associada à amplitude do fluxo rotórico está relacionada com a componente d da corrente estatórica. Assim, colocando esta componente constante é possível obter um desacoplamento análogo ao existente na máquina de corrente contínua. O binário resulta do produto entre a amplitude do fluxo rotórico e da componente q da mesma corrente estatórica.

Assumindo que o fluxo magnético já se encontra em regime permanente, a velocidade é assim apenas comandada pela evolução da componente q da corrente estatórica (que se assume perfeitamente controlada). Deste modo, e assumindo todas as simplificações anteriormente descritas, estamos perante um sistema de 1ª ordem ( 5.24 ) relativamente à velocidade do accionamento electromecânico.

$$\frac{d\omega}{dt} = \frac{M\psi_r}{JL_r}i_{qsref} - \frac{B}{J}\omega - T_{ext} \quad (5.24)$$

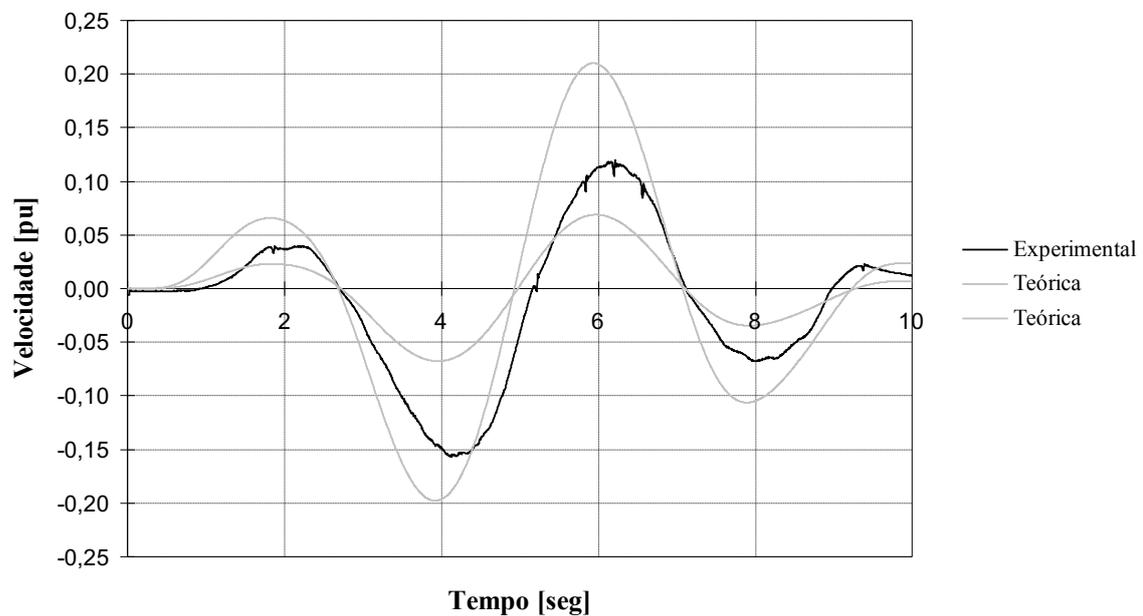
### 5.2.1.4.3 Dependência de parâmetros

Técnicas de comando baseadas no princípio da orientação de campo são bastante utilizadas, com bons resultados, em várias aplicações práticas com recurso ao uso de cadeias de retroacção. No entanto a sua validade encontra-se dependente de um bom conhecimento dos parâmetros envolvidos. Eventuais determinações imprecisas ou alterações comprometem a precisão do modelo. A constante de tempo mecânica ( 5.25 ) depende do coeficiente de atrito, o qual é difícil de determinar. A constante de tempo rotórica ( 5.26 ) depende da resistência rotórica, a qual varia, por exemplo, com a temperatura.

$$\tau_m = \frac{B}{J} \quad (5.25)$$

$$\tau_r = \frac{L_r}{R_r} \quad (5.26)$$

Na Figura 5.7 apresenta-se, para as mesmas componentes das referências das correntes estatóricas, as respostas experimental e teórica provenientes do modelo ( 5.24 ), quando este é sujeito a uma variação de 50%, aditiva e subtractiva, na constante de tempo mecânica.



**Figura 5.7. Resposta experimental e teórica (modelo sujeito a uma variação de 50% na constante de tempo mecânica).**

É possível constatar a forte dependência da resposta teórica face a uma variação nos parâmetros do modelo. Este resultado reforça a necessidade de recorrer a uma aprendizagem contínua, por forma a manter o modelo permanentemente actualizado. A variação ou má determinação dos parâmetros de um modelo comprometem a sua validade.

Para a implementação de controladores baseados no princípio da orientação de campo tanto os parâmetros como a posição do referencial dq do fluxo rotórico devem ser conhecidos com rigor. No entanto, tal não corresponde exactamente à realidade. Um dos maiores problemas reside no conhecimento exacto da posição do referencial. Os métodos utilizados para a obtenção de tal posição são classificados na literatura em métodos directos ou indirectos (Sen, 1990). Não considerando a utilização do método directo, por obrigar a alterações construtivas na máquina, a solução usual consiste na utilização de observadores de fluxo.

Como no referencial do fluxo rotórico a componente q do fluxo é nula, é possível estabelecer uma relação entre as velocidades mecânica e do referencial e a componente q da corrente estatórica. Esta relação ( 5.28 ) obtém-se a partir da segunda equação do modelo ( 5.23 ).

$$\omega_R = \omega + \frac{M}{\tau_r} \frac{i_{qsref}}{\psi_r} \quad ( 5.27 )$$

Um dos observadores mais simples considerados na literatura consiste em integrar a equação anterior por forma a obter a evolução do ângulo de posição do referencial do fluxo.

$$\hat{\alpha} = \hat{\alpha}_0 + \int_0^t \left( \omega_R + \frac{M i_{qsref}}{\tau_r \psi_r} \right) dt \quad ( 5.28 )$$

A consideração deste observador faz subir a ordem do sistema, bem como a sua complexidade e não-linearidade. Assumindo que as correntes se encontram perfeitamente controladas, o sistema resultante é estável e o observador  $\hat{\alpha}(t)$  converge efectivamente para  $\alpha(t)$ . As constantes de tempo associadas à dinâmica do fluxo e do erro do observador coincidem com a constante de tempo rotórica. Sendo esta bastante inferior à constante de tempo mecânica, o comportamento geral do sistema é fundamentalmente ditado por esta. Refira-se ainda que o bom desempenho do observador depende do conhecimento exacto dos parâmetros da máquina, nomeadamente a constante de tempo rotórica e o coeficiente de indução mútua, o que nem sempre é possível.

## 5.2.2 Relação funcional

Como visto na secção 5.2.1 o comportamento dinâmico do accionamento electromecânico de velocidade variável em estudo, pode ser representado por um conjunto de equações de estado não lineares do tipo ( 3.1 ).

$$\begin{cases} \dot{\mathbf{x}}(t) = \Phi(\mathbf{x}(t), \mathbf{u}(t)), \mathbf{x}_0 \\ \mathbf{y}(t) = \Psi(\mathbf{x}(t)) \end{cases} \quad (3.1)$$

Como descrito na secção 3.4, para um qualquer sistema dinâmico entende-se por aprendizagem a extracção, baseada num conjunto de exemplos, de uma relação funcional ( 3.5 ) representativa do funcionamento do sistema em causa, onde  $\mathbf{X}$  representa um conjunto de variáveis significativas.

$$\mathbf{y} = f(\mathbf{X}) \quad (3.5)$$

Sendo a geração de variáveis de comando e a aquisição experimental de variáveis de estado e saída baseada num sistema digital, constituído por um computador pessoal equipado com uma carta de aquisição e comando, o sistema pode ser descrito por um modelo de estado discreto ( 3.7 ), tal como analisado na secção 3.5.

$$\begin{cases} \mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k) \quad , \quad \mathbf{x}_0 \\ \mathbf{y}_k = h(\mathbf{x}_k, \mathbf{u}_k) \end{cases} \quad (3.7)$$

Neste modelo discreto o sistema digital deverá obedecer a alguns requisitos: as entradas de comando –  $\mathbf{u}_k$  – devem ser contínuas à direita durante todo o intervalo de amostragem e mantidas constantes nesse mesmo intervalo ( 5.29 ), o qual deverá ser uniforme ( 5.30 ).

$$\forall t \in [t_k, t_{k+1}[ \Rightarrow \mathbf{u}(t) = \mathbf{u}(t_k) \quad (5.29)$$

$$t_{k+1} = t_k + \Delta T \quad , \quad \Delta T = \text{cte} \quad (5.30)$$

O sistema obtido assume-se como estável, com parametrização conhecida, mas com os valores dos parâmetros não totalmente conhecidos. Uma possível relação funcional representativa da dinâmica do accionamento electromecânico será dada por ( 3.8 ), sendo as relações entre as variáveis impostas por ( 3.7 ).

$$\mathbf{y}_{k+1} = \mathbf{g}_{\mathbf{x}_0, \mathbf{p}}(\mathbf{x}_k, \mathbf{u}_k) \quad (3.8)$$

Atendendo à não linearidade do modelo de estado, bem como à incerteza no conhecimento de alguns parâmetros, apresenta-se difícil o determinar, com exactidão, a relação funcional ( 3.8 ). Por esta razão recorre-se ao uso de algoritmos de aprendizagem, os quais, de uma forma automática, permitem extrair uma relação funcional ( 5.31 ) que supostamente representa a dinâmica do accionamento.

$$\hat{\mathbf{y}}_{k+1} = \hat{\mathbf{g}}_{\mathbf{x}_0, \mathbf{p}}(\mathbf{x}_k, \mathbf{u}_k) \quad (5.31)$$

Esta relação funcional será uma boa representação do accionamento se, quando sujeita a um mesmo comando  $-\mathbf{u}_k-$  e estado  $-\mathbf{x}_k-$ , produzir uma saída  $-\hat{\mathbf{y}}_{k+1}-$  que se encontre suficientemente próxima da saída real  $-\mathbf{y}_{k+1}-$  do accionamento ( 5.32 ).

$$\|\mathbf{y}_{k+1} - \hat{\mathbf{y}}_{k+1}\| = \|\mathbf{g}_{\mathbf{x}_0, \mathbf{p}}(\mathbf{x}_k, \mathbf{u}_k) - \hat{\mathbf{g}}_{\mathbf{x}_0, \mathbf{p}}(\mathbf{x}_k, \mathbf{u}_k)\| < \varepsilon, \quad \forall \mathbf{x}_k, \mathbf{u}_k \quad (5.32)$$

Para o caso concreto em análise, e considerando as simplificações que originaram o modelo de estado de 1ª ordem ( 5.24 ) para a dinâmica da velocidade do sistema de accionamento electromecânico, a relação funcional descritiva do seu comportamento terá a forma ( 5.33 ), aparte de eventuais não linearidades excluídas do modelo.

$$\omega_{k+1} = f(\omega_k, i_{qs_k}) \quad (5.33)$$

Nesta relação funcional assume-se que o conversor electrónico de potência providencia que a componente q da corrente estatórica siga a referência imposta e que tanto os parâmetros como a posição do referencial são conhecidos com rigor.

### 5.2.3 Especificação da gramática

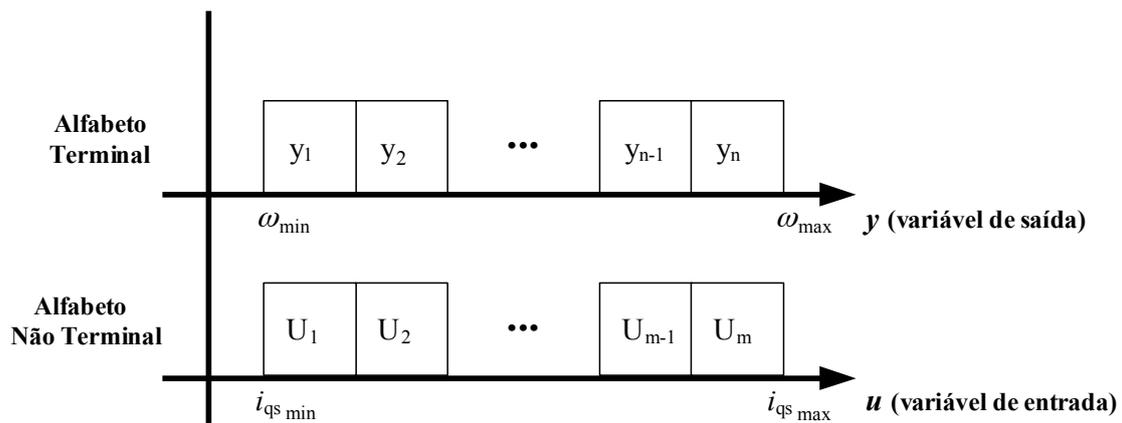
Por forma a inferir a gramática que descreve a linguagem associada ao accionamento electromecânico, torna-se necessário especificar os alfabetos terminal e não terminal para, a partir da análise dos dados experimentais de entrada/saída, inferir o conjunto de produções. Admite-se a existência de um símbolo inicial  $S$  e do símbolo não terminal  $\delta$  utilizado nas produções ( 3.18 ).

#### 5.2.3.1 Alfabetos terminal e não terminal

Mantendo a componente  $d$  da corrente estatórica num valor apropriado constante, por forma a manter um adequado nível de magnetização da máquina assíncrona, a variável de comando considerada será a componente  $q$  da mesma corrente ( $i_{qs}$ ) e a variável de saída a velocidade da máquina assíncrona ( $\omega$ ). Assumindo como válidas as anteriores simplificações, considera-se ( 5.33 ) como uma possível relação funcional descritiva do comportamento dinâmico do accionamento modelizado por ( 5.24 ).

Deste modo o alfabeto não terminal será obtido a partir da quantificação da variável de comando ( $i_{qs}$ ) e o alfabeto terminal a partir da quantificação da variável de saída ( $\omega$ ), como definido em ( 5.34 ). Assumindo esta codificação da informação disponível, os alfabetos são estabelecidos dividindo o domínio das variáveis do sistema de accionamento electromecânico em intervalos iguais, e associando a cada um deles um símbolo do alfabeto, como se apresenta na Figura 5.8.

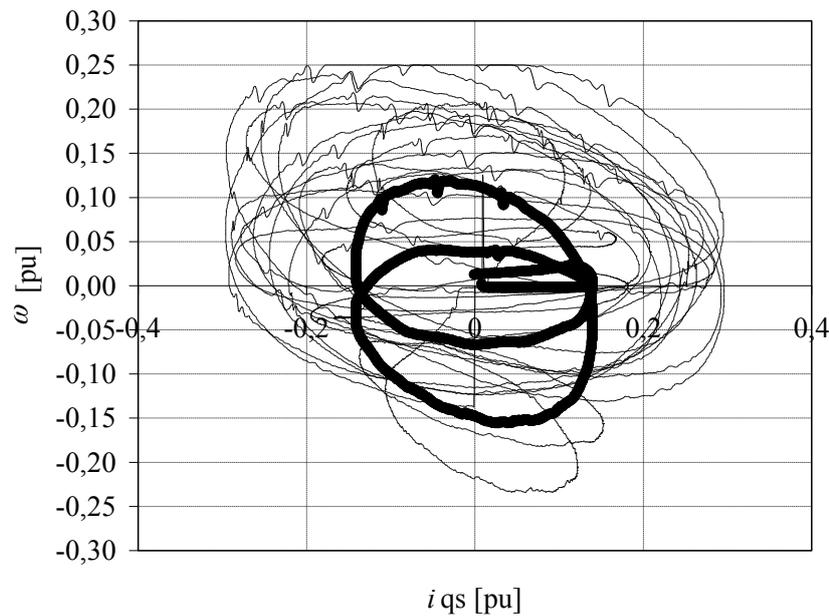
$$\begin{array}{l} \omega \xrightarrow{\text{codificação}} y_j \in \{y_1, y_2, \dots, y_n\} \\ i_{qs} \xrightarrow{\text{codificação}} U_i \in \{U_1, U_2, \dots, U_m\} \end{array} \quad ( 5.34 )$$



**Figura 5.8. Codificação da informação para os alfabetos terminal e não terminal.**

Para obter um conjunto de treino representativo considerou-se, para a variável de comando no accionamento electromecânico ( $i_{qs}$ ), um sinal obtido a partir de uma combinação de diferentes amplitudes e frequências por forma a obter valores da variável de saída ( $\omega$ ) no domínio de funcionamento do accionamento. Desta forma, os valores de amplitude e frequência podem ser estabelecidos até ao limite da resposta do accionamento, evitando o problema da filtragem, e permitindo que os dados experimentais obtidos constituam um bom conjunto de treino representativo do domínio de funcionamento do accionamento.

A aquisição do conjunto de treino é um passo essencial para obter um bom conhecimento do sistema que se pretende representar. Este conjunto deverá abranger uma parte significativa do domínio de funcionamento, pelo que é desejável que caracterize as zonas de funcionamento usual. Caso o accionamento altere a sua zona de funcionamento habitual o algoritmo de aprendizagem através de exemplos deverá adaptar o conjunto de treino por forma a incluir esta nova informação. A Figura 5.9 representa a evolução dos conjuntos de treino e de teste, no espaço entrada/saída, sendo que o conjunto de teste é apresentado a traço carregado.



**Figura 5.9. Conjuntos de treino e teste para o accionamento electromecânico.**

Para analisar a validade da gramática inferida consideraram-se dois tipos distintos de alfabetos. O primeiro utiliza um intervalo de quantificação com 0,01 ‘por unidade’ (pu), tanto para a variável de comando como para a de saída, o que se traduz em alfabetos terminal e não terminal com 60 símbolos. O segundo utiliza um menor intervalo de quantificação com 0,1 ‘por unidade’ (pu), também para ambas as variáveis, o que se traduz em alfabetos terminal e não terminal com apenas 6 símbolos. Aos alfabetos não terminais é necessário adicionar os símbolos especiais S e  $\delta$ .

Após a especificação do alfabeto a evolução das variáveis do conjunto de treino e de teste são codificadas em palavras pertencentes à linguagem que se assume provir do accionamento.

### 5.2.3.2 Produções

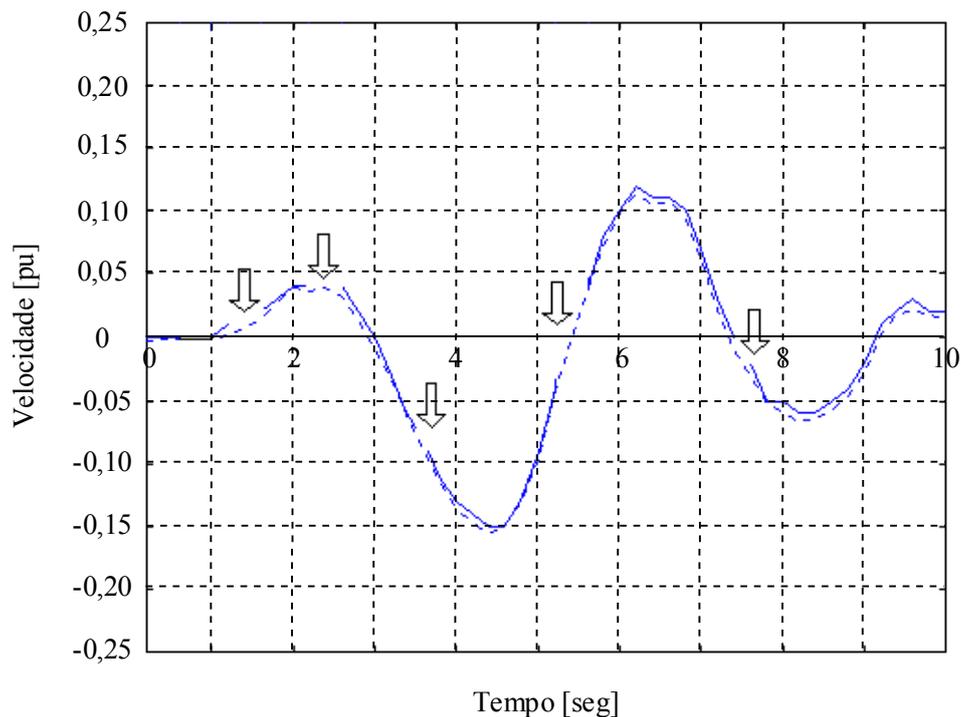
Para as produções, da forma ( 3.18 ), considerou-se  $p_{\max}=2$  como o tipo máximo ( 5.35 ). Como já visto anteriormente, para evitar a influência do ruído nas produções inferidas uma produção só é considerada válida se fôr extraída um número suficiente de vezes, funcionando este procedimento como um filtro contra as perturbações devidas à presença de ruído.

$$y_1 y_2 U_2 \rightarrow y_1 y_2 y_3 \delta \quad ( 5.35 )$$

### 5.2.4 Resultados experimentais

Relativamente ao primeiro alfabeto considerado, e após o processo de inferência das produções, foram inferidas 183 produções com a seguinte distribuição: 5 *produções de tipo 0*, 156 *produções de tipo 1* e 22 *produções de tipo 2*. A forte predominância de *produções do tipo 1* evidencia o carácter de 1ª ordem do modelo considerado para o accionamento ( 5.24 ), não sendo os resultados influenciados pela dinâmica do observador da posição do fluxo magnético.

Aplicando a gramática obtida ao conjunto de teste, obtêm-se os resultados de modelização apresentados na Figura 5.10. A velocidade do accionamento electromecânico é representada por uma linha de traço interrompido e a resposta da gramática por uma linha contínua.

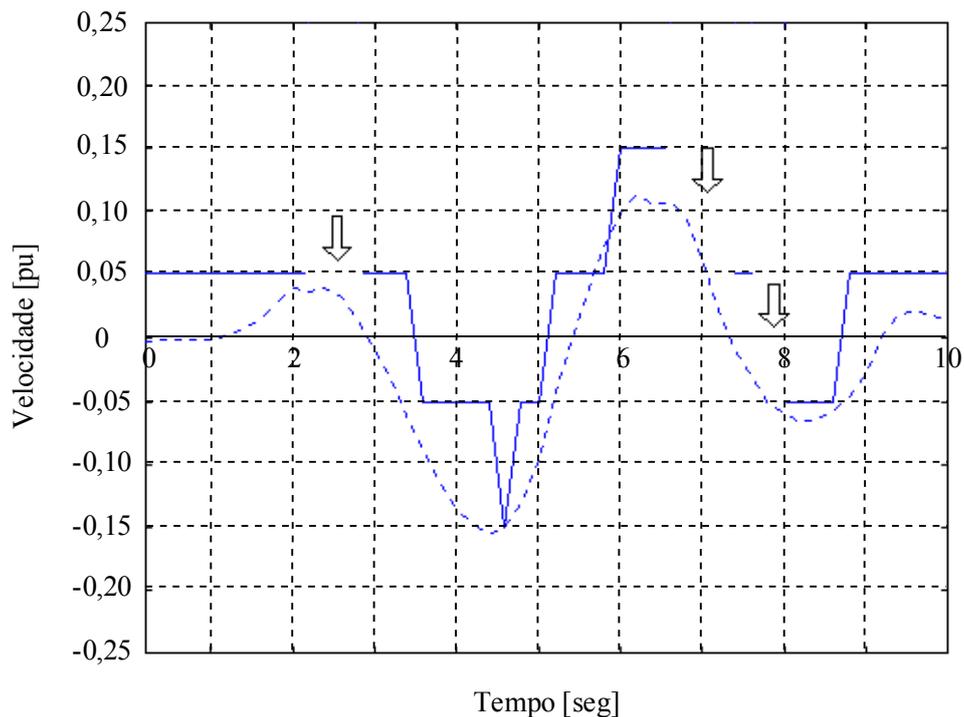


**Figura 5.10. Resultados de modelização com um alfabeto de 60 símbolos.**

Não considerando a pequena diferença de resultados resultante do processo de quantificação, os resultados podem ser considerados satisfatórios pois a evolução da velocidade do accionamento e a resposta da gramática são similares. Será importante salientar que existem algumas situações, referenciadas por setas brancas, onde a gramática não providencia nenhuma resposta – lacunas gramaticais. Isto acontece porque nenhuma produção foi inferida representando aquelas particulares relações entrada/saída. Mais adiante analisar-se-á como o mecanismo de interpolação gramatical pode minimizar este problema.

Relativamente ao segundo alfabeto, e após o processo de inferência das produções, foram inferidas 36 produções com a seguinte distribuição: 0 produções de tipo 0, 4 produções de tipo 1 e 32 produções de tipo 2.

Aplicando a gramática obtida ao conjunto de teste, obtêm-se os resultados de modelização apresentados na Figura 5.11. Tal como anteriormente, velocidade do accionamento electromecânico é representada por uma linha de traço interrompido e a resposta da gramática por uma linha contínua.



**Figura 5.11. Resultados de modelização com um alfabeto de 6 símbolos.**

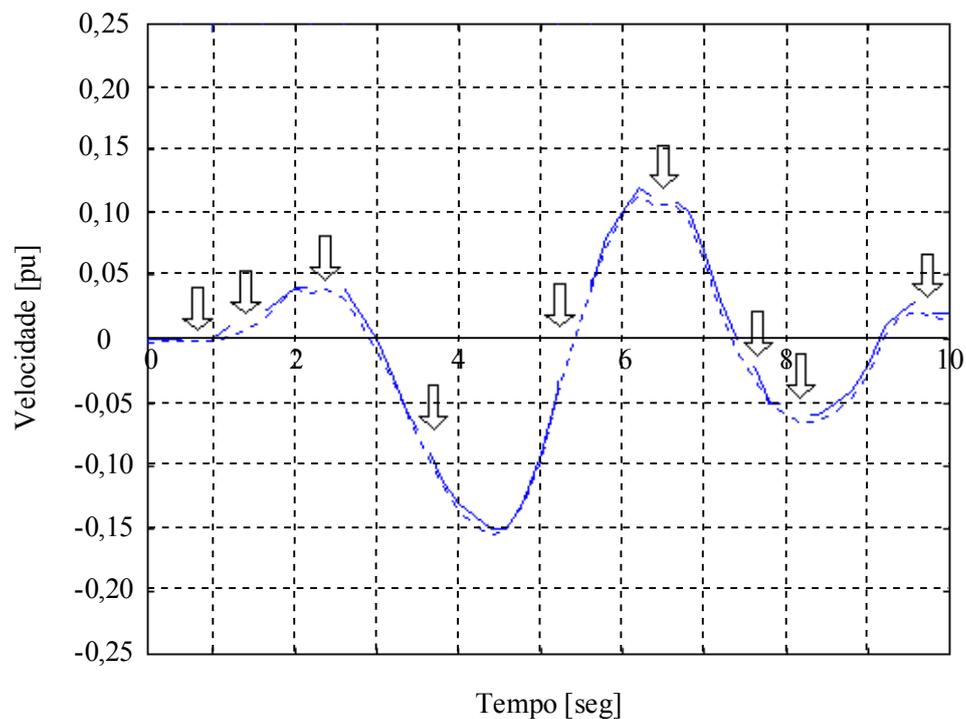
Tal como esperado, os resultados de modelização deterioram-se do ponto de vista quantitativo, em virtude de terem sido considerados alfabetos de muito menor dimensão. Existe uma enorme diferença quantitativa entre a velocidade do accionamento e a resultante da reconstrução gramatical. No entanto, quando a modelização é considerada qualitativamente os resultados podem ser considerados bastante satisfatórios, sendo que a evolução da resposta gramatical acompanha a evolução real da velocidade. Para processos de reconstrução qualitativa a consideração de um menor alfabeto não constitui uma dificuldade, apresentando como vantagem um mais fácil processamento. Contudo continuam a existir, tal como em alfabetos de maior dimensão, os problemas associados à existência de lacunas gramaticais.

### 5.2.4.1 Interpolação gramatical

A quantificação discreta subjacente à criação dos alfabetos implica uma resposta também discreta quando se efectua uma modelização de um accionamento electromecânico. Por forma a melhorar as capacidades de reconstrução quantitativa podem ser consideradas duas soluções. A primeira passa por aumentar o número de

símbolos do alfabeto, isto é, aumentar o nível de discretização aquando da sua construção. Esta solução aumenta necessariamente o número de produções inferidas dificultando o processo de reconstrução. Alternativamente, pode considerar-se a aplicação do mecanismo de interpolação gramatical. Este procedimento, descrito na secção 3.5.3, não aumenta arbitrariamente o número de produções, apenas criando-as nas zonas onde são detectadas lacunas gramaticais.

Para ilustrar a aplicação do mecanismo de interpolação gramatical, considerem-se os anteriores alfabetos de 60 símbolos e outro conjunto de treino, agora com metade da dimensão do representado na Figura 5.9. Após aplicação do mecanismo de inferência gramatical os resultados de modelização, utilizando a gramática obtida sobre o conjunto de teste, encontram-se na Figura 5.12.



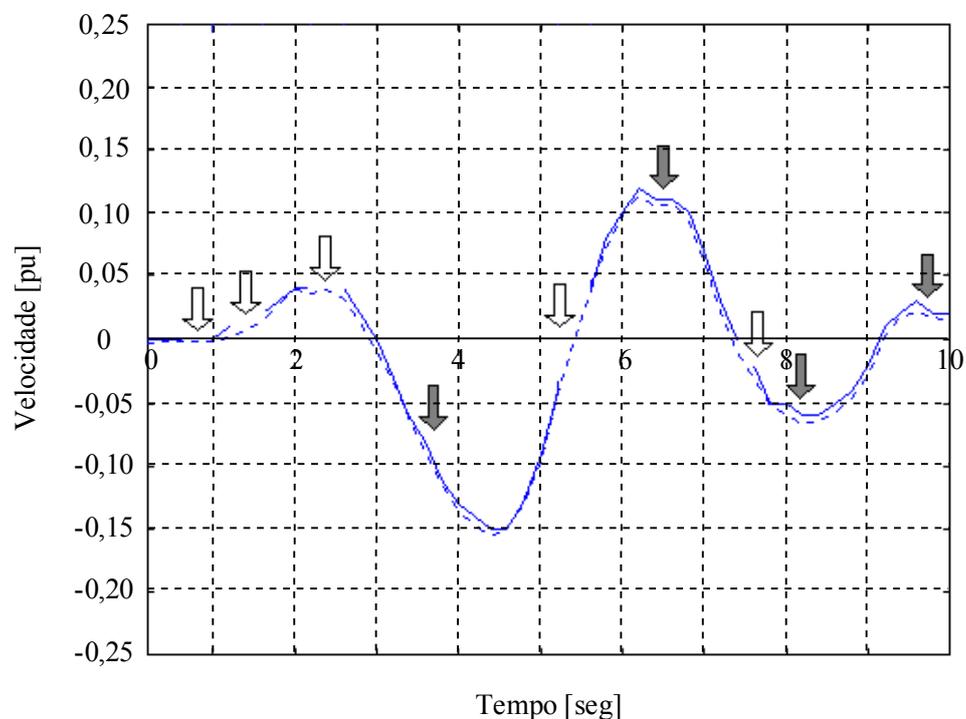
**Figura 5.12. Resultados de modelização a partir de um conjunto de treino pobre.**

Pode verificar-se que o processo de reconstrução apresenta mais falhas que o apresentado na Figura 5.10, isto porque, para um conjunto de treino mais pobre, o número de produções obtidas é consideravelmente menor, originando um maior número de lacunas gramaticais. Um menor conjunto de treino origina uma gramática inferida

que não representa na totalidade a linguagem associada ao accionamento, originando falhas de reconstrução em algumas palavras.

Contudo, como os símbolos definidos para os alfabetos resultam da codificação dos valores numéricos assumidos pelas variáveis no seu domínio de funcionamento, é possível definir uma métrica no espaço das palavras. Deste modo, podem ser obtidas novas produções, a partir das inferidas a partir do conjunto de treino pobre, recorrendo ao mecanismo de interpolação gramatical apresentado na secção 3.5.3. A utilização deste mecanismo, por aplicação da fórmula ( 3.35 ), vai permitir preencher algumas das lacunas gramaticais existentes.

Na Figura 5.13 apresentam-se os resultados de reconstrução para a gramática anterior, após a aplicação do mecanismo de interpolação gramatical. O efeito das produções inexistentes é atenuado, e o número de palavras não reconstruídas é menor. As setas cinzentas referem-se às situações onde ocorreu com sucesso a interpolação de produções inexistentes, e as setas brancas às situações onde o processo de interpolação ainda não conseguiu obter novas produções

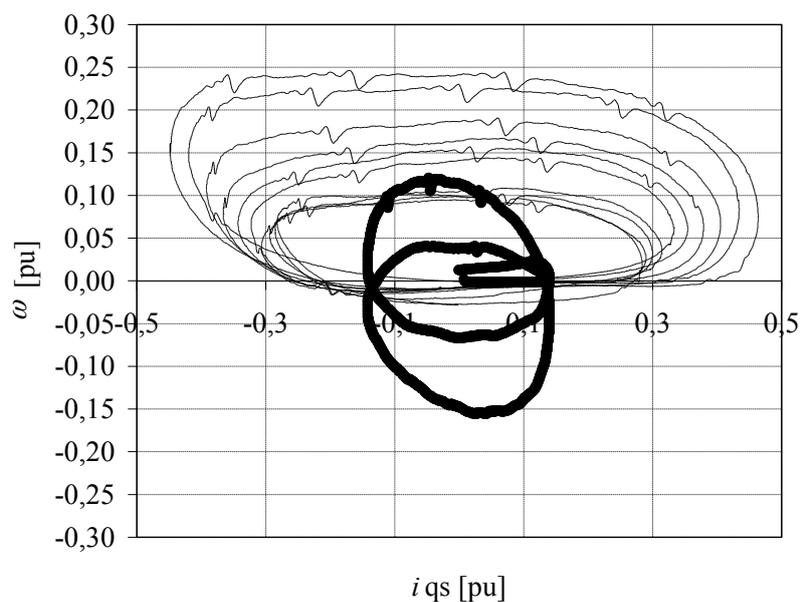


**Figura 5.13. Resultados de modelização após aplicação do mecanismo de interpolação gramatical.**

### 5.2.4.2 Generalização

Por generalização entende-se a capacidade de um qualquer algoritmo providenciar um bom processo de reconstrução, mesmo perante condições de trabalho distintas das observadas durante a fase de treino. O mecanismo de interpolação gramatical providencia maiores capacidades de generalização ao algoritmo, aparte das que ele intrinsecamente possui. No entanto, este tipo de generalização tem apenas um efeito local no domínio de funcionamento do accionamento electromecânico.

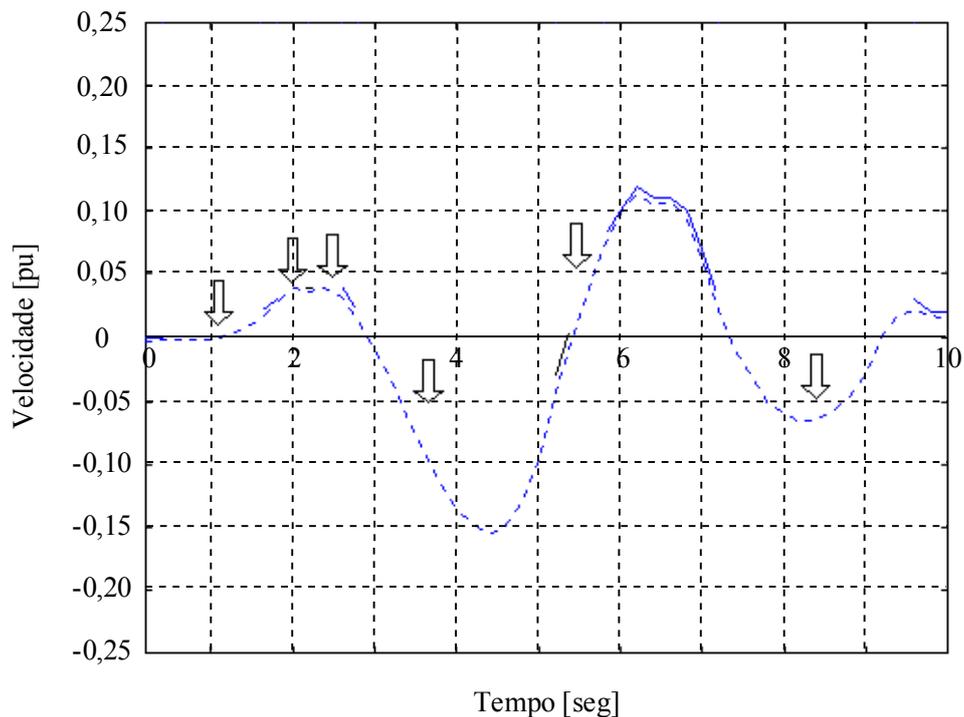
Para testar a capacidade de generalização, aplicou-se o algoritmo de inferência gramatical a um conjunto de treino limitado, contendo apenas palavras referentes a valores de velocidade positivos, apresentado na Figura 5.14, e testou-se a gramática inferida no mesmo conjunto de teste considerado anteriormente. Também aqui o conjunto de teste é apresentado a traço carregado.



**Figura 5.14. Conjuntos de treino limitado e de teste para o accionamento electromecânico.**

Os resultados de reconstrução encontram-se apresentados na Figura 5.15. Neste caso também a velocidade do accionamento electromecânico é representada por uma linha de traço interrompido e a resposta da gramática por uma linha contínua. Pode verificar-se

que o processo falha essencialmente para as zonas do domínio de funcionamento não abrangidas pelo conjunto de treino. Em particular, os símbolos que codificam velocidades negativas do accionamento electromecânico não são reconstruídos, evidenciando o carácter local da generalização. É assim extremamente importante que os dados experimentais utilizados como conjunto de treino sejam constituídos a partir de uma amostra significativa de todo o domínio de funcionamento do accionamento electromecânico.



**Figura 5.15. Generalização utilizando o mecanismo de interpolação gramatical.**

Se o conjunto de treino se limitar a exemplos pertencentes apenas a um subconjunto restrito do domínio de funcionamento, não existe garantia de que o algoritmo extraia uma relação funcional com boas características de generalização, obtendo-se uma relação funcional local válida apenas na zona do domínio à qual pertencem as palavras exemplo utilizadas na aprendizagem. A gramática inferida representará apenas um subconjunto da totalidade da linguagem, sendo assim importante manter o processo de aprendizagem contínuo.

Será ainda importante referir que o mecanismo de interpolação gramatical funciona como uma forma de generalização, permitindo a reconstrução mesmo em zonas do

domínio não percorridas na fase de treino. No entanto, esta forma de generalização apresenta também carácter local, pois o mecanismo de interpolação gramatical apenas é eficiente em torno de um vizinhança das produções inferidas a partir dos exemplos de treino.

## 5.3 DETECÇÃO DE ANOMALIAS

Qualquer gramática  $G$  codifica uma classe de padrões que pertencem à linguagem  $L(G)$  gerada por essa gramática. Deste modo, é possível utilizar a gramática para reconhecer palavras bem formadas e rejeitar as anómalas. Sempre que a gramática tenha sido inferida a partir das condições de funcionamento normal de um qualquer sistema dinâmico, se num instante temporal posterior este estiver sujeito a alguma anomalia, as palavras provenientes da sua evolução dinâmica não serão compatíveis com a gramática inferida. É razoável assumir que este facto pode ocorrer mesmo numa fase inicial da avaria, apresentando-se o algoritmo linguístico como uma boa ferramenta para uma atempada detecção de anomalias.

Sem pretensões de exaustão, pode afirmar-se que a detecção de anomalias consiste em simplesmente efectuar uma decisão binária. No entanto, existem vários factores que contribuem para que a tomada desta decisão, antes da avaria se tornar irreversível, seja extremamente difícil em sistemas reais, como sejam a incerteza no conhecimento da própria dinâmica do sistema, perturbações desconhecidas ou observações imprecisas. Deste modo torna-se importante obter o máximo de informação possível acerca do sistema, quer seja sob a forma de modelo matemático, descrição qualitativa ou obtida através de exemplos, por forma a atingir um sistema de detecção de anomalias de confiança e robusto. A noção de robustez é entendida no sentido de efectuar uma detecção de confiança num estágio inicial da avaria, evitando consequências mais gravosas.

Uma grande variedade de métodos de detecção de anomalias tem sido proposto e estudado na literatura (Patton, Frank e Clark, 2000), sendo possível, em função das técnicas usadas, dividir todas essas metodologias em três grupos: baseadas no modelo, no conhecimento ou na análise de sinal. As técnicas baseadas no modelo, cujo princípio de funcionamento se alicerça na ideia de que perante uma anomalia o modelo do sistema deixa de ser válido, podem recorrer a modelos quantitativos do sistema (conjunto de equações diferenciais, descrição em espaço de estados, funções de transferência ou outros), ou então recorrer a técnicas de *soft computing* (métodos de inteligência artificial, redes neuronais, lógica *fuzzy* ou outros). As técnicas baseadas no

conhecimento obrigam a uma percepção das condições de funcionamento do sistema em várias situações anómalas, enquanto que as técnicas baseadas na análise de sinal recorrem, por exemplo, a métodos de análise espectral ou de estimação de parâmetros sem aplicação explícita de modelos.

### 5.3.1 Detecção de anomalias por utilização de gramática

A detecção de anomalias por utilização da gramática que representa a linguagem associada ao sistema dinâmico, pode enquadrar-se quer nas técnicas baseadas na análise de sinal quer nas técnicas baseadas no modelo, sendo que neste último caso não se considera um modelo quantitativo do sistema dinâmico mas sim um obtido a partir da análise de exemplos provenientes do próprio sistema.

As técnicas baseadas no modelo baseiam-se na estimação do estado, saída ou parâmetros associados ao sistema. Em presença de uma anomalia, quer proveniente da alteração de parâmetros do próprio sistema ou de observações anómalas, a evolução da saída vai sofrer alterações. Deste modo, monitorizando a evolução da saída é possível detectar a anomalia, sendo para tal fundamental possuir conhecimento sobre a relação funcional representativa do sistema dinâmico.

Existem algumas técnicas que se baseiam em modelos qualitativos do sistema dinâmico para comparar as respostas do sistema e do seu modelo. Neste caso não é importante uma descrição quantitativa do sistema mas sim da sua evolução qualitativa, o que pode ser atingido com uma descrição gramatical menos ‘pesada’, tal como acontece nos resultados apresentados na Figura 5.11.

O sistema dinâmico pode ser descrito por uma relação funcional do tipo ( 5.36 ), onde  $u$  representa a variável de comando,  $y$  representa a variável de saída e o parâmetro  $\tau$ , não se apresentando necessariamente fixo, depende das relações entre variáveis para cada zona do domínio de funcionamento do sistema.

$$y_{k+1} = g(y_{k-\tau}, \dots, y_k, u_k) \quad ( 5.36 )$$

Por aplicação do algoritmo de inferência gramatical, e dependendo do conjunto de treino considerado, é possível extrair várias gramáticas –  $G_1 \dots G_n$  – que representem a

linguagem  $L$  associada ao sistema dinâmico descrito por ( 5.36 ). Qualquer que seja a gramática inferida, gerando a mesma linguagem todas se dizem equivalentes e passíveis de representar o sistema dinâmico na ausência de anomalias.

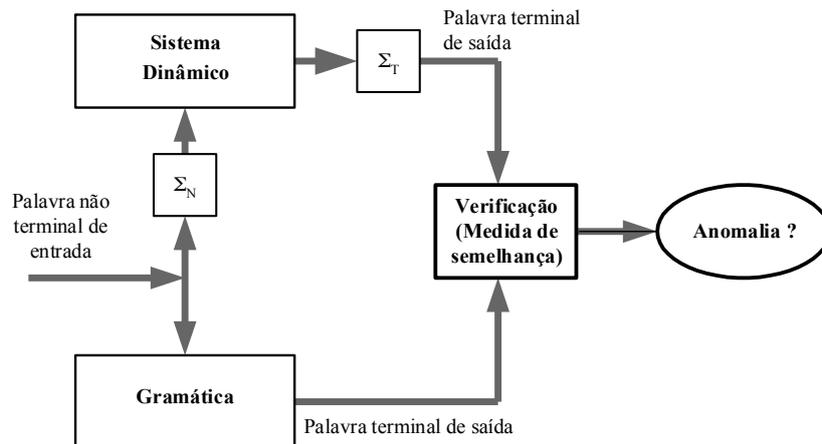
Perante uma anomalia, proveniente de uma alteração de parâmetros do próprio sistema ou de uma observação defeituosa da evolução das variáveis envolvidas, o modelo do sistema altera-se passando a ser descrito, respectivamente, por ( 5.37 ) ou ( 5.38 ). Neste caso o sistema passará a ter associada uma nova linguagem  $\tilde{L}(\tilde{G})$ , descrita por uma nova gramática  $\tilde{G}$ .

$$\tilde{y}_{k+1} = \tilde{g}(\tilde{y}_{k-\tau}, \dots, \tilde{y}_k, u_k) \quad ( 5.37 )$$

$$\tilde{y}_{k+1} = g(\tilde{y}_{k-\tau}, \dots, \tilde{y}_k, \tilde{u}_k) \quad ( 5.38 )$$

Dadas duas palavras distintas, provenientes da observação da variável de saída do sistema dinâmico em causa –  $z$  e  $\tilde{z}$  –, respectivamente numa situação normal e numa situação anómala, a severidade da anomalia será tanto maior quanto maior for a distância  $d(z, \tilde{z})$ , tal como definida em ( 3.32 ), entre ambas as palavras. Esta detecção de anomalias é assim baseada na distância e medida de semelhança já consideradas para o mecanismo de interpolação gramatical. O algoritmo de detecção de anomalias, cujo princípio de funcionamento se encontra representado na Figura 5.16, verifica a distância entre uma palavra proveniente do sistema dinâmico e outra proveniente da gramática inferida a partir do funcionamento normal desse mesmo sistema. Se essa distância for superior a uma distância limiar ( 5.39 ), então é considerada a existência de anomalia.

$$d(z, \tilde{z}) > d_{\text{limiar}} \quad ( 5.39 )$$

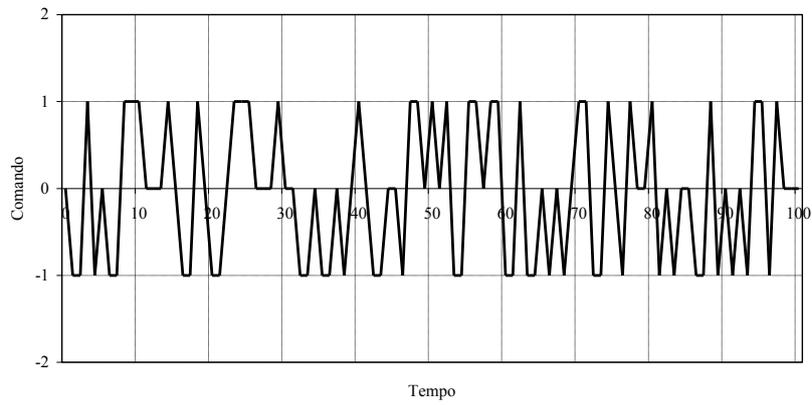


**Figura 5.16. Princípio de funcionamento do mecanismo de detecção de anomalias.**

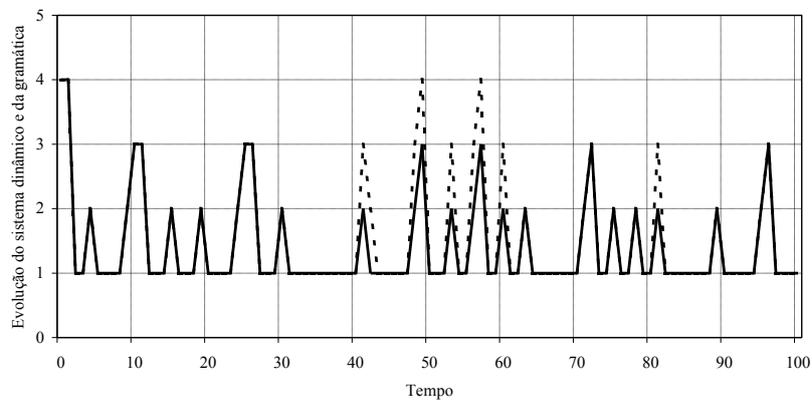
Como exemplo de aplicação considere-se o sistema dinâmico ( 3.21 ), com o mesmo alfabeto e produções inferidas anteriormente na secção 3.7. Para a aplicação do algoritmo de detecção de anomalias considera-se um comando aleatório, durante 100 instantes de tempo. Uma anomalia sustentada foi simulada durante o intervalo de tempo  $k \in [30;60]$ , e uma espúria durante o intervalo  $k \in [80;85]$ . A simulação da anomalia é resultante da simples substituição do parâmetro 2 por 3 no sistema dinâmico ( 3.21 ). O parâmetro do sistema dinâmico varia e, assim, algumas das palavras geradas não coincidem com as provenientes da gramática inferida.

$$\begin{cases} x_{k+1} = x_k - x_{k-1} + 2u_k \\ y_k = x_k \end{cases} \quad \text{com} \quad \begin{cases} x_k \in \{1,2,3,4\} \\ u_k \in \{-1,0,+1\} \end{cases} \quad (3.21)$$

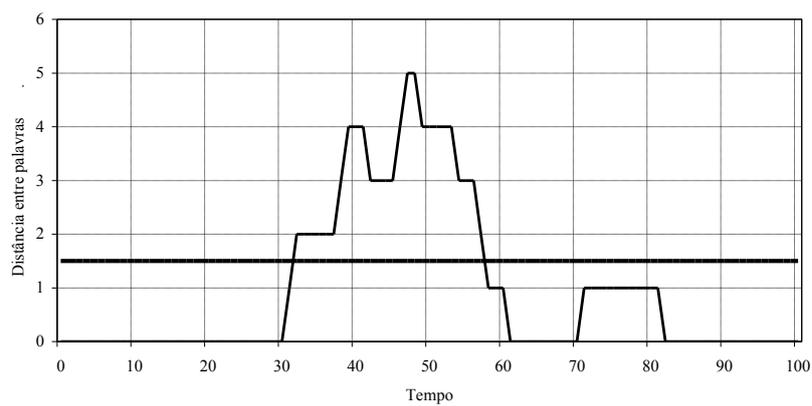
Na Figura 5.17 apresentam-se os resultados da aplicação do algoritmo de detecção de anomalias. Em resposta à evolução da variável de comando representada na Figura 5.17 (a), tanto a gramática inferida como o sistema dinâmico produzem palavras terminais de saída. Estas apresentam-se na Figura 5.17 (b), onde a evolução das palavras provenientes da gramática é representada por uma linha contínua e a do sistema dinâmico por uma linha a traço interrompido. Pode verificar-se que em presença de anomalias alguns dos símbolos não coincidem. Na Figura 5.17 (c) é apresentada a distância entre as palavras produzidas pela gramática e as provenientes da codificação da variável de saída do sistema dinâmico, onde se pode verificar que a distância somente excede a distância limiar para anomalias sustentadas, sendo que para o caso da anomalia espúria não é reportada avaria.



(a)



(b)



(c)

**Figura 5.17. Aplicação do algoritmo de detecção de anomalias ao sistema dinâmico exemplo ( 3.49 ). (a) Comando; (b) Evolução da saída do sistema e da gramática; (c) Distância entre palavras.**

### 5.3.2 Detecção de anomalias na máquina assíncrona por análise da corrente estatórica

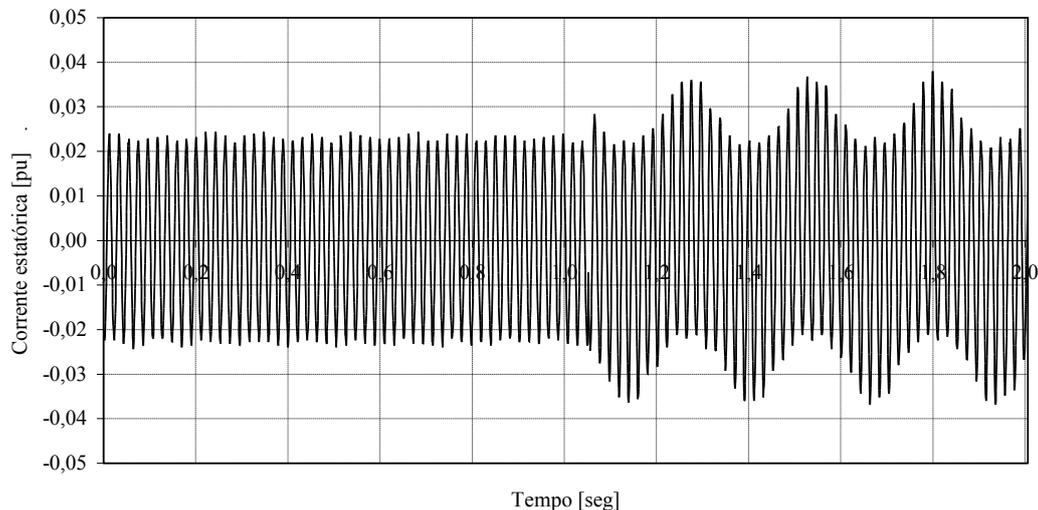
Desde o início da década de 80 (Thomason e Rankin, 1987) que a análise da corrente estatórica, devido às características não invasivas desta análise, tem sido um dos parâmetros mais utilizados para detectar falhas no rotor de máquinas assíncronas de rotor em curto circuito.

Nos últimos anos, vários testes têm sido sugeridos para detectar barras de rotor partidas. Estes testes baseiam-se na análise de diferentes tipos de grandezas, como sejam a monitorização de vibrações (Leonard e Thomson, 1986), o teste por descargas parciais (Stone e Sedding, 1995), a análise do fluxo axial (Penman, Sedding, Lloyd e Fink, 1994), a utilização da potência instantânea (Maier, 1992) ou do binário (Hsu, 1995). Paralelamente a estas técnicas têm evoluído também aquelas que se baseiam na monitorização da corrente estatórica (Stavrou, Sedding e Penman, 1999) (Cruz e Cardoso, 2000) (Bellini, Filippetti, Franceschini, Tassonnie e Kliman, 2000).

Uma assimetria eléctrica ou magnética no rotor de uma máquina assíncrona de indução provoca o aparecimento de componentes adicionais de frequência  $(1-2s)\omega_s$  – onde  $s$  representa o escorregamento e  $\omega_s$  a frequência fundamental das correntes estatóricas – no espectro das correntes que percorrem o estator da máquina. A existência de barras quebradas (variação da resistência do rotor) traduz-se numa assimetria eléctrica com a consequente modelação da corrente estatórica (Kovacs, 1984). Será importante referir que defeitos como as oscilações mecânicas também provocam modulação nas correntes estatóricas.

Como visto anteriormente, na secção 3.6, a utilização do mecanismo de inferência gramatical possibilita a identificação de formas de onda – padrões – características de um dado sistema dinâmico. Neste contexto, a utilização da gramática para detecção de anomalias enquadra-se no conjunto de técnicas designadas como baseadas na análise de sinal. Apresenta-se de seguida um pequeno exemplo de detecção de anomalias numa máquina assíncrona onde a variável codificada como alfabeto terminal é a corrente estatórica. A gramática é inferida a partir do normal funcionamento da máquina, sendo provocada uma anomalia por alteração brusca da resistência de uma das fases do rotor.

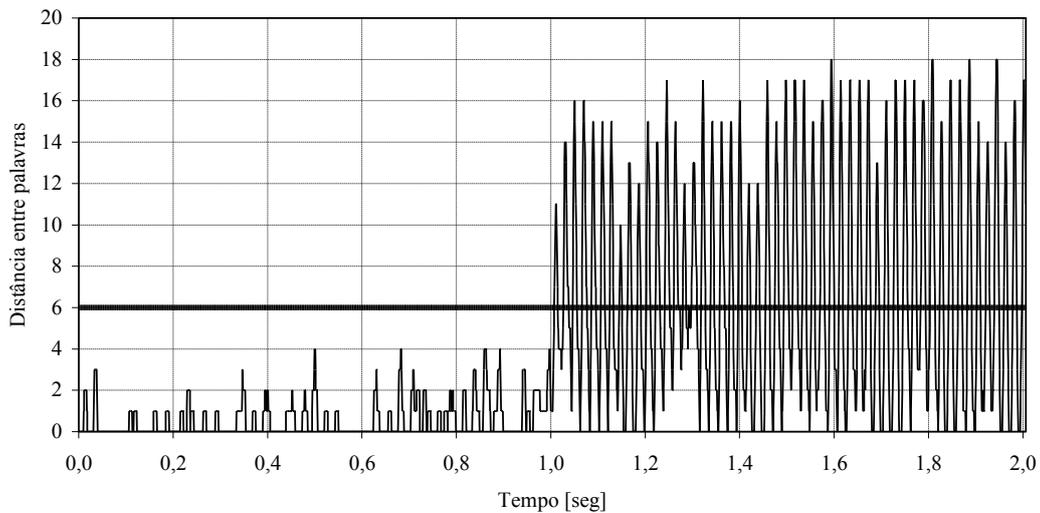
Na Figura 5.18 apresenta-se a evolução da corrente estatórica, onde se provocou a anomalia após o primeiro segundo.



**Figura 5.18. Evolução da corrente estatórica.**

Após a alteração da resistência do rotor, e devido ao aparecimento de componentes de frequência diferente da fundamental na corrente estatórica, o padrão destas mesmas correntes altera-se. Deste modo a gramática, cuja linguagem se refere à codificação de uma corrente estatórica cujos padrões de funcionamento são determinados essencialmente pela frequência fundamental (ausência de anomalia), apresenta-se incapaz de representar os diferentes padrões da nova linguagem.

O algoritmo de detecção de anomalias vai assim detectar a existência de padrões anómalos, e analisar a distância entre as palavras provenientes da codificação da corrente estatórica e os padrões descritos pela gramática inferida a partir do normal funcionamento da máquina. Verifica-se que esta distância, apresentada na Figura 5.19, somente ultrapassa o limiar estabelecido após o aparecimento da anomalia. Este limiar não nulo é estabelecido para evitar que as pequenas diferenças nas linguagens, provenientes da gramática inferida e da máquina, devidas à presença de ruído e a pequenas falhas nas produções inferidas, provoquem o assinalar de alguma anomalia inexistente.



**Figura 5.19. Distância entre palavras.**

Os exemplos anteriores evidenciam as capacidades de utilização da inferência gramatical em sistemas de detecção de anomalias. No primeiro exemplo a detecção é basicamente efectuada seguindo o paradigma das técnicas de detecção baseadas em modelos. No segundo, e fazendo uso das capacidades das gramáticas inferrem tanto uma relação funcional entrada/saída como um determinado padrão de sinal, é considerada a detecção de um tipo particular de anomalia recorrendo à análise de sinais típicos de funcionamento do sistema dinâmico.

## 5.4 CONTROLO

Nos sistemas de controlo clássico, o conhecimento dos sistemas a serem controlados, é apresentado sob a forma de um conjunto de equações diferenciais algébricas, que relacionam analiticamente os comandos, os estados e as saídas do sistema. Contudo, como já referido anteriormente, essas representações apresentam-se geralmente complexas, baseando-se em muitas hipóteses simplificativas, e contendo vários parâmetros que, além da sua difícil medida, apresentam variações ao longo do tempo que dificultam a sua modelização. A teoria de controlo clássico apresenta ainda outras limitações como sejam a consideração da linearidade e invariância no tempo dos sistemas a serem controlados. Por forma a superar estas deficiências, ao longo dos últimos anos, técnicas de *soft computing* têm sido cada vez mais utilizadas em vários tipos de sistemas, nomeadamente nos sistemas de accionamento electromecânico. Entre as suas várias vantagens podem salientar-se a não necessidade de existência de um modelo matemático, a melhoria do desempenho, a capacidade de rejeição do ruído, a capacidade de aprendizagem e adaptação evolutiva, a capacidade de generalização, e a sua capacidade de controlo sobre sistemas não lineares. Na realidade, os sistemas físicos apresentam-se na sua generalidade como não-lineares, sendo que a consideração da sua linearidade advém de maiores ou menores simplificações que são efectuadas aquando do seu estudo. Quando se projectam controladores não-lineares, e uma vez que o tratamento de sistemas não-lineares é matematicamente difícil, é prático considerar as aproximações que levam a um sistema linear e depois estender esse estudo ao caso não-linear.

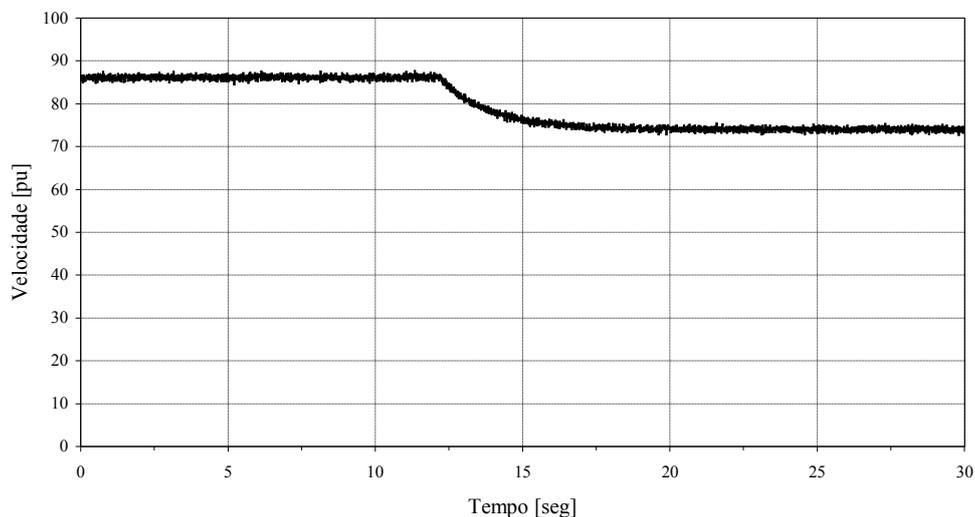
### 5.4.1 Sistema de accionamento electromecânico

Nesta secção descreve-se a aplicação das técnicas de linguagem formal ao controlo de um accionamento electromecânico, baseado numa máquina assíncrona com um comando do tipo V/f.

Em condições de pequeno escorregamento, normal para o funcionamento habitual da maior parte destas máquinas, o comando do tipo V/f possibilita a imposição da velocidade através da frequência imposta às correntes estatóricas. As dificuldades

associadas a este método surgem para baixas velocidades, pois a queda de tensão resistiva torna-se predominante, ou para situações onde ocorre um aumento da resistência do rotor. A natureza fortemente oscilatória do sistema impede uma boa compensação em cadeia fechada, por métodos convencionais, caso se pretenda grande rapidez de resposta (Palma, 1999).

A variação da constante de tempo rotórica, nomeadamente devido à variação do valor da resistência dos enrolamentos do rotor, afecta o normal funcionamento de um comando do tipo V/f, aumentando o escorregamento. Na Figura 5.20 pode observar-se a diminuição de velocidade verificada quando, mantendo a frequência de alimentação constante e igual a 50 Hz, ocorre um aumento da resistência do rotor da máquina.

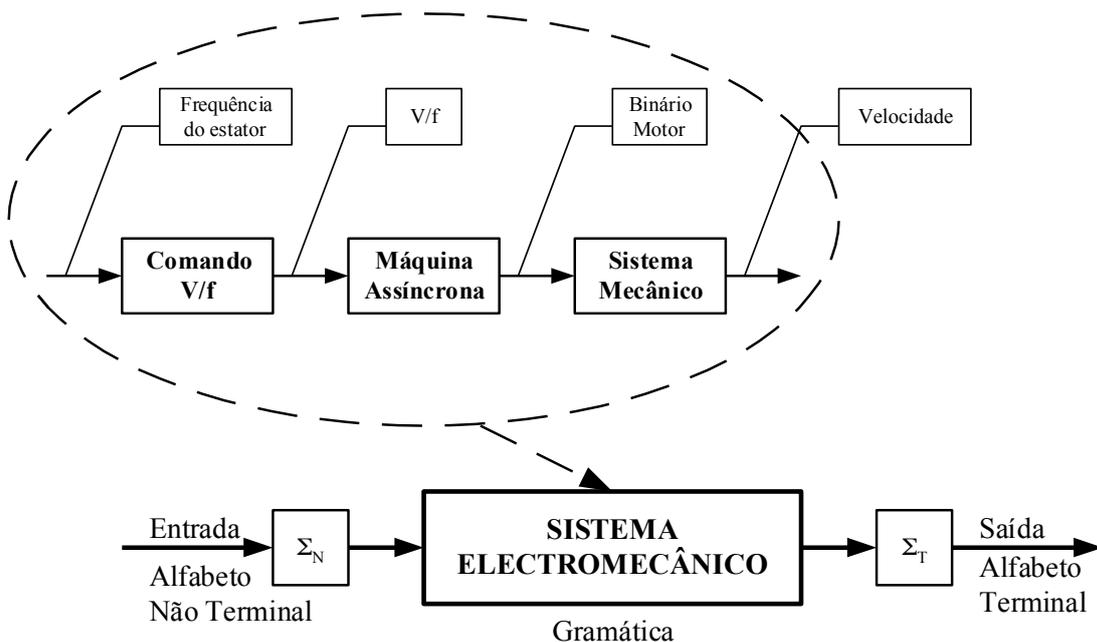


**Figura 5.20. Diminuição da velocidade de um accionamento com comando do tipo V/f devida ao aumento da resistência rotórica.**

Considerando um referencial dq solidário com o vector de tensão do estator, estando a componente d paralela a esse vector e desprezando a dinâmica do conversor electrónico de potência, a dinâmica do accionamento é dada por ( 5.40 ). A existência de um comando do tipo V/f, sem a compensação para baixas frequências das grandezas do estator, implica que o valor da tensão estatórica da máquina varia linearmente com a frequência imposta às correntes  $u = k\omega_s$ .

$$\left\{ \begin{aligned} \frac{di_{ds}}{dt} &= -\left(\frac{1}{\sigma\tau_s} + \frac{1-\sigma}{\sigma\tau_r}\right)i_{ds} + \omega_s i_{qs} + \frac{1-\sigma}{\sigma\tau_r M} \psi_{dr} + \frac{1-\sigma}{\sigma M} \omega \psi_{qr} + \frac{1}{\sigma L_s} k \omega_s \\ \frac{di_{qs}}{dt} &= -\omega_s i_{ds} - \left(\frac{1}{\sigma\tau_s} + \frac{1-\sigma}{\sigma\tau_r}\right)i_{qs} + \frac{1-\sigma}{\sigma M} \omega \psi_{dr} + \frac{1-\sigma}{\sigma\tau_r M} \psi_{qr} \\ \frac{d\psi_{dr}}{dt} &= -\frac{1}{\tau_r} \psi_{dr} + (\omega_s - \omega) \psi_{qr} + \frac{M}{\tau_r} i_{ds} \\ \frac{d\psi_{qr}}{dt} &= -(\omega_s - \omega) \psi_{dr} - \frac{1}{\tau_r} \psi_{qr} + \frac{M}{\tau_r} i_{qs} \\ \frac{d\omega}{dt} &= \frac{M}{JL_r} (-i_{ds} \psi_{qr} + i_{qs} \psi_{dr}) - \frac{B}{J} \omega - \frac{1}{J} T_{ext} \end{aligned} \right. \quad (5.40)$$

O sistema electromecânico considerado, isto é, aquele que vai gerar a linguagem cuja gramática se pretende inferir, é constituído pela associação entre a carga mecânica, a máquina assíncrona, o comando de velocidade V/f e os respectivos sensores; tal como se apresenta na Figura 5.21.

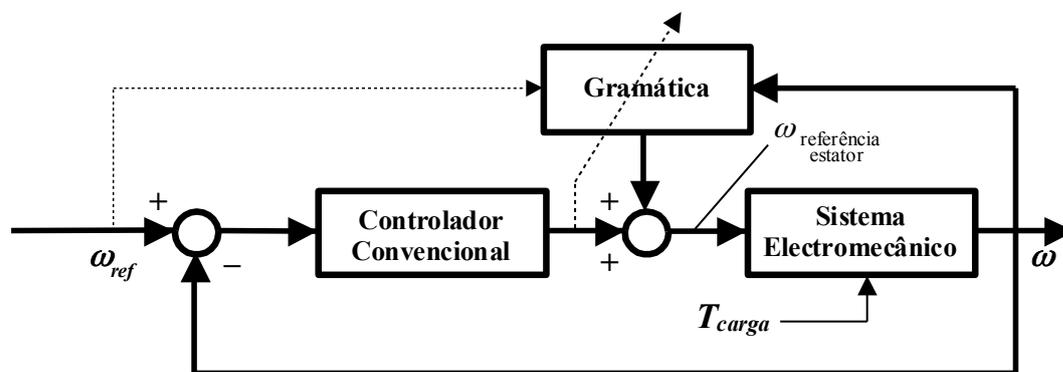


**Figura 5.21. Sistema de accionamento electromecânico e respectiva gramática.**

À semelhança do procedimento seguido aquando da modelização do accionamento electromecânico, optou-se por associar o alfabeto terminal à velocidade mecânica do accionamento (variável de saída do sistema electromecânico) e o alfabeto não terminal à referência fornecida a esse mesmo sistema (variável de entrada).

## 5.4.2A Gramática como regulador

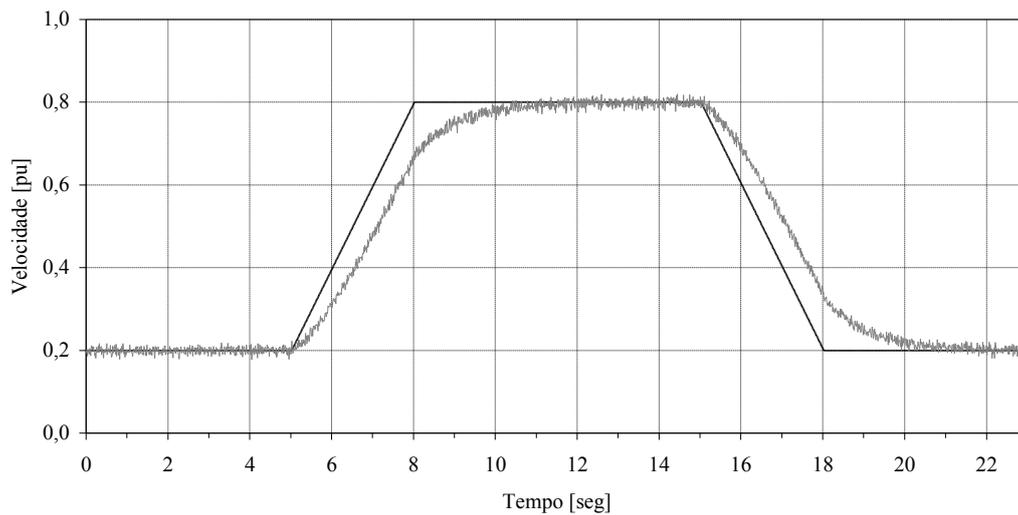
A gramática extraída a partir do anterior sistema de accionamento electromecânico vai ser utilizada como regulador numa topologia de compensação indirecta, representada na Figura 5.22. Este tipo de compensação não se encontra sujeita às oscilações presentes nas topologias de compensação directa, independentemente do tipo de algoritmo considerado, quer sejam baseados em redes neuronais (Katic e Vukobratovic, 1995), quer utilizem a decomposição bi-ortogonal (Maia, 1995) ou lógica *fuzzy* (Branco, 1998). Neste tipo de compensação (Miyamoto, Kawato, Setoyama e Suzuki, 1988) obtém-se um comportamento mais estável associando o regulador gramatical a um controlador convencional do tipo PI.



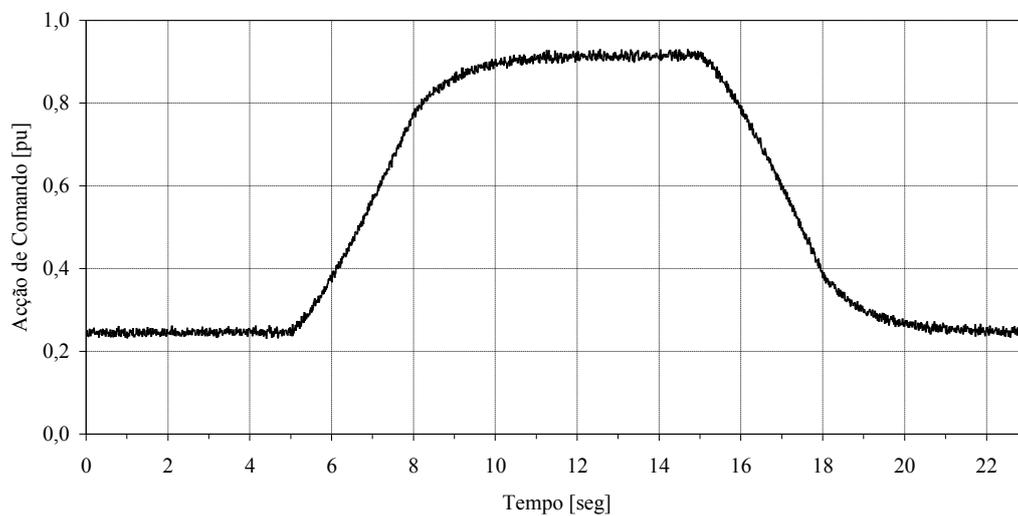
**Figura 5.22. Compensação Indirecta.**

O controlador convencional é aqui utilizado para assegurar um deslocamento significativo através de todo o domínio de funcionamento durante a fase de aprendizagem, onde a gramática infere a relação entre o comando e a saída do accionamento electromecânico. Quando a descrição gramatical do sistema é perfeita a saída do controlador convencional é nula, passando o sistema global a funcionar como se este não existisse. Perante uma alteração das condições de funcionamento o controlador convencional vai actuar no sentido de atenuar a diferença entre o valor da saída e o valor desejado, até que a gramática estabeleça novas produções (ou corrija as existentes) que se adaptem às novas condições de funcionamento.

Inicialmente desencadeia-se a fase de aprendizagem onde apenas se considera a existência do controlador convencional, cujo fraco desempenho se apresenta na Figura 5.23. Nesta figura a referência encontra-se representada a traço contínuo preto e a evolução da velocidade do accionamento a cinzento.



(a)

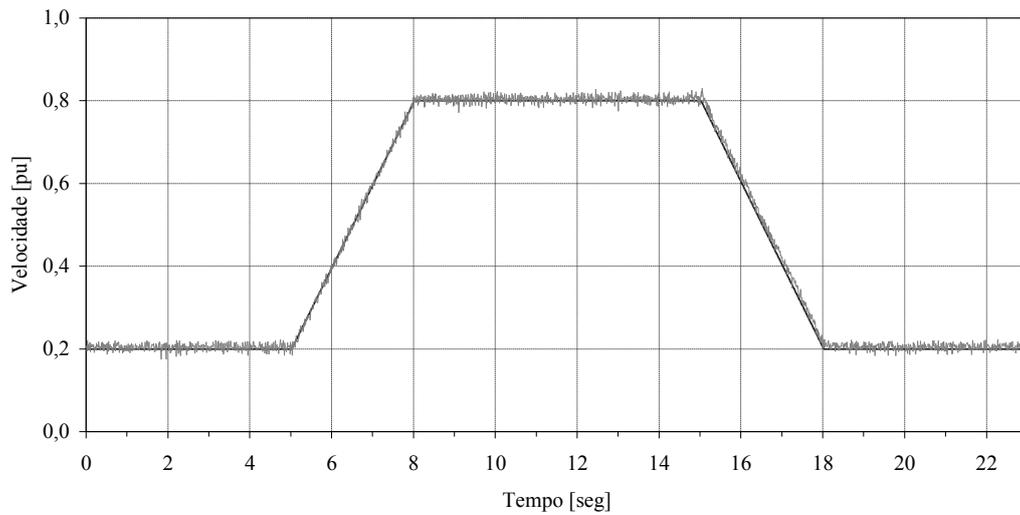


(b)

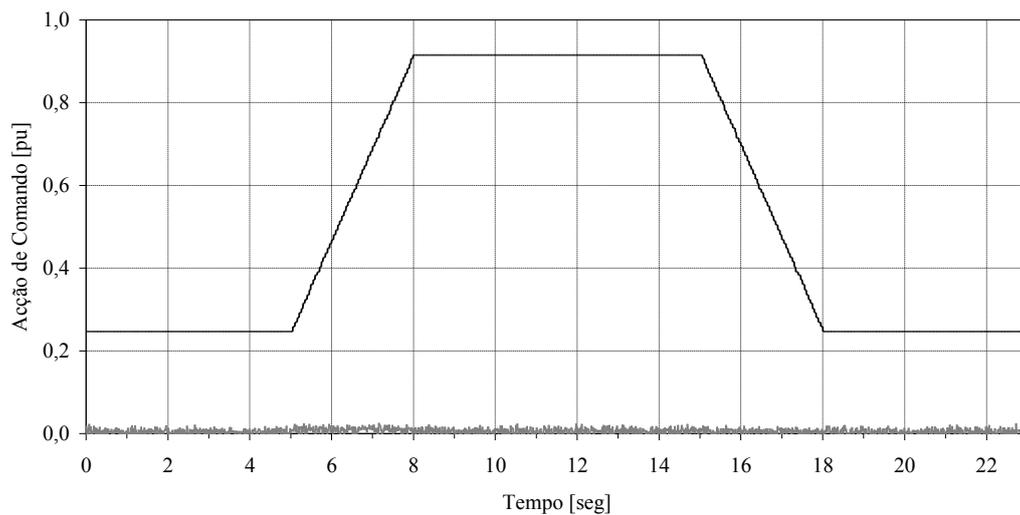
**Figura 5.23. Controlo de velocidade utilizando apenas o compensador convencional: (a) Evolução da velocidade e sua referência; (b) Evolução da acção de comando do controlador convencional.**

Será importante salientar não existir necessidade de ajustar os parâmetros deste controlador para um elevado desempenho. Tal tarefa, além de se afigurar complicada para todo o domínio de funcionamento do sistema, é desnecessária pois, nesta fase inicial, as suas funções resumem-se a providenciar que o accionamento evolua por todo o seu domínio de funcionamento por forma a recolher as palavras exemplo necessárias a que o algoritmo de inferência gramatical extraia as produções do sistema de rescrita da gramática. O facto de os desempenhos não serem elevados não é importante porque a gramática não vai emular o controlador convencional, qualquer que ele seja, mas sim extrair a relação funcional entre o comando e a saída do sistema de accionamento electromecânico. Daí que a principal preocupação, na fase de aprendizagem, seja um extenso 'varrimento' do domínio de funcionamento por forma a que as palavras exemplo recolhidas sejam o mais 'ricas' possível.

Após aprendizagem da relação funcional a gramática encontra-se em condições de ser uma boa representação da linguagem do sistema de accionamento electromecânico. Na Figura 5.24 apresenta-se a evolução da velocidade do accionamento electromecânico após o processo de inferência das produções. Nesta figura a acção de comando proveniente da gramática encontra-se representada a traço contínuo preto e a proveniente do controlador convencional a cinzento. Pode verificar-se que a gramática é uma boa representação do accionamento, apresentando-se a saída do controlador convencional praticamente nula.



(a)



(b)

**Figura 5.24. Controlo de velocidade: (a) Evolução da velocidade e sua referência; (b) Evolução das acções de comando dos controladores convencional e gramatical.**

Os bons resultados obtidos, com o controlador gramatical, essencialmente através da inferência de *produções de tipo 1*, evidenciam um comportamento dinâmico predominante de primeira ordem para o sistema electromecânico ( 5.40 ). Tratando-se de um sistema não linear, uma técnica possível de análise é a sua linearização em torno de um ponto de funcionamento (Leigh, 1983). O objectivo será o de substituir o sistema não linear por uma aproximação linear do mesmo, em torno de um determinado ponto

de funcionamento. O uso desta técnica possibilita a utilização de ferramentas de análise lineares para o estudo do sistema não linear.

Considere-se que a solução da equação não linear invariante no tempo ( 5.41 ) – onde, por simplicidade, se omitiu o parâmetro  $t$  – para um determinado comando  $\tilde{\mathbf{u}}(t)$ , partindo de um estado inicial  $\tilde{\mathbf{x}}_0$ , corresponde a uma solução  $\tilde{\mathbf{x}}(t)$  e respectiva saída  $\tilde{\mathbf{y}}(t)$ .

$$\begin{cases} \dot{\mathbf{x}} = \Phi(\mathbf{x}, \mathbf{u}) \\ \mathbf{y} = \Psi(\mathbf{x}) \end{cases} \quad ( 5.41 )$$

Considerando pequenas variações em torno de um determinado comando e estado inicial ( 5.42 ) é expectável que a solução correspondente seja uma pequena variação em torno do ponto de funcionamento considerado ( 5.43 ). Esta análise é válida se  $\|\mathbf{x}_{0\delta}\|$  e  $\|\mathbf{u}_{\delta}(t)\|$  forem suficientemente pequenos para  $t \geq t_0$ .

$$\begin{cases} \mathbf{u}(t) = \tilde{\mathbf{u}}(t) + \mathbf{u}_{\delta} \\ \mathbf{x}_0 = \tilde{\mathbf{x}}_0 + \mathbf{x}_{0\delta} \end{cases} \quad ( 5.42 )$$

$$\begin{cases} \mathbf{x}(t) = \tilde{\mathbf{x}}(t) + \mathbf{x}_{\delta}(t) \\ \mathbf{y}(t) = \tilde{\mathbf{y}}(t) + \mathbf{y}_{\delta}(t) \end{cases} \quad ( 5.43 )$$

Substituindo ( 5.42 ) e ( 5.43 ) na equação de estado ( 5.41 ), desenvolvendo em série de Taylor e desprezando os termos de ordem superior à primeira, obtém-se um sistema dinâmico linearizado ( 5.44 ).

$$\begin{cases} \dot{\mathbf{x}}_{\delta} = \mathbf{A}\mathbf{x}_{\delta}(t) + \mathbf{B}\mathbf{u}_{\delta}(t) \\ \mathbf{y}_{\delta} = \mathbf{C}\mathbf{x}_{\delta}(t) \end{cases} \quad ( 5.44 )$$

O sistema linear anterior, cujas matrizes descritivas do sistema linearizado são dadas por ( 5.45 ), é obtido admitindo a existência dos respectivos jacobianos, a existência de uma solução para  $t \geq t_0$  e é válido apenas em torno de um determinado ponto de funcionamento, para pequenas variações de sinal.

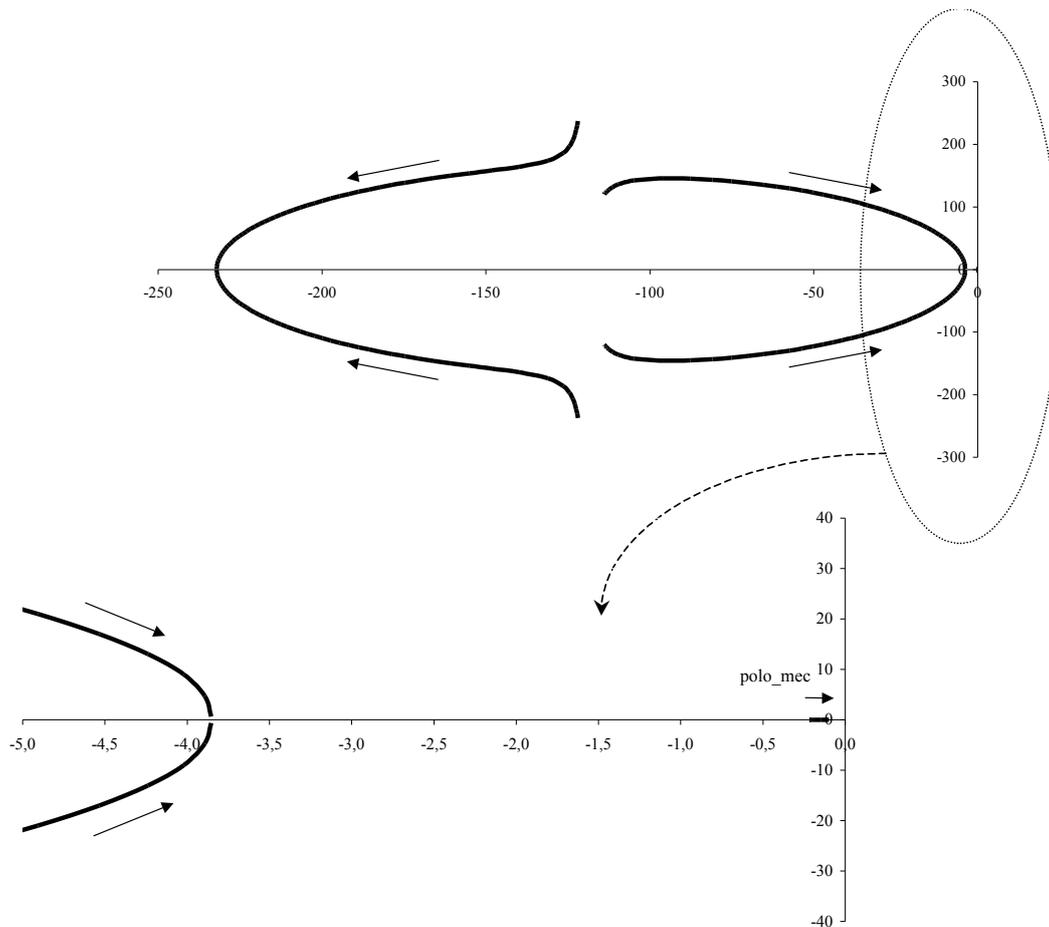
$$\begin{aligned}
 \mathbf{A} &= \frac{\partial \Phi}{\partial \mathbf{x}}(\tilde{\mathbf{x}}, \tilde{\mathbf{u}}) \\
 \mathbf{B} &= \frac{\partial \Phi}{\partial \mathbf{u}}(\tilde{\mathbf{x}}, \tilde{\mathbf{u}}) \\
 \mathbf{C} &= \frac{\partial \Psi}{\partial \mathbf{x}}(\tilde{\mathbf{x}}, \tilde{\mathbf{u}})
 \end{aligned} \tag{5.45}$$

Este procedimento permite a aplicação de conceitos e técnicas de análise e identificação válidas para sistemas lineares ao sistema não linear, em regiões onde este pode ser considerado aproximadamente linear.

Linearizando o sistema ( 5.40 ) em torno de um ponto de funcionamento  $(\tilde{\omega}_s, \tilde{\psi}_{dr}, \tilde{\psi}_{qr}, \tilde{\mathbf{I}}_{ds}, \tilde{\mathbf{I}}_{qs}, \tilde{\omega})$  obtém-se um sistema do tipo ( 5.44 ) com a matriz  $\mathbf{A}$  dada por ( 5.46 ), à qual estão associados quatro pólos electromagnéticos e um mecânico. Relativamente aos pólos electromagnéticos existem dois modos rápidos (associados à dinâmica dos fluxos de dispersão) e dois modos lentos (associados à dinâmica dos fluxos principais).

$$\mathbf{A} = \begin{bmatrix} -\left(\frac{1}{\sigma\tau_s} + \frac{1-\sigma}{\sigma\tau_r}\right) & \tilde{\omega}_s & \frac{1-\sigma}{\sigma\tau_r M} & \frac{1-\sigma}{\sigma M} \tilde{\omega} & \frac{1-\sigma}{\sigma M} \tilde{\psi}_{qr} \\ -\tilde{\omega}_s & -\left(\frac{1}{\sigma\tau_s} + \frac{1-\sigma}{\sigma\tau_r}\right) & -\frac{1-\sigma}{\sigma M} \tilde{\omega} & \frac{1-\sigma}{\sigma\tau_r M} & -\frac{1-\sigma}{\sigma M} \tilde{\psi}_{dr} \\ \frac{M}{\tau_r} & 0 & -\frac{1}{\tau_r} & \tilde{\omega}_s - \tilde{\omega} & \tilde{\psi}_{qr} \\ 0 & \frac{M}{\tau_r} & -(\tilde{\omega}_s - \tilde{\omega}) & -\frac{1}{\tau_r} & \tilde{\psi}_{dr} \\ -\frac{M}{JL_r} \tilde{\psi}_{qr} & \frac{M}{JL_r} \tilde{\psi}_{dr} & \frac{M}{JL_r} \tilde{\mathbf{I}}_{qs} & -\frac{M}{JL_r} \tilde{\mathbf{I}}_{ds} & -\frac{B}{J} \end{bmatrix} \tag{5.46}$$

Na Figura 5.25 apresenta-se a evolução do lugar geométrico dos referidos cinco pólos – valores próprios da matriz ( 5.46 ) – para valores decrescentes da frequência das correntes estatóricas  $\tilde{\omega}_s$ . Pode verificar-se que a dinâmica associada aos quatro pólos electromagnéticos é muito mais rápida que a dinâmica associada ao pólo mecânico (referenciado por ‘polo\_mec’ na figura), apresentando-se este como dominante.



**Figura 5.25. Evolução do lugar geométrico dos valores próprios da matriz ( 5.46 ), para valores decrescentes de  $\tilde{\omega}_s$ .**

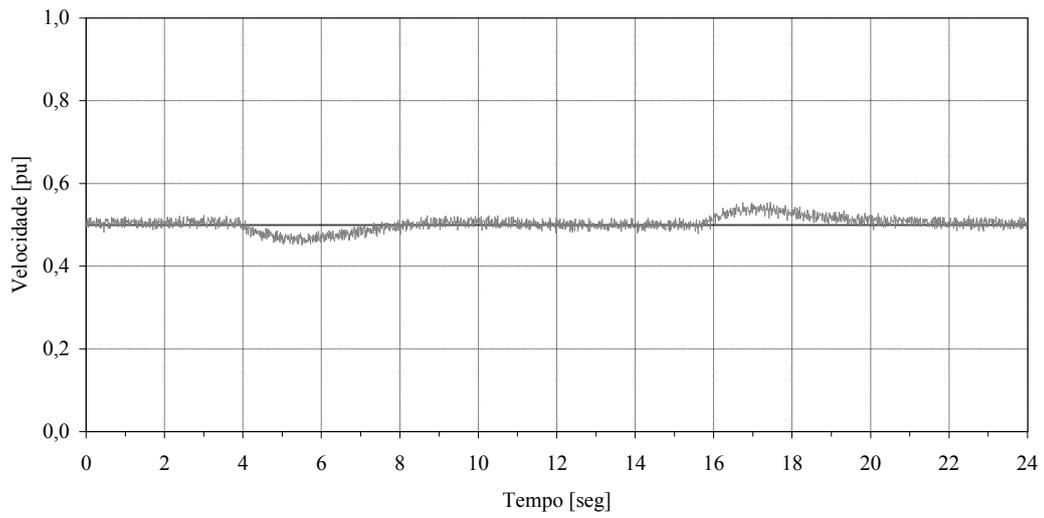
Sendo assim é possível desprezar a dinâmica electromagnética face à mecânica, o que se traduz numa simplificação da relação funcional de todo o sistema, com a conseqüente facilidade de implementação laboratorial, pois o número de variáveis a considerar é mais reduzido. Este comportamento predominante de primeira ordem traduz-se na inferência de uma gramática cujo sistema de rescrita é constituído essencialmente por um conjunto de *produções do tipo 1*.

### **5.4.2.1 Capacidade de adaptação e aprendizagem contínua**

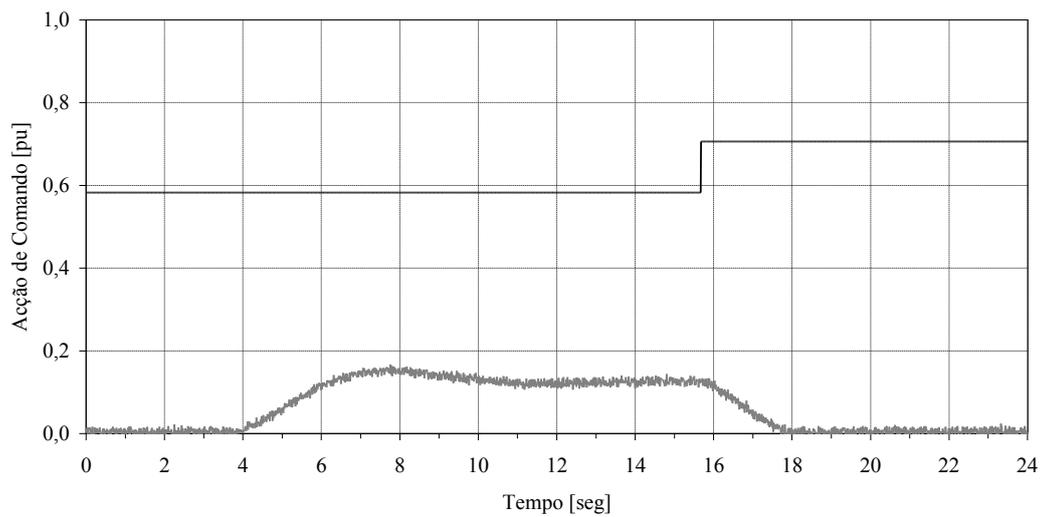
Uma das grandes vantagens da utilização de algoritmos de aprendizagem através de exemplos traduz-se na sua capacidade de adaptação a novas situações de funcionamento dos sistemas a controlar.

Para testar esta adaptabilidade provocou-se uma alteração nas condições de funcionamento do accionamento electromecânico variando o seu binário de carga. O accionamento encontra-se a funcionar com uma velocidade de referência igual a 50% da sua velocidade nominal, sendo que quatro segundos após o início do ensaio este fica sujeito a um novo valor de binário até então desconhecido. Esta alteração das condições de funcionamento provoca uma alteração da linguagem proveniente do accionamento, não representada pela gramática entretanto inferida, como se pode verificar na Figura 5.26. A acção de comando proveniente do controlador gramatical (representada a preto) não satisfaz na totalidade e o controlador convencional (representada a cinzento) tende a compensar essa falha. No entanto, e perante uma acção de comando não nula do controlador convencional o algoritmo de inferência gramatical providencia a extracção de novas produções que se adaptem à nova situação de funcionamento. Após este processo as novas produções são incorporadas no sistema de rescrita da gramática, passando o controlador gramatical a fazer uso das mesmas. Imediatamente a acção de comando do controlador convencional começa a decrescer para zero traduzindo o facto de a nova produção gramatical extraída se adaptar à nova situação.

Será importante referir que mesmo que a referência seja alterada o algoritmo de inferência gramatical pode utilizar a informação entretanto adquirida para produzir novas produções. Estas novas produções serão introduzidas no sistema de rescrita e utilizadas sempre que a situação se repita.



(a)

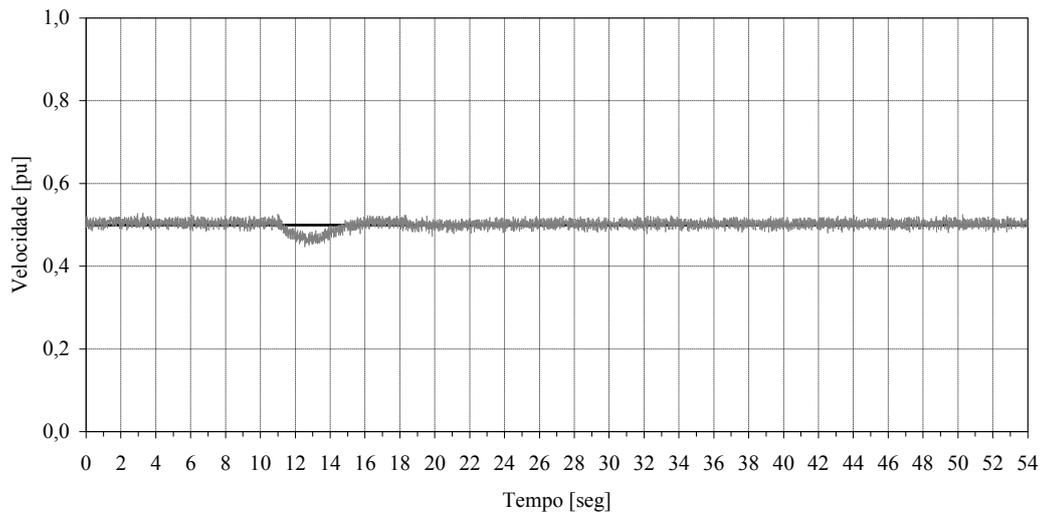


(b)

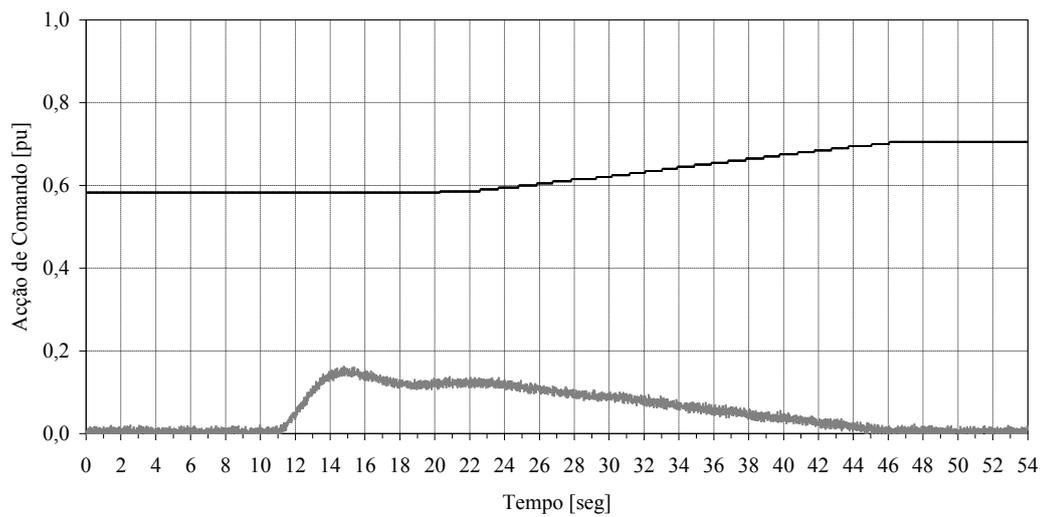
**Figura 5.26. Alteração da carga do accionamento electromecânico – criação de novas produções: (a) Evolução da velocidade e sua referência; (b) Evolução das acções de comando dos controladores convencional e gramatical.**

Uma alternativa ao processo de adaptação anterior consiste, não na criação de novas produções, mas sim na alteração das já existentes. Este procedimento está mais conducente com uma alteração dos próprios parâmetros do sistema e não das condições de funcionamento. O modelo da máquina assíncrona, considerado para o accionamento electromecânico ( 5.40 ), é bastante sensível a variações na constante de tempo rotórica. Tal já foi apresentado na Figura 5.7, pelo que se torna importante que a gramática reaja rapidamente a novas situações paramétricas.

Para testar esta adaptabilidade provocou-se uma alteração da resistência do rotor da máquina assíncrona, alterando assim a sua constante de tempo rotórica. O accionamento encontra-se novamente a funcionar com uma velocidade de referência igual a 50% da sua velocidade nominal, provocando-se essa alteração do valor da resistência do rotor da máquina assíncrona onze segundos após o início do ensaio, como se pode verificar na Figura 5.27. Este facto provoca uma alteração da linguagem proveniente do accionamento, não representada pela gramática entretanto inferida. A acção de comando do controlador gramatical (representada a preto) não satisfaz na totalidade e o controlador convencional (representada a cinzento) tende a compensar essa falha. No entanto, e perante uma acção de comando não nula do controlador convencional, o algoritmo de inferência gramatical providencia a correcção progressiva da produção em causa, no sentido de levar ao decrescimento da acção de comando proveniente do controlador convencional. Findo este processo a acção de comando do controlador convencional anula-se naturalmente traduzindo o facto de que a alteração introduzida na produção se adapta aos novos parâmetros do sistema.



(a)



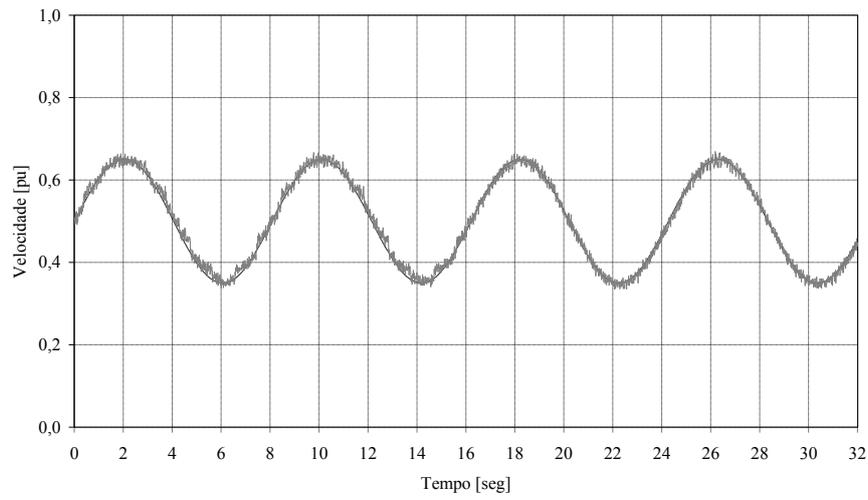
(b)

**Figura 5.27. Variação da resistência do rotor da máquina assíncrona – alteração das produções: (a) Evolução da velocidade e sua referência; (b) Evolução das ações de comando dos controladores convencional e gramatical.**

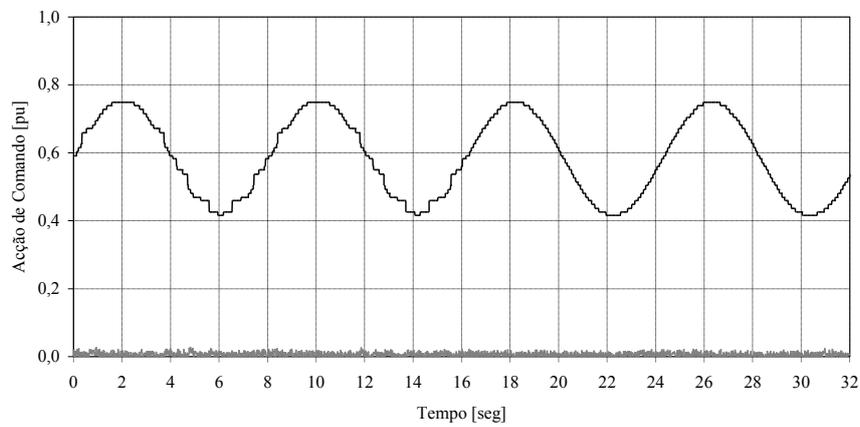
Os mecanismos de adaptação e aprendizagem contínua analisados consistem em utilizar a acção de comando do controlador convencional como informação de que o sistema de rescrita da gramática não se adapta bem ao funcionamento desejado para o accionamento electromecânico. Deste modo podem ser inferidas novas produções, traduzindo a existência de novos pontos de funcionamento no domínio do accionamento, ou corrigidas as existentes, traduzindo a adaptabilidade a novos parâmetros. Em qualquer dos casos o processo de aprendizagem força naturalmente a evolução do controlador convencional para zero, ficando a cadeia de controlo assegurada exclusivamente pelo controlador gramatical.

#### **5.4.2.2 Mecanismo de interpolação gramatical**

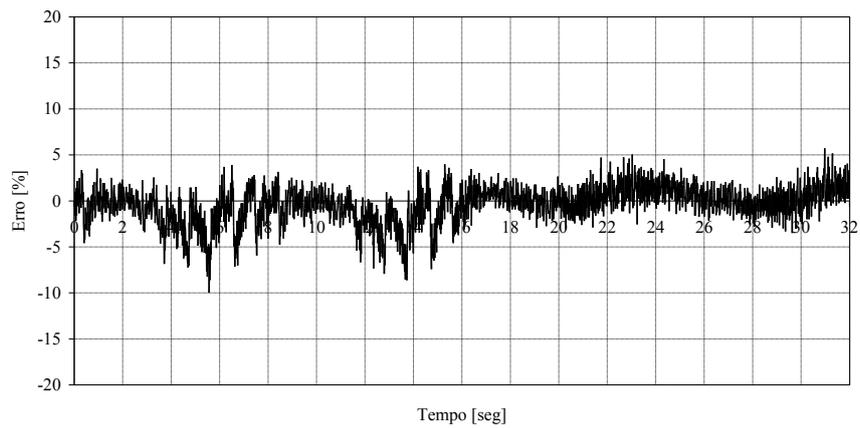
Tal como no processo de modelização o controlo gramatical de accionamentos electromecânicos beneficia da aplicação do mecanismo de interpolação gramatical. O ensaio apresentado na Figura 5.28 pretende ilustrar esta funcionalidade. Neste ensaio considerou-se um processo de aprendizagem mais pobre cuja consequência é uma gramática inferida com um menor número de produções. A primeira metade do ensaio reporta-se a esta situação onde, não considerando a existência do mecanismo de interpolação gramatical, se verifica alguma oscilação nos resultados de velocidade devida à ausência de produções gramaticais essenciais a um melhor desempenho. Na segunda metade do ensaio considerou-se a aplicação do mecanismo de interpolação gramatical, ampliando o anterior conjunto de produções gramaticais com aquelas que se obtêm pela aplicação do referido mecanismo. Observa-se uma melhoria do desempenho, verificando-se uma diminuição da banda de erro absoluto de 5,8 pu para 4,1 pu e do respectivo desvio padrão de 0,96 para 0,67, a partir do momento em que se considera a aplicação do mecanismo de interpolação gramatical.



(a)



(b)



(c)

**Figura 5.28. Interpolação gramatical: (a) Evolução da velocidade e sua referência; (b) Evolução das acções de comando dos controladores convencional e gramatical; (c) Evolução do erro.**

### 5.4.3A Gramática como supervisor

Uma vez inferida uma gramática a partir de um qualquer sistema dinâmico, esta, definindo a estrutura das palavras produzidas pela fonte linguística, apresenta-se como um modelo dessa fonte. Deste modo é possível utilizar as produções para prever a evolução do sistema ou para estabelecer uma sequência de comandos que levam o sistema de um estado para outro. Esta última filosofia de controlo aproxima-se mais de uma perspectiva de controlo de supervisão, predição ou óptimo, onde a gramática terá como objectivo descrever a sequência de comandos em função da evolução que se pretende que o sistema siga, não sendo fundamental o seguimento de uma referência num curto espaço de tempo (função de um regulador).

De seguida descreve-se uma possível implementação desta filosofia de controlo. Considere-se uma palavra, composta por símbolos terminais, que representa a sequência de estados que se pretenda que o sistema siga. Sabendo o estado, ou saída, inicial do sistema, para cada símbolo na palavra terminal objectivo, escolhe-se uma produção que leve o sistema da sua saída presente para a desejada. Em cada instante escolhe-se a produção que, por aplicação do respectivo símbolo não terminal ao comando do sistema dinâmico, o leve para a saída o mais próximo possível da saída desejada. Caso, devido a alguma perturbação do sistema, não seja atingido o valor esperado para a saída, o algoritmo de controlo de supervisão reformula a palavra objectivo em função do objectivo inicialmente estabelecido, reformulando também a geração dos símbolos não terminais, recomeçando deste novo valor da saída. Este procedimento é repetido até que todos os símbolos na palavra objectivo sejam atingidos.

Como exemplo de aplicação considere-se novamente o sistema dinâmico ( 3.21 ).

$$\begin{cases} x_{k+1} = x_k - x_{k-1} + 2u_k \\ y_k = x_k \end{cases} \quad \text{com} \quad \begin{cases} x_k \in \{1,2,3,4\} \\ u_k \in \{-1,0,+1\} \end{cases} \quad ( 3.21 )$$

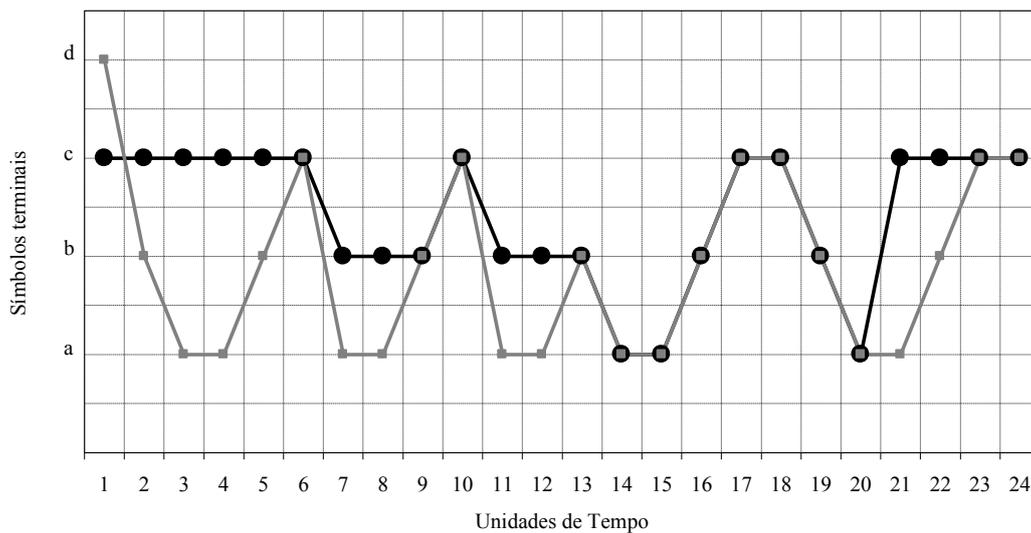
Para a obtenção dos alfabetos terminal e não terminal considere-se a mesma codificação apresentada na Tabela 3.II, e assumo-se como palavra objectivo a sequência de símbolos terminais apresentada em ( 5.47 ).

**Tabela 3.II. Codificação escolhida para o sistema dinâmico ( 3.21 ).**

$y_k \leftrightarrow \sum_T$	$u_k \leftrightarrow \sum_N$
1 ↔ a	-1 ↔ A
2 ↔ b	0 ↔ B
3 ↔ c	+1 ↔ C
4 ↔ d	

$$\{c b c b a a b c c c b a c c\} \quad (5.47)$$

A evolução da sequência de saídas do sistema, para um símbolo inicial ‘d’, é apresentada na Figura 5.29, onde a sequência objectivo é representada a traço contínuo preto e a sequência de símbolos terminais provenientes do sistema dinâmico a cinzento.



**Figura 5.29. Evolução controlada de um sistema dinâmico supervisionado.**

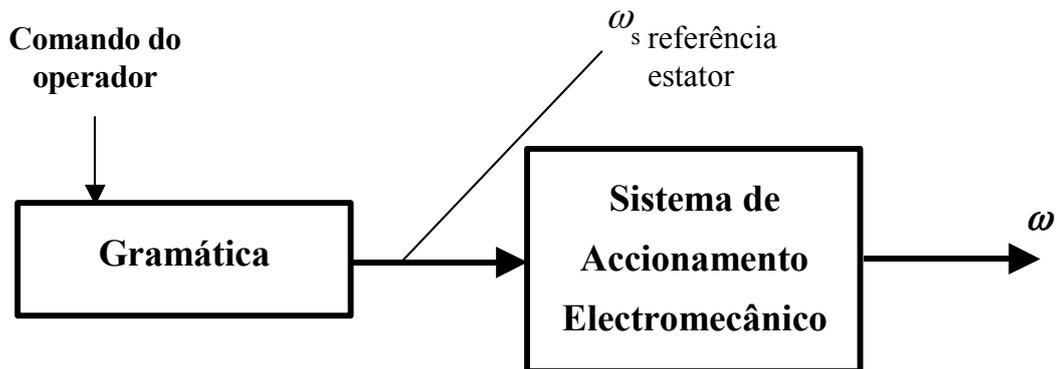
É importante notar que para evoluir de um símbolo para outro o sistema pode ter de passar por outros símbolos terminais. Em função da dinâmica interna e do conjunto de comandos disponíveis para o anterior sistema dinâmico não é possível atingir o símbolo ‘c’ partindo do símbolo ‘a’, isto é, não existe no sistema de rescrita nenhuma produção  $y \text{ e } a U \rightarrow y \text{ e } a c \delta$ , onde  $y$  representa um qualquer símbolo terminal e  $U$  um qualquer símbolo não terminal. Torna-se assim necessário aplicar sucessivamente as produções  $b \text{ e } a A \rightarrow b \text{ e } a a \delta$ ,  $a \text{ e } a C \rightarrow a \text{ e } a b \delta$  e  $a \text{ e } b C \rightarrow a \text{ e } b c \delta$ , quando o objectivo é evoluir do símbolo ‘a’ para o símbolo ‘c’. A palavra desejada não é seguida exactamente, apesar de

todos os seus símbolos serem atingidos, pois tal apresenta-se como fisicamente impossível. Como se pode verificar na Figura 5.29 não foi obtida a palavra ( 5.47 ) mas sim a palavra ( 5.48 ), onde os símbolos que se encontram traçados referem-se à palavra objectivo inicial ( 5.47 ). Todos os símbolos da sequência objectivo são atingidos, só que entre eles são inevitavelmente acrescentados alguns não previstos inicialmente. A palavra objectivo ( 5.47 ) não faz parte da linguagem associada ao sistema dinâmico ( 3.21 ), sendo ( 5.48 ) a melhor aproximação que a gramática consegue impor ao sistema.

$$\{d b a a b e a a b e a a b a a b e e b a a b e e\} \quad ( 5.48 )$$

Uma alternativa ao controlo de supervisão descrito anteriormente é a utilização da linguagem formal para aprendizagem e supervisão de processos industriais onde o comportamento do mesmo é determinado pelo valor assumido por uma entrada de comando. Considerando um qualquer processo industrial, onde a sua evolução seja periódica/cíclica e determinada por uma qualquer entrada de comando (proveniente por exemplo de um operador), é possível utilizar as boas características das gramáticas em representar padrões.

Como exemplo de aplicação, considere-se o anterior sistema electromecânico constituído pela associação entre a carga mecânica, uma máquina assíncrona, um comando do tipo V/f e os respectivos sensores, tal como se apresentou na Figura 5.21. Considere-se um processo industrial onde a evolução desejada para este accionamento electromecânico é determinada por uma entrada de comando externa, o qual se apresenta na Figura 5.30. Conforme o tipo de comando o funcionamento desejado pode ser uma velocidade constante, uma aceleração, uma velocidade de variação sinusoidal, ou outro.



**Figura 5.30. Controlo gramatical supervisionado, em cadeia aberta, de um processo industrial baseado num accionamento com máquina assíncrona.**

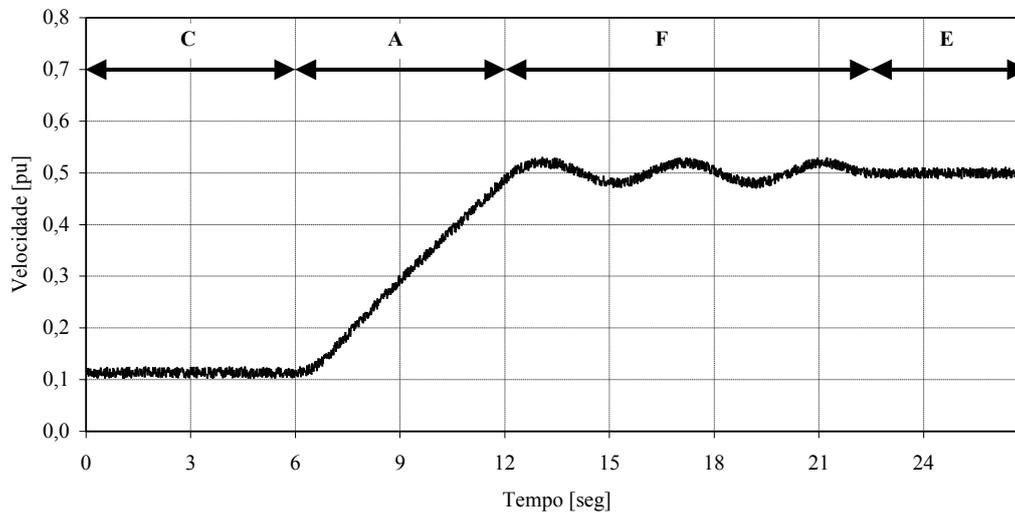
Neste caso optou-se por associar o alfabeto terminal à referência das correntes estatóricas da máquina assíncrona do accionamento electromecânico, e o alfabeto não terminal à entrada de comando externo. O alfabeto terminal será obtido por codificação da variável de comando do accionamento electromecânico e o alfabeto não terminal resultará da codificação da entrada de comando externo, tal como se apresenta na Tabela 5.III.

**Tabela 5.III – Codificação do alfabeto não terminal em função do comando do operador.**

$\Sigma_N$	Comando Externo
A	Aceleração linear
B	Desaceleração linear
C	Velocidade constante baixa
D	Velocidade sinusoidal baixa
E	Velocidade constante média
F	Velocidade sinusoidal média
G	Velocidade constante elevada
H	Velocidade sinusoidal elevada

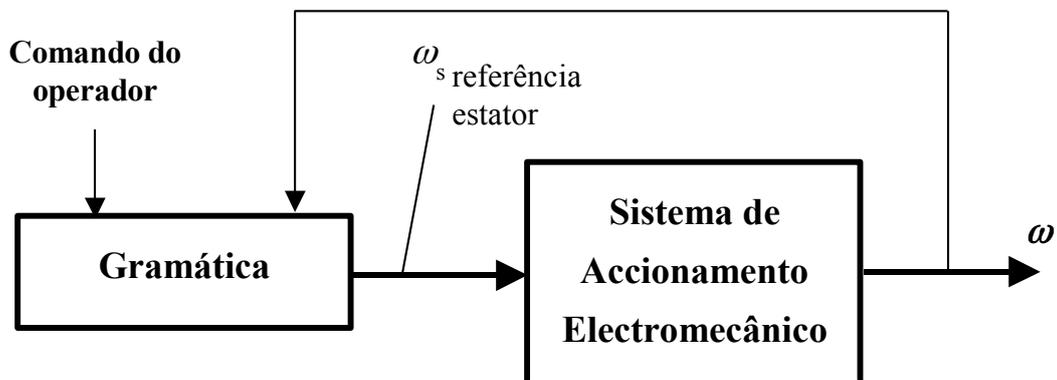
Deste modo, em função da entrada de comando externo as produções vão representar padrões típicos de funcionamento. Na Figura 5.31 apresenta-se um exemplo de aplicação, onde a gramática gera os padrões da variável de comando do accionamento em função da entrada de comando externa. Pode verificar-se, para uma carga constante,

quatro padrões de funcionamento distintos codificados através dos símbolos não terminais {'C', 'A', 'F', 'E'} da variável de comando.



**Figura 5.31. Comando supervisionado de um processo industrial.**

Em alternativa ao anterior comando de supervisão em cadeia aberta, pode considerar-se um controlo de supervisão em cadeia fechada, tal como se apresenta na Figura 5.32.



**Figura 5.32. Controlo gramatical supervisionado, em cadeia fechada, de um processo industrial baseado num accionamento com máquina assíncrona.**

A existência de um comando de operador pode ser entendida como um critério de escolha entre diferentes linguagens, todas elas associadas ao processo considerado. Deste modo a linguagem associada a todo o processo resultará da união de todas as

linguagens associadas a cada tipo de operação ( 5.49 ), onde  $L_i$  representa a linguagem associada ao valor  $i$  da variável de comando externa.

$$L = \bigcup_i L_i \quad ( 5.49 )$$

Uma vez que todas as linguagens partilham o mesmo alfabeto, cada uma das linguagens ( 5.50 ) é gerada por uma gramática  $G_i$  (possuindo o seu sistema de rescrita), associada a um determinado tipo de operação desejada.

$$L_i(G_i) = \left\{ z \in \Sigma_T^* \mid S \xRightarrow{R_i} z \right\} \quad ( 5.50 )$$

Em cada gramática associa-se o alfabeto terminal à velocidade (variável de saída do sistema de accionamento electromecânico) e o alfabeto não terminal às correntes estatóricas da máquina assíncrona desse mesmo sistema (variável de entrada).

Para as produções, da forma ( 3.18 ), considerou-se dois como o tipo máximo ( 5.51 ), onde  $U$  resulta da codificação da referência da frequência do estator fornecida ao accionamento electromecânico e  $y_i$  resulta da codificação da evolução da velocidade do accionamento. Tal como anteriormente, para evitar a influência do ruído nas produções inferidas, uma produção só é considerada válida se for extraída um número suficiente de vezes. Este procedimento funciona como um filtro contra perturbações de ruído.

$$y_1 y_2 U \rightarrow y_1 y_2 y_3 \delta \quad ( 5.51 )$$

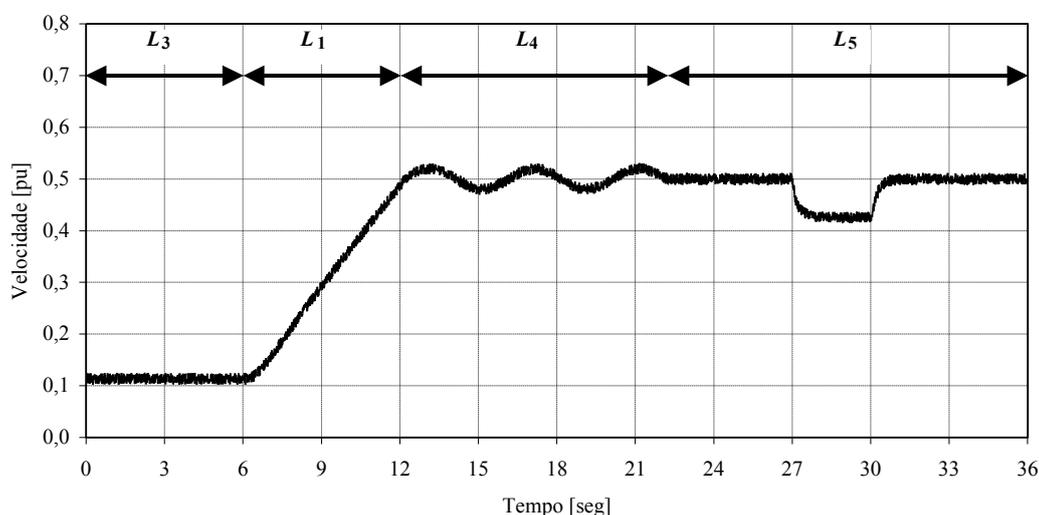
Tratando-se de um controlo em cadeia fechada, a gramática deverá providenciar a necessária evolução da corrente estatórica por forma a que a velocidade siga uma referência (linguagem) pré-determinada e função da variável de comando externa. Assim o processo de inferência das produções deverá, para cada linguagem, considerar informação proveniente de várias situações de funcionamento, isto é, para cargas distintas. Para cada linguagem, as produções da gramática associada deverão estabelecer a relação entre o comando e a saída do accionamento, para diferentes regimes de carga.

O processo anteriormente descrito pode ser observado na Figura 5.33. Considerando que a variável de comando externa providencia oito comportamentos distintos,

referenciados na Tabela 5.IV, para cada linguagem desejada (escolhida pelo operador) o accionamento deverá seguir a referência desejada. Perante uma alteração na carga mecânica, o sistema de rescrita deverá utilizar as produções necessárias por forma a que o accionamento continue a seguir a referência associada a essa linguagem, tal como pode ser observado na parte final do ensaio apresentado na Figura 5.33, onde o comando externo solicitava uma velocidade constante média.

**Tabela 5.IV – Especificação das linguagens em função do comando do operador.**

Linguagem	Comando Externo
1	Aceleração linear
2	Desaceleração linear
3	Velocidade constante baixa
4	Velocidade sinusoidal baixa
5	Velocidade constante média
6	Velocidade sinusoidal média
7	Velocidade constante elevada
8	Velocidade sinusoidal elevada



**Figura 5.33. Controlo supervisionado de um processo industrial.**

Nos exemplos anteriores as gramáticas, descritivas da linguagem associada a um qualquer sistema dinâmico comandado, foram utilizadas como entidades supervisoras do comportamento desses mesmos sistemas. Foram consideradas três perspectivas

distintas. Na primeira utilizaram-se as produções do sistema de rescrita para fazer evoluir o sistema dinâmico de forma a que a saída deste seguisse uma determinada palavra objectivo. Na segunda considerou-se um comando em cadeia aberta, onde a gramática é responsável por gerar padrões de funcionamento desejados por um operador externo. Na terceira, e utilizando a união de várias linguagens, estendeu-se o anterior comando a um controlo em cadeia fechada. Cada linguagem representa um comportamento típico desejado, sendo a gramática responsável pela utilização das produções que levam o sistema a seguir esse mesmo comportamento.

## 5.5 CONCLUSÕES

Neste capítulo foi analisada uma aplicação experimental do algoritmo de inferência gramatical em três vertentes distintas: modelização, detecção de anomalias e controlo. Foi considerado um sistema de accionamento electromecânico baseado numa máquina assíncrona alimentada por um conversor electrónico de potência.

A utilização de accionamentos baseados neste tipo de máquina levanta sérios problemas na sua modelização, os quais podem ser atenuados por recurso à utilização de algoritmos de aprendizagem através de exemplos. Estes algoritmos extraem uma relação funcional representativa do comportamento dinâmico do accionamento, a partir de um conjunto de dados experimentais.

Relativamente à primeira vertente, começou-se por caracterizar matematicamente o sistema de accionamento electromecânico, estabelecendo um conjunto de hipóteses simplificativas e especificando os alfabetos terminal e não terminal.

Foi analisada a sensibilidade da gramática inferida em relação ao tipo de alfabeto considerado, sendo que alfabetos de menor dimensão introduzem maiores erros quantitativos mas continuam a modelizar qualitativamente o accionamento electromecânico.

Foi também testado o mecanismo de interpolação gramatical, onde o efeito de produções inexistentes é atenuado pelo processo de preenchimento das lacunas gramaticais.

Analisou-se por fim a capacidade de generalização do algoritmo, confirmado-se o facto desta propriedade possuir carácter local, isto é, evidenciar-se essencialmente nas zonas do domínio onde ocorre recolha de informação utilizada na inferência das produções.

As gramáticas inferidas a partir de informação experimental, codificando uma classe de padrões pertencentes à linguagem associada ao sistema de accionamento electromecânico, apresentam-se como boas ferramentas na detecção de anomalias. A

utilização das gramáticas tanto pode ser realizada na óptica do modelo como da análise de padrões.

Sob o ponto de vista da detecção de anomalias baseada em modelos, por análise das palavras terminais, provenientes quer do accionamento quer da gramática a este associada, é possível avaliar a severidade da anomalia. Esta avaliação é efectuada medindo a distância entre ambas as palavras.

Sob o ponto de vista da análise de padrões utiliza-se a capacidade das gramáticas reconhecerem padrões em sinais. Assim perante um sinal que exiba um padrão diferente do habitual é reportada uma anomalia.

Depois de ter sido analisada a capacidade de aprendizagem, quer na extracção de um modelo representativo do comportamento do sistema de accionamento electromecânico quer na detecção de anomalias, foi avaliada a capacidade de utilização das gramáticas inferidas no controlo do sistemas de accionamento electromecânico.

Nesta perspectiva considerou-se a sua utilização quer como regulador quer como controlador de supervisão.

Como regulador, o controlador gramatical foi inserido, juntamente com um controlador convencional, numa topologia com retroacção da variável de saída. O controlador convencional considerado foi um controlador do tipo proporcional integral, permitindo este que, na fase inicial, se inicie o processo de recolha de informação experimental na ausência total de produções no sistema de rescrita da gramática. Assim que esta informação se apresentar em quantidade suficiente as produções entretanto inferidas são suficientes para que o controlo seja assegurado exclusivamente pelo controlador gramatical.

O processo de recolha de informação e actualização da gramática deve ser contínuo, pois somente desta forma é possível garantir a sua adaptação a novas situações de funcionamento. Esta capacidade foi analisada na adaptação a novas situações de carga e a alterações de parâmetros, onde se considerou quer a inferência de novas produções quer a correcção das já existentes.

Foi ainda testado o mecanismo de interpolação gramatical na regulação de velocidade do accionamento electromecânico. A sua consideração diminui as oscilações devidas à ausência de produções essenciais a um bom desempenho.

Como controlador de supervisão o objectivo é a geração de uma referência em função do objectivo a atingir. Este caso revela-se de particular importância no controlo de processos onde a evolução do mesmo se encontre sujeita a padrões determinados. A capacidade evidenciada pelas gramáticas em reconhecer ou gerar padrões é assim utilizada para gerar a referência inerente a um dado processo.



## CAPÍTULO 6 - CONCLUSÕES

**Resumo:** Neste capítulo apresentam-se as considerações finais sobre o presente trabalho. Elabora-se a sua síntese e apresentam-se as contribuições originais. É ainda perspectivado o seu desenvolvimento futuro.



## 6.1 SÍNTESE DO TRABALHO E CONTRIBUIÇÕES ORIGINAIS

Os accionamentos electromecânicos de velocidade variável são um elemento de grande importância na generalidade dos processos industriais. Estes têm sido alvo de enorme investigação e desenvolvimento estimulado quer por necessidade de melhores desempenhos dinâmicos quer pela preocupação na redução de custos.

Em particular, os accionamentos electromecânicos baseados em máquinas assíncronas de rotor em curto-circuito encontram-se em expansão, em detrimento de outro tipo de accionamentos mais clássicos. Tal deve-se ao baixo custo de fabricação e manutenção deste tipo de máquina e ao enorme desenvolvimento de que foram alvo os conversores electrónicos de potência utilizados na sua alimentação.

Os modernos accionamentos electromecânicos apresentam-se assim constituídos por uma associação entre a máquina eléctrica e o conversor electrónico de potência que a alimenta, a que se encontram associados sensores e processadores digitais, utilizados no controlo do accionamento. Todos estes elementos apresentam dinâmicas não lineares e variáveis no tempo. Deste modo a obtenção de modelos que, com precisão, representem o seu comportamento dinâmico, é tarefa complicada e alvo de investigação permanente.

O conhecimento preciso do modo de funcionamento de todos os componentes do accionamento é fundamental para a obtenção de bons desempenhos. Para tal é necessário possuir informação sobre o seu comportamento dinâmico, quer em qualidade quer em quantidade suficiente.

Perante este facto a utilização de algoritmos de aprendizagem através de exemplos apresenta-se como um bom procedimento no estabelecimento de modelos do accionamento. Estes algoritmos revelam-se bem adaptados à extracção de uma relação funcional que correlacione as variáveis presentes. A escolha das variáveis significativas na modelização é fundamental para que o processo de aprendizagem apresente sucesso. Aqui torna-se importante o conhecimento prévio sobre o funcionamento do accionamento, permitindo assim estabelecer relações entre as variáveis mais adequadas

à descrição dos fenómenos físicos que regem o comportamento dinâmico do accionamento.

Existem várias metodologias que permitem estabelecer, de uma forma automática e a partir de exemplos de funcionamento, relações funcionais representativas da dinâmica dos sistemas de accionamento electromecânico. Neste estudo recorreu-se á utilização de elementos da teoria formal da linguagem que, através de uma correcta especificação gramatical, possibilitam a modelização de sistemas dinâmicos comandados através de mecanismos de inferência gramatical.

A enorme complexidade revelada para todo o accionamento serviu de motivação para a simplificação dos modelos representativos da sua dinâmica por forma a simplificar o processo de modelização. O controlo por modo de deslizamento das correntes que percorrem os enrolamentos do estator da máquina assíncrona revelou-se uma boa opção, simplificando o modelo do sistema em estudo. Este tipo de controlo encontra-se bem adaptado a estruturas de topologia variável, como o conversor electrónico de potência utilizado, e permite ultrapassar as dificuldades relacionadas com a enorme complexidade do accionamento, facilitando a modelização do mesmo. O controlo efectivo das correntes é assegurado, como atestam os resultados experimentais.

As técnicas de linguagem formal têm sido utilizadas, no passado, para estudar sistemas dinâmicos autónomos. No entanto, para sistemas dinâmicos comandados – como é o caso dos sistemas de accionamento electromecânico – foram necessárias novas especificações que permitem distinguir entre informação gerada pelo sistema e pela entrada de comando. A aplicação destas técnicas a estes sistemas levou à formulação da modelização em termos de gramáticas dependentes do contexto.

O processo de modelização gramatical inicia-se necessariamente pela especificação dos alfabetos da gramática. No formalismo desenvolvido, o alfabeto terminal é associado à variável de saída do sistema dinâmico e o alfabeto não terminal à variável de comando. As produções estabelecem as relações entre o comando e a saída do sistema dinâmico, modelizando diferentes tipos de comportamento dinâmico em função do tipo considerado. O obtenção das produções resulta da aplicação de um algoritmo de inferência gramatical sobre um conjunto de dados experimentais, provenientes do sistema dinâmico.

Efectuando o processo de aprendizagem de uma forma criteriosa, foram verificados os bons desempenhos da gramática, na extração de uma relação funcional representativa da dinâmica do accionamento. A utilização da gramática providencia boas características de memorização, generalização, robustez e aprendizagem contínua. Aspecto importante relaciona-se com o carácter local da generalização. Os bons resultados encontram-se confinados a zonas do domínio onde se verificou a recolha experimental de exemplos, utilizados na aprendizagem.

A capacidade de aprendizagem evolutiva permite que a relação funcional se mantenha permanentemente actualizada e possa intervir, de uma forma automática, em cadeias de controlo. Eventuais alterações paramétricas são também assimiladas mediante aprendizagem.

Sendo a inferência gramatical um algoritmo de aprendizagem através de exemplos, podem existir falhas de informação que conduzam ao aparecimento de lacunas gramaticais – ausência de produções inferidas. Além da solução óbvia de aumentar a quantidade e qualidade de informação disponível para o treino, foi desenvolvido um mecanismo de inferência gramatical que visa estabelecer as produções em falta, apenas por recurso às entretanto já inferidas.

A utilização das gramáticas inferidas para reconhecer palavras bem formadas e rejeitar as anómalas potencia o estabelecimento de procedimentos de detecção de anomalias em sistemas dinâmico comandados. O algoritmo linguístico apresenta-se como uma boa ferramenta para uma atempada detecção de anomalias, rejeitando as espúrias e reportando apenas as sustentadas.

Os testes efectuados à modelização apontam para a sua utilização em estruturas de controlo ‘inteligente’, onde a sua capacidade de adaptação autónoma e automática a novas situações de funcionamento lhe aumentam a faculdade de intervenção em estruturas de controlo industrial.

Sendo a gramática uma boa representação da relação funcional entrada/saída do accionamento electromecânico, a sua utilização como regulador passa pela geração das acções de comando convenientes por forma que o accionamento siga, com erro

reduzido, uma dada referência de velocidade. O controlador gramatical foi assim integrado, com sucesso, numa topologia de compensação indirecta.

Alternativamente à utilização da gramática como regulador, esta foi também utilizada como supervisor em estruturas de controlo hierarquicamente superiores. Aqui o objectivo não foi o de regular a variável de saída do accionamento mas sim de especificar a sua entrada de comando em função do comportamento desejado para o mesmo.

A definição de contribuições originais de um qualquer trabalho de investigação é sempre tarefa difícil. O seu grau de incerteza não permite eliminar a possibilidade de resultados semelhantes terem sido atingidos por outras equipas de investigação. Assim, os seguintes aspectos são referidos dentro do conhecimento que o autor do presente trabalho possui na área de engenharia em que este se insere.

Como contribuição original principal destaque-se a utilização dos elementos da linguagem formal na identificação da relação funcional representativa do comportamento dinâmico de sistemas dinâmicos comandados.

Foi desenvolvido todo o formalismo necessário a que os elementos da teoria formal da linguagem se adaptassem, de uma forma coerente, à representação dos sistemas dinâmicos.

Foram ainda estabelecidos um algoritmo de inferência gramatical, responsável pelo estabelecimento das produções do sistema de rescrita da gramática considerada, e um mecanismo de interpolação gramatical, destinado a suprir a ausência de produções essenciais à descrição da linguagem associada ao sistema a representar.

Realizou-se uma selecção criteriosa das variáveis a considerar por forma a diminuir o seu número. Esta selecção resultou da simplificação efectuada no modelo do accionamento electromecânico, para a qual foi fundamental uma nova técnica de comando desenvolvida para o conversor electrónico de potência.

Foi ainda desenvolvida uma implementação conexionista aplicável às gramáticas inferidas a partir da informação proveniente de sistemas dinâmicos. A estrutura

conexionista desenvolvida, constituída por um nível de identificação de símbolos (terminais e não terminais) e por um nível de estabelecimento e aplicação de produções, possibilita o processo de aprendizagem e trata a informação recebida de uma forma distribuída e paralela.

Considera-se ainda inovador a utilização do formalismo gramatical na modelização, detecção de anomalias e controlo dos sistemas de accionamento electromecânico utilizados.

Em paralelo com o estudo efectuado sobre aprendizagem foi estabelecida uma correspondência entre os elementos dos sistemas lógicos (bases de dados, regras, inferência e questões) e as operações dinâmicas presentes em redes neuronais de ordem superior. Esta correspondência foi estabelecida como um dicionário que permite a correspondência entre as formulações simbólicas e as redes neuronais.



## **6.2 PERSPECTIVAS DE TRABALHO FUTURO**

Este trabalho de investigação não se encontra terminado, tendo-se cumprido apenas uma primeira etapa. Pretende-se continuar a desenvolver o estudo das linguagens formais e a sua aplicação a sistemas dinâmicos comandados.

As potencialidades evidenciadas pela metodologia apresentada, permitem encarar várias aplicações no âmbito do controlo de sistemas de accionamento electromecânico de velocidade variável com dinâmica mais complexa, designadamente em sistemas com várias variáveis de comando e saída.

Importante característica ao nível dos algoritmos de aprendizagem, ainda não explorada, é o efectivo varrimento do domínio de funcionamento durante a fase de aprendizagem, de modo a constituir um conjunto de treino que seja uma amostra representativa do referido domínio. Existe um significativo trabalho a efectuar nesta área.

Pretende-se ainda estudar a associação de probabilidades a cada produção, obtendo gramáticas probabilísticas, bem como a consideração de regras semânticas, por incorporação de características conhecidas do sistema.



## **BIBLIOGRAFIA**

- (Angluin, 1988) D. Angluin; “Queries and concept learning”, *Machine Learning* 2, pp. 319-342, 1982.
- (Aubry, Guyonnet e Lima, 1991) N. Aubry, R. Guyonnet e R. Lima; “Spatio-temporal analysis of complex signals: Theory and applications”, *J. Stat. Phys.* 66, 683, 1991.
- (Auerbach e Procaccia, 1990) D. Auerbach e I. Procaccia; “Grammatical complexity of strange sets”, *Phys. Rev. A* 41, pp. 6602-6614, 1990.
- (Babuska e Verbruggen, 1995) R. Babuska e H. B. Verbruggen; “Identification of composite linear models via fuzzy clustering”, *Proceedings da 3<sup>rd</sup> ECC’95*, pp. 1207-1212, 1995.
- (Barnden e Pollack, 1991) J. A. Barnden e J. B. Pollack; “Advances in Connectionist and Neural Computation Theory, vol. 1: High Level Connectionist Models”, Ablex, Norwood, 1991.
- (Bellini, Filippetti, Franceschini, Tassonie e Kliman, 2000) A. Bellini, F. Filippetti, G. Franceschini, C. Tassonie e G. B. Kliman; “Quantitative Evaluation on Induction Motor Broken Bars by Means of Electrical Signature Analysis”, *Conference Record of the 35th IEEE-IAS Annual Meeting and World Conference on Industrial Applications of Electrical Energy (IAS’2000)*, Roma, Itália, 2000.
- (Beghelli, Guidorzi e Soverini, 1990) S. Beghelli, R. P. Guidorzi e U. Soverini; “The Frisch scheme in dynamic system identification”, *Automatica*, vol. 26, no. 1, pp 171-176, 1990.

- (Blaschke, 1972) F. Blaschke; “The principle of field orientation as applied to the new transvector closed-loop control system for rotated-field machines”, Siemens review, 1972.
- (Bodson, Chaisson e Novotnak, 1995) M. Bodson, J. Chaisson e R. T. Novotnak; “Nonlinear speed observer for high-performance induction motion control”, IEEE Transactions on Industrial Electronics, Vol. 42, Agosto, 1995.
- (Botta, Giordana e Piola, 1997) M. Botta, A. Giordana and R. Piola; “Refining first order theories wh neural networks”, em Foundations of Intelligent Systems, Z. W. Ras e A. Skowron (Eds.), pp. 84-93, Lecture Notes in Artificial Intelligence 1325, (Springer, Berlin), pp. 84-93, 1997.
- (Branco, 1998) P. J. Costa Branco; “Aprendizagem por Exemplos Utilizando Lógica ‘Fuzzy’ na Modelização e Controlo de um Accionamento Electro-Hidráulico”, Tese de Doutoramento, Instituto Superior Técnico, Universidade Técnica de Lisboa, Setembro, 1998.
- (Branco e Dente, 1998) P. J. Costa Branco e J. A. Dente; “An experiment in automatic modeling an electrical drive system using fuzzy logic”, IEEE Trans. on Syst. Man & Cybern., Vol. 28, pp. 254-262, Maio, 1998.
- (Bunke, 1990) H. Bunke; “String Matching for Structural Pattern Recognition”, em “Syntactic and Structural Pattern Recognition. Theory and Applications”, H. Bunke e A. Sanfeliu (Eds.), World Scientific, pp. 119- 144, Londres, 1990.
- (Chomsky, 1959) N. Chomsky; “On certain formal properties of grammar”, Information and Control 2, pp. 137-167, 1959.
- (Chomsky, 1965) N. Chomsky; “Aspects of the theory of syntax”, MIT Press, Cambridge, Massachusetts, 1965.

- (Clarke, Gernoth e Ristig, 1994) J. W. Clarke, K. A. Gernoth e M. L. Ristig; “High-order probabilistic perceptrons as Bayesian inference engines”, Trieste preprint Ic/94/236, 1994.
- (Coelho e Cotta, 1988) Helder Coelho e José C. Cotta; Prolog by Example, Springer-Verlag, 1988.
- (Collet, Galves e Lopes, 1995) P. Collet, and A. Galves and A. Lopes; “Maximum likelihood and minimum entropy identification of grammars”, *Random & Computational Dynamics* 3, pp. 241, 1995.
- (Cruz e Cardoso, 2000) S. M. A. Cruz e A. J. M. Cardoso; “Stator Winding Fault Diagnosis in Three-Phase Synchronous and Asynchronous Motors, by the Extended Park’s Vector Approach”, Conference Record of the 35th IEEE-IAS Annual Meeting and World Conference on Industrial Applications of Electrical Energy (IAS’2000), Roma, Itália, 2000.
- (Dente, 1994) J. A. Dente; “Accionamentos Electromecânicos ‘Inteligentes’”, Seção de Máquinas Eléctricas e Electrónica de Potência, Instituto Superior Técnico, Universidade Técnica de Lisboa, 1994.
- (Dente e Vilela Mendes, 1996) J. A. Dente e R. Vilela Mendes; “Unsupervised learning in general connectionist systems”, *Network: Computation in Neural Systems*, 4, pp.337-347, 1996.
- (Dhaouadi, Bolognami e Zigliotto, 1992) R. Dhaouadi, S. Bolognami e M. Zigliotto; “Effective estimation of speed and rotor position of a PM synchronous motor drive by a Kalman filtering technique”, *Proceedings da IEEE Power Electronic Specialists Conference (PESC’92)*, pp. 951-957, Toledo, Espanha, 1992.
- (Duda e Hart, 1973) R. O. Duda e P. E. Hart; “Pattern Classification and Scene Analysis”, J. Wiley, Nova Iorque, 1973.

- (Elman, 1990) J. L. Elman; “Finding Structure in Time”, *Cognitive Science*, 14, pp. 179-211, 1990.
- (Elman, 1995) J. L. Elman; “Language as a dynamical system”, em “Explorations in the Dynamics of Cognition”, R. F. Port e T Van Gelder (Eds.), Cambridge, MIT Press, pp. 195-223, 1995.
- (Farmer, 1990) J. D. Farmer, “A Rosetta stone for connectionism”, *Physica D*42, pp. 153-187, 1990.
- (Fitzgerald, Kingsley e Umans, 1985) A. E. Fitzgerald, C. Kingsley e S. D. Umans, “Electric Machinery”, 4ª edição, McGraw Hill, 1985.
- (Fu, 1994) Li Min Fu; “Rule generation from neural networks”, “IEEE Trans. on Systems, Man and Cybernetics”, 24, pp. 1114-1124, 1994.
- (Fu e Booth, 1975a) K.S. Fu and T.L. Booth; “Grammatical inference: Introduction and Survey – Part I”, *IEEE Trans. on SMC* 5, no 1, pp. 95-111, 1975.
- (Fu e Booth, 1975b) K.S. Fu and T.L. Booth; “Grammatical inference: Introduction and Survey – Part II”, *IEEE Trans. on SMC* 5, no 4, pp 409-423, 1975.
- (Gallant, 1993) Stephen I. Gallant; “Neural Network Learning and Expert Systems”, MIT Press, Massachusetts, 1993.
- (Gold, 1967) E. M. Gold; “Language identification in the limit”, *Information and Control* 10, pp. 447-474, 1967.
- (Haykin, 1999) S. Haykin; “Neural Networks. A Comprehensive Foundation”, Prentice-Hall International, 1999.

- (Hertz, Krogh e Palmer, 1991) J. Hertz, A. Krogh e R. G. Palmer; "Introduction to the Theory of Neural Computation", Addison-Wesley, reading, MA, 1991.
- (Honavar e Uhr, 1994) V. Honavar e L. Uhr; "Artificial Intelligence and Neural Networks: Steps towards principled integration", Academic Press, San Diego, 1994.
- (Howie, 1991) J. M. Howie; "Automata and Languages", Oxford Science Publications, 1991.
- (Hsu, 1995) J. S. Hsu; "Monitoring of defects in induction motors through air-gap torque observation", IEEE Transactions on Industrial Applications, vol. 31, pp. 1016-1021, Setembro/Outubro, 1995.
- (Kain, 1972) Richard Y. Kain; "Automata Theory: Machines and Languages", McGraw Hill, 1972.
- (Kanal, 1974) L. Kanal; "Patterns in Pattern Recognition: 1968-1974", IEEE Transactions on Information Theory, vol. IT-20, pp. 697-722, Novembro, 1974.
- (Kantz e Schreiber, 1997) H. Kantz e T. Schreiber; "Non Linear Time Series Analysis", Cambridge Nonlinear Science series 7, Cambridge University Press, 1997.
- (Katic e Vukobratovic, 1995) D. M. Katic e M. K. Vukobratovic; "Highly Efficient Robot Dynamics Learning by Decomposed Connectionist Feedforward Control Structure", IEEE Trans. on Systems Man & Cybernetics, vol. 25, no. 1, pp. 145-158, 1995.
- (Kovacs, 1984) P. K. Kovacs; "Phenomena in Transient Analysis of Electrical Machines", Amesterdão, Holanda, Elsevier, 1984.

- (Lakdawala, 1996) P. Lakdawala; “The computational complexity of symbolic dynamics at the onset of chaos”, *Phys. Rev.* E53, pp. 4477-4485, 1996.
- (Leonard e Thomson, 1986) R. A. Leonard e W. T. Thomson; “Vibration and stray flux monitoring for unbalanced supply and inter-turn winding fault diagnosis in induction motors”, *Br. J. Non-Destr. Test.*, pp. 211-215, Julho, 1986.
- (Leonaritis e Billings, 1985) I. J. Leonaritis e S. A. Billings; “Input-Output parametric models for non-linear systems”, *International Journal of Control*, vol. 41, pp. 303-344, 1985.
- (Leigh, 1983) J. R. Leigh; “Essential of Nonlinear Control Theory”, *IEE Topics in Control, Series 2*, Peter Peregrinus Ltd, Londres, 1983.
- (Low, Lee e Lim, 1993) T.-S. Low, T. H. Lee e H. K. Lim; “*A methodology for neural network training from control of drives with nonlinearities*”, *IEEE Transactions on Industrial Electronics*, vol. 40, pp. 243-249, Abril, 1993.
- (Maia, 1994) J. H. Maia e J. A. Dente; “The use of the bi-orthogonal decomposition in a non-linear controller for electrical drives”, *Proceedings da 5th International Conference on Power Electronics and Variable-Speed Drives – PEVD’94*, pp. 107-112, Londres, Inglaterra, Outubro, 1994.
- (Maia, 1995) J. H. Maia; “Modelização Automática e Comando de Sistemas de Accionamento Electromecânico”, Tese de Doutoramento, Instituto Superior Técnico, Universidade Técnica de Lisboa, Setembro, 1995.
- (Maier, 1992) R. Maier; “Protection of squirrel-cage induction motor utilizing instantaneous power and phase information”, *IEEE Transactions on Industrial Applications*, vol. 28, pp. 376-380, Março/Abril, 1992.

- (Martins, 1996) J. F. Martins; “Modelização Automática Entrada/Saída de um Accionamento Electromecânico baseado numa máquina assíncrona de rotor em curto circuito”, Tese de Mestrado, IST, UTL, Lisboa, 1996.
- (Martins, Pires e Dente, 1997) “Automatic Input/Output Modeling of a Squirrel-Cage Induction Motor Drive System using Neural Network”; J. F. Martins, A. J. Pires, J. A. Dente; EPE’97; pp. 4.632-4.637; Trondheim; Noruega; Setembro 1997.
- (Martins, Pires e Dente, 1998) “A Choice of Variables in Automatic Modelling of AC-Drive Systems”; J. F. Martins, A. J. Pires, J. A. Dente; PEMC’98, pp. 5.7-5.12; Praga, República Checa; 1998.
- (Martins, Pires e Silva, 1998) J. F. Martins, A. J. Pires, J. F. Silva; “A Novel and Simple Current Controller for Three-Phase PWM Power Inverters”, IEEE Transactions on Industrial Electronics, vol. 45, pp. 802-805, Outubro, 1998.
- (Martins, Pires, Vilela Mendes, Dente, 2000a) J.F. Martins, A.J. Pires, R. Vilela Mendes, J.A. Dente; “A Novel Grammatical Learning Algorithm for Automatic Modelling of Drive Systems”, IEEE IAS 2000 Conference and Annual Meeting, Roma, Itália, Outubro 2000.
- (Martins, Pires, Vilela Mendes, Dente, 2000b) J.F. Martins, A.J. Pires, R. Vilela Mendes, J.A. Dente; “Modelling Electromechanical Drive Systems: A Formal Language Approach”, 1<sup>st</sup> ICAF Conference on Mechatronic Systems, Darmstad, Alemanha, Setembro 2000.
- (Martins, Branco, Pires, Dente, 2000) J.F. Martins, P. J. Costa Branco, A.J. Pires, J.A. Dente; “Fault Detection using Immune-Based Systems and Formal Language Algorithms”, 39<sup>th</sup> IEEE Conference on Decision and Control, CDC2000, Sidney, Austrália, Dezembro 2000.

- (Martins, Branco, Pires, Dente, 2001) J.F. Martins, P. J. Costa Branco, A.J. Pires, J.A. Dente; “Formal Language and Fuzzy Logic Theories: New Trends In Recognizing Experimental Drives”, IEEE Transactions on Fuzzy Systems, Special Issue of TFS on Recognition Technology, Vol. 9, No. 1, pp. 68-87, Fevereiro 2001.
- (Martins, Pires, Vilela Mendes, Dente, 2001) J.F. Martins, A.J. Pires, R. Vilela Mendes, J.A. Dente; “Language Identification of Controlled Systems: Modelling, Control and Anomaly Detection”, IEEE Transactions on Systems, Man and Cybernetics (Part C), Vol. 31, No. 2, pp. 234-242, Maio 2001.
- (Martins, Pires, Vilela Mendes, Dente, 2002a) J.F. Martins, A.J. Pires, R. Vilela Mendes, J.A. Dente; “Automatic Language Control of Drives”, 6<sup>th</sup> IEEE International Conference on Intelligent Engineering Systems, INES2002, Opatija, Corácia, Maio 2002.
- (Martins, Pires, Vilela Mendes, Dente, 2002b) J.F. Martins, A.J. Pires, R. Vilela Mendes, J.A. Dente; “Formal Language Control of Induction Motor Drives”, aceite para apresentação na Conference on Simulated Evolution And Learning (SEAL’02), Singapura, Novembro 2002.
- (Martins, Pires, Vilela Mendes, Dente, 2002c) J.F. Martins, A.J. Pires, R. Vilela Mendes, J.A. Dente; “Formal Language Based Learning Algorithm for Electrical Drives Control”, aceite para apresentação na 28<sup>th</sup> Annual Conference of the IEEE Industrial Electronics Society (IECON’02), Sevilha, Espanha, Novembro 2002.
- (Martins e Vilela Mendes, 2001) João Martins e R. Vilela Mendes; “Neural networks and logical reasoning systems. A translation table”, International Journal of Neural Sciences, Vol. 11, No. 2, pp. 179-186, 2001.
- (McClelland e Rumelhart, 1988) J. L. McClelland e D. E. Rumelhart; “Explorations in Parallel Distributed Processing”, Cambridge, Massachusetts, Bradford Books/MIT Press, 1988.

- (Miyamoto, Kawato, Setoyama e Suzuki, 1988) H. Miyamoto, M. Kawato, T. Setoyama e R. Suzuki; “Feedback-Error-Learning Neural Network for Trajectory Control of a Robotic Manipulator”, *Neural Networks*, vol. 1, pp. 251-265, 1988.
- (Narendra e Thathachar, 1989) K. S. Narendra e M. Thathachar; “Learning Automata. An Introduction”, Prentice-Hall International, 1989.
- (Nowak, Plotkin e Jansen, 2000) M. A. Nowak, J. B. Plotkin e V. A. A: Jansen; “The evolution of syntactic communication”, *Nature*, vol. 404, pp. 495-498, Março, 2000.
- (Omlin e Giles, 1994) C. W. Omlin e C. L. giles; “Extraction and Insertion of Symbolic Information in Recurrent Neural Networks”, em “Artificial Intelligence and Neural Networks: Steps Toward Principles Integreation”, V. Honovar e L. Uhr (Eds.), Academic Press, pp. 271-299, 1994.
- (Palma, 1984) J. C. Palma; “Electrónica de Potência”, Folhas de Apoio, Centro de Electrotecna e Electrónica Industrial, Instituto Superior de Engenharia de Lisboa, 1984.
- (Palma, 1993) J. C. Palma; “Controlo da Máquina Assíncrona Trifásica com Ondulador de Tensão - Desenvolvimento e Simplificação de Técnicas de Controlo Vectorial”, Tese de Doutoramento, Instituto Superior Técnico, Universidade Técnica de Lisboa, Maio, 1993.
- (Palma, 1999) J. C. Palma; “Accionamentos Electromcânicos de Velocidade Variável”, Fundação Calouste Gulbenkian, 1999.
- (Patton, Frank e Clark, 2000) Ron J. Patton, Paul M. Frank e Robert N. Clark; “Issues of Fault Diagnosis for Dynamic Systems”, Springer-Verlag, 2000.

- (Penman, Sedding, Lloyd e Fink, 1994) J. Penman, H. G. Sedding, B. A. Lloyd e W. T. Fink; “Detection and location of interturn short circuits in the stator windings of operation motors”, IEEE Transactions on energy Conversion, vol. 9, pp. 652-658, Dezembro, 1994.
- (Pinker, 2000) S. Pinker; “Survival of the clearest”, Nature, vol. 404, pp. 441-442, Março, 2000.
- (Pinker e Prince, 1988) S. Pinker e A. Prince; “On language and connectionism: Analysis of a parallel distributed processing model of language acquisition”, Cognition 28 (1988) 73.
- (Pires e Dente, 1991a) A. J. Pires e J. A. Dente; “A New Induction Machine Model for Tracking--Controllers Synthesis”, Proceedings 13th Congress IMACS'91, vol. 3, pp. 1306-1307, Dublin, Irlanda, Julho, 1991.
- (Pires, e Dente, 1991b) A. J. Pires, A. Dente; “A New Induction Machine Model for the Synthesis of Tracking Controllers - Implementation Aspects Approach,” EPE'91, Florência, Itália, 1991.
- (Pollack, 1990) J. B. Pollack; “Recursive distributed representations”, Artificial Intelligence 46, pp. 77-105, 1990.
- (Sage e Melsa, 1971) A. P. Sage e J. L. Melsa; “System Identification”, Academic Press, Nova Iorque, EUA, 1971.
- (Sakakibara e Golea, 1995) Y. Sakakibara e M. Golea; “Simple recurrent networks as generalized hidden markov models with distributed representations”, Proceedings da IEEE Internat. Conf. on Neural Networks (ICNN'95), IEEE Computer Society Press, New York, pp. 974-984, 1995.
- (Santana e Labrique, 1991) J. Santana e F. Labrique; “Electrónica de Potência”, Ed. Gulbenkian, Lisboa, 1991.

- (Sen, 1990) P. Sen; “Electric Motor Drives and Control - Past, Present and Future”, IEEE Transactions on Industrial Electronics, vol. 37, n° 6, pp. 562-575, Dezembro, 1990.
- (Sethi e Yoo, 1996) I. K. Sethi and J. H. Yoo; “Symbolic mapping of neurons in feedforward networks”, Pattern Recognition Lett. 17, pp. 1035-1046, 1996.
- (Shastri, 1990) L. Shastri; “Connectionism and the computational effectiveness of reasoning”, Theoretical Linguistics 16, pp. 65-87, 1990.
- (Simões e Bose, 1994) M. G. Simões e B. K. Bose; “Neural network based estimation of feedback signals for a vector controlled induction motor drive”, IEEE IAS Annual Meeting, pp. 471-479, 1994.
- (Softy e Kammen, 1991) W. R. Softy e D. M. Kammen; “Correlations in high dimensional or asymmetric data sets: Hebbian neuronal processing”, Neural Networks, 4, pp. 337-347, 1991.
- (Stavrou, Sedding e Penman, 1999) A. Stavrou, H. Sedding e J. Penman; “Current monitoring for detecting inter-turn short circuits in induction motors”, Proceedings da IEEE International Electric Machines and Drives Conference, pp. 345-347, Seattle, EUA, Maio, 1999.
- (Stone e Sedding, 1995) G. C. Stone e H. G. Sedding; “In-service evaluation of motor and generator stator windings using partial discharges tests”, IEEE Transactions on Industrial Applications, vol. 31, pp. 299-303, Março/Abril, 1995.
- (Sun, 1992) R. Sun; “On variable binding in connectionist networks”, Connection Science 4, pp. 84-93, 1992.
- (Sun, 1993) R. Sun; Integrating Rules and Connectionism for Robust Reasoning, Wiley, Nova Iorque, 1993.

- (Taylor e Coombes, 1993) J. G. Taylor e S. Coombes; “Learning higher order correlations”, *Neural Networks*, 6, pp. 423-427, 1993.
- (Thomason, 1990) Michael G. Thomason; “Introduction and Overview”, em “Syntactic and Structural Pattern Recognition. Theory and Applications”, H. Bunke e A. Sanfeliu (Eds.), World Scientific, pp. 119- 144, Londres, 1990.
- (Thomason e Rankin, 1987) W. T. Thomason e D. Rankin; “Case Histories of Rotor Winding Fault Diagnosis in Induction Motors”, 2<sup>nd</sup> Int. Conf. Proc. on Condition Monitoring, University College Swansea, 1987.
- (Towell e Shavlik, 1993) G. Towell and J. Shavlik; “The extraction of refined rules from knowledge-based neural networks”, *Machine Learning* 13, pp. 71-101, 1993.
- (Troll, 1993) G. Troll; “Formal languages in dynamical systems”, *Acta Univ. Carolin. Math. Phys.* 34, pp. 117, 1993.
- (Tsetlin, 1962) M. Tsetlin; “On the behaviour of finite automata in random media”, *Automat. remote Contr.*, vol. 22, pp. 1210-1219, 1962.
- (Utkin, 1977) V. I. Utkin; “Variable Structure Systems with Sliding Modes”, *IEEE Transactions on AC*, vol. 22, n° 2, pp. 212-222, Abril 1977.
- (Utkin, 1987) V. I. Utkin; “Discontinuous Control Systems: State of Art in Theory and Applications”, *Proceedings 10° IFAC*, pp. 75-94, Munique, RFA, Julho 1987.
- (Valiant, 1984) L. G. Valiant; “A theory of the learnable”, *Comm. ACM* 27, pp. 1134-1142, 1984.

(Vas, Drury e Stronach, 1996) P. Vas, W. Drury e A. F. Stronach; “Full fuzzy control of a DSP-based high-performance induction motor drive”, IEE Proceedings, 144, pp. 361-368.

(Wang e Xie, 1994) Y. Wang and H. Xie; “Grammatical complexity of unimodal maps with eventually periodic kneading sequences”, Nonlinearity 7, pp. 1419-1436, 1994.

(Wolfram, 1984) S. Wolfram; “Computational theory of cellular automata”, Commun. Math. Phys. 96, pp. 15-57, 1984.



# APÊNDICE A - REDES NEURONAIS E SISTEMAS LÓGICOS

**Resumo:** Neste apêndice é estabelecida uma correspondência entre os elementos dos sistemas lógicos (bases de dados, regras, inferência e questões) e o *hardware* e operações dinâmicas das redes neuronais. No capítulo 4 foi estabelecida uma implementação conexionista, possibilitando o tratamento da informação recebida de uma forma distribuída e paralela, da gramática inferida a partir do comportamento de um sistema dinâmico. Neste apêndice a correspondência é estabelecida como um dicionário de transição que permite a correspondência entre as formulações simbólicas e das redes neuronais. No formalismo das cláusulas Horn as proposições atômicas de  $n$  argumentos correspondem a nós com sinapses de ordem  $n$ , as regras correspondem a constrangimentos nas intensidades sinápticas, o *forward chaining* à dinâmica das sinapses e as questões quer à activação simples de um nó quer à dinâmica de tensores.



## A.1 INTRODUÇÃO

As redes neuronais e os sistemas lógicos, uma vez que ambos se baseiam em modelos de computação do tipo máquina de Turing, são equivalentes no sentido em que o que é computável num formalismo também o deve ser noutro. A forma de estabelecer esta equivalência não é trivial porque os problemas simples num formalismo apresentam-se como intratáveis noutro e vice-versa. Por exemplo, problemas de optimização combinatoria são tratados de uma forma eficiente através de redes neuronais, ao invés dos sistemas lógicos que sucumbem a uma explosão combinatoria. Por outro lado, para problemas altamente estruturados, onde uma heurística apropriada é conhecida, é mais simples e eficiente a utilização de sistemas lógicos, sendo que a correspondente abordagem neuronal levará muito tempo a afinar, pois a informação baseada em regras não se implementa facilmente em redes neuronais.

Esta situação levou um certo número de autores (Sun, 1993) (Honavar e Uhr, 1994) (Barnden e Pollack, 1991) (Shastri, 1990) (Pollack, 1990) (Pinker e Prince, 1988) a propor a utilização de sistemas híbridos para resolver problemas complexos, com programação lógica para algumas partes do processo e redes para as restantes. Por outro lado, alguns autores têm abordado o problema de identificar estruturas de redes que possam representar as estruturas lógicas dos sistemas simbólicos (Gallant, 1993) (Towell e Shavlik, 1993) (Fu, 1994) (Sethi e Yoo, 1996) (Botta, Giordana e Piola, 1997) (Sun, 1992), e assim como a natureza da informação baseada em regras que pode ser extraída de redes.

A extração de regras de redes é um assunto importante na construção de sistemas periciais a partir de redes treinadas através de exemplos. Por outro lado, quando alguma informação *á priori*, baseada em regras, é conhecida sobre um determinado problema, mesmo que a implementação através de uma rede seja apropriada, será útil possuir alguns procedimentos simples que implementem a informação simbólica conhecida na arquitectura da rede. As redes neuronais apresentam a vantagem das implementações VLSI que largamente aumentam a velocidade de processamento, o que não é claro nas implementações de processamento simbólico. Desta forma, mesmo para problemas que são claramente definidos em termos lógicos, pode ser útil a utilização de um dicionário

que auxilie a sua implementação em *hardware*. O estabelecimento, de uma forma concisa, de uma correspondência entre as formulações simbólicas e de redes é útil em sistemas de aprendizagem e no estabelecimento de redes de multicomputadores. Nestas redes é essencial que os algoritmos e as arquitecturas se encaixem o melhor possível. Independentemente dos aspectos práticos, o estabelecimento de um dicionário entre os dois paradigmas é um desenvolvimento importante no estabelecimento de uma linguagem unificada para os processos cognitivos.

Num artigo clássico, Doyne Farmer (Farmer, 1990) mostrou que existe uma organização comum na qual redes neuronais, sistemas de classificação, redes imunitárias e redes de reacção autocalítica podem ser tratadas de uma forma unificada. O modelo geral, para o qual todos esses modelos são transpostos, providencia uma extensão do âmbito dos modelos das redes neuronais (Dente e Vilela Mendes, 1996). De uma forma similar, o providenciar uma ponte entre as redes neuronais e os sistemas lógicos pode sugerir novos algoritmos e aplicações em ambos os domínios.

A correspondência entre redes neuronais e sistemas lógicos será efectuada envolvendo os respectivos elementos básicos de cada formalismo. Numa rede neuronal existe uma memória (distribuída) nas conexões sinápticas (sinapses), uma dinâmica de aprendizagem nas sinapses e uma dinâmica de activação nas intensidades dos nós. Nos sistemas lógicos existe um conjunto de factos acerca dos objectos do domínio, um conjunto de regras que constituem conhecimento potencial acerca das relações entre os objectos e um mecanismo de inferência (*backward* ou *forward chaining*) que permite a extracção de mais informação e a resposta a questões.

Tem sido argumentado que a tentativa de isolar, numa rede, as estruturas que correspondem a afirmações ou operações lógicas específicas é uma perda de tempo porque numa rede tudo (memória, regras e inferência) se encontra distribuído por todo o lado de uma forma não separável. Isto pode ser verdade para algumas arquitecturas e algumas classes de conceitos. No entanto, mesmo a identificação de estruturas modulares que correspondam a elementos lógicos é um importante desenvolvimento. Concluir-se-á que a representação natural, numa rede neuronal, de uma proposição atómica é um nó com sinapses de ordem  $n$ .

## A.II REPRESENTAÇÃO CONEXIONISTA DE UM SISTEMA LÓGICO

O âmbito dos sistemas lógicos aqui considerado é um subconjunto da linguagem de programação *Prolog* – linguagem de programação lógica desenvolvida a partir da década de 70. Nomeadamente, o sistema lógico é especificado por um conjunto de cláusulas Horn que são construídas a partir de termos que se apresentam como constantes, variáveis ou estruturas. Uma constante representa um objecto concreto no domínio do problema. É representada no sistema lógico por um símbolo indecomponível elementar (um átomo). As estruturas restringem-se a proposições atómicas com a forma geral ( A.1 ).

$$\textit{functor} (\text{lista de parâmetros}) \quad (\text{ A.1 } )$$

O *functor* é um átomo e a lista de parâmetros é uma qualquer lista de átomos, variáveis ou qualquer outra proposição atómica. Por fim, uma variável é uma entidade que pode ser em qualquer altura ligada a qualquer átomo (constante ou *functor*). Usar-se-ão letras minúsculas para representar átomos e maiúsculas para as variáveis.

Como primeiro passo para estabelecer a representação, na rede neuronal, das entidades básicas do sistema lógico, far-se-á corresponder cada átomo a um nó. Assim, não apenas as constantes mas também os *functor* serão estabelecidos como um nó da rede.

Os nós da rede terão saída zero ou um e um limiar unitário. Os nós serão conectados na rede, associando-se às entradas  $e_1 \cdots e_n$  conexões sinápticas (sinapses)  $W_{f,i_1,\dots,i_n}$  de ordem  $n$  (Softy e Kammen, 1991) (Taylor e Coombes, 1993) (Clarke, Gernoth e Ristig, 1994), onde  $n$  representa o número de parâmetros da lista de parâmetros do functor. A constante é um caso particular em que não existem entradas. A activação de cada nó é dada por ( A.2 ), onde a polarização  $y_f$  é nula excepto quando se considera uma questão que envolva o nó em causa. A Figura A.1 apresenta a representação, na rede neuronal, de uma constante e a Figura A.2 de um *functor*. A notação  $\rho(\cdot)$  refere-se à função de Heaviside com um número arbitrário de argumentos.

$$s_f = \rho \left( \sum_{i_1, \dots, i_n} W_{f, i_1, \dots, i_n} e_1 \cdots e_n + y_f \right) \quad (A.2)$$

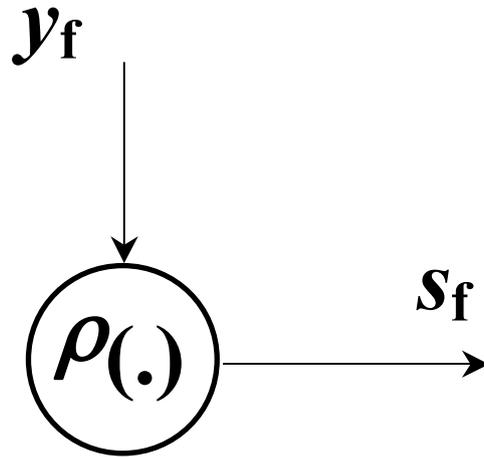


Figura A.1. Estabelecimento de uma constante na rede neuronal.

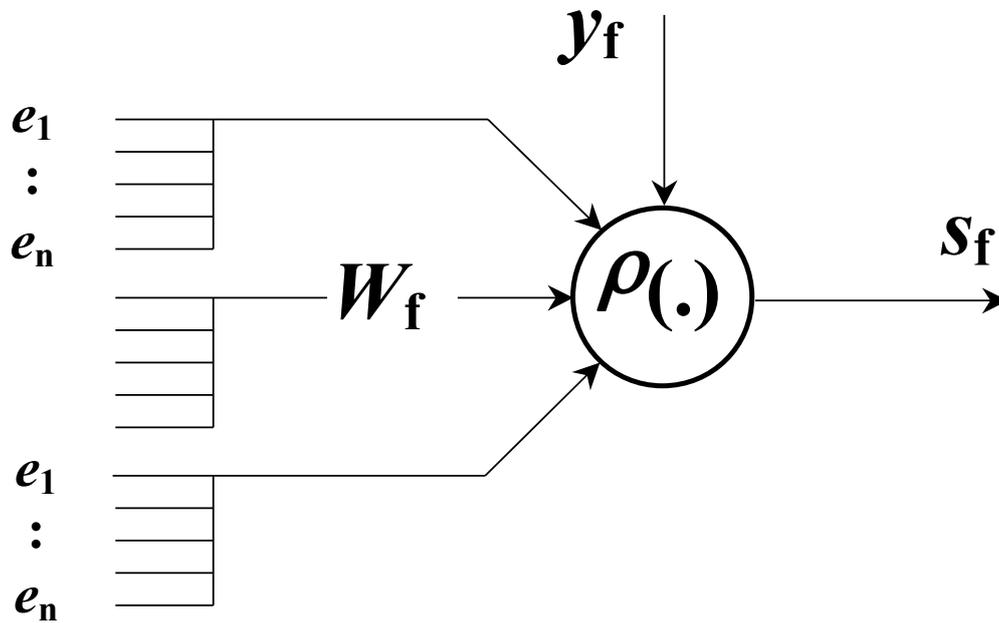


Figura A.2. Estabelecimento de um *functor* na rede neuronal.

Sempre que, na base de conhecimento lógico, a relação ( A.3 ) se apresentar como verdadeira a ligação sináptica correspondente será dada por ( A.4 ). Nesta fórmula, e nas seguintes,  $f$ ,  $m$ ,  $p$ , e  $g$  representam quaisquer *functor*.

$$f(i_1, \dots, i_n) \quad (A.3)$$

$$W_{f, i_1, \dots, i_n} = 1 \quad (A.4)$$

Em caso contrário, se a relação  $f$  não for verdadeira para um dado conjunto de parâmetros a respectiva conexão sináptica anular-se-á. Desta forma representa-se, na rede neuronal, os factos da base de conhecimento lógica, isto é, as relações que são explicitas acerca dos átomos. Por outro lado as relações e cláusulas que envolvam variáveis correspondem a constrangimentos nas restantes conexões sinápticas. Considere-se como exemplo a seguinte base de conhecimento, dada por ( A.5 ). Aqui a, b, c e d representam constantes e  $X$ ,  $Y$  e  $Z$  variáveis.

$$\begin{aligned} & f(b, c) \\ & m(a, c) \\ & m(d, a) \\ & p(X, Y) \Leftarrow m(X, Y) \\ & p(X, Y) \Leftarrow f(X, Y) \\ & g(X, Z) \Leftarrow p(X, Y) \wedge p(Y, Z) \end{aligned} \quad (A.5)$$

Os factos (referentes às três primeiras cláusulas) implicam as conexões sinápticas ( A.6 ) e todas as restantes  $W_{fXY}$  e  $W_{mXY}$  permanecerão nulas.

$$\begin{aligned} W_{fbc} &= 1 \\ W_{mac} &= 1 \\ W_{mda} &= 1 \end{aligned} \quad (A.6)$$

As conexões sinápticas tomarão apenas os valores zero ou um. As últimas três cláusulas em ( A.5 ) implicarão os constrangimentos ( A.7 ) nas restantes conexões sinápticas  $W_{pXY}$  e  $W_{gXY}$ .

$$\begin{aligned}
 W_{pXY} &\geq W_{mXY} \\
 W_{pXY} &\geq W_{fXY} \\
 W_{gXZ} &\geq W_{pXY}W_{pYZ}
 \end{aligned}
 \tag{A.7}$$

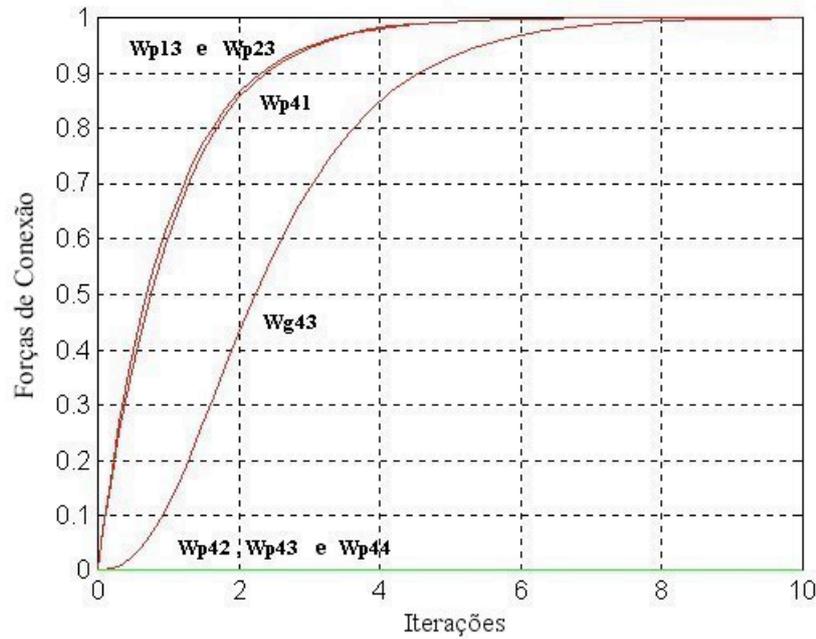
Estes constrangimentos podem ser realizados num rede neuronal através de uma lei dinâmica paralela. Para tal, define-se o potencial  $V_1$  ( A.8 ), onde  $\chi(x) = x^2 \rho(x)$ .

$$V_1 = \frac{1}{2} \sum_{XY} \chi(W_{mXY} - W_{pXY}) + \frac{1}{2} \sum_{XY} \chi(W_{fXY} - W_{pXY}) + \frac{1}{2} \sum_{XY} \chi(W_{pXY}W_{pYZ} - W_{gXZ}) \tag{A.8}$$

O conjunto de conexões ( A.6 ), fixadas pelos factos, são consideradas congeladas e as restantes evoluem em paralelo de acordo com a seguinte dinâmica de gradiente descendente ( A.9 ).

$$\begin{aligned}
 \frac{dW_{pXY}}{dt} &= - \frac{\partial V_1}{\partial W_{pXY}} \\
 \frac{dW_{gXY}}{dt} &= - \frac{\partial V_1}{\partial W_{gXY}}
 \end{aligned}
 \tag{A.9}$$

Esta lei dinâmica leva as forças de conexão ou para zero ou para um valor que satisfaça os constrangimentos ( A.7 ). Na Figura A.3 apresenta-se um exemplo da evolução dinâmica das forças de conexão em função das iterações.



**Figura A.3. Exemplo da evolução das forças de conexão.**

O método utilizado neste exemplo é válido em geral, e por cada regra da forma ( A.10 ) deve acrescentar-se um termo ( A.11 ) ao potencial  $V_1$ , sendo a soma efectuada sobre todas variáveis envolvidas.

$$g(X_1, \dots, X_n) \Leftarrow \prod_i \wedge f_i(Y_1, \dots, Y_{p_i}) \quad (\text{A.10})$$

$$\frac{1}{2} \sum_{\{X\}\{Y\}} \chi \left( \prod_i W_{f_i Y_1 \dots Y_{p_i}} - W_{g X_1 \dots X_n} \right) \quad (\text{A.11})$$

Até aqui foram estabelecidas as correspondências entre as proposições atômicas com  $n$  parâmetros e nós de ordem  $n$ , bem como entre regras e a dinâmica paralela das sinapses. Uma vez que a dinâmica das sinapses é equivalente a um passo de *forward chaining*, a rede contém nas suas forças de conexão toda informação acerca do problema. Deste modo a informação pode ser retirada observando apenas os valores das forças de conexão. Alternativamente pode ser estabelecido um esquema explícito para responder a questões. Considerar-se-ão dois tipos de questões: questões simples sem variáveis e questões que envolvam variáveis.

No primeiro caso, por exemplo em ( A.12 ), a questão é saber se a relação  $g$  é válida para os átomos ‘a’ e ‘b’. Neste caso é suficiente excitar as polarizações  $y_a$  e  $y_b$  dos nós ‘a’ e ‘b’ e verificar se o nó ‘g’ fica activo.

$$? \Leftarrow g(a,b) \quad (A.12)$$

O mesmo acontece em questões mais complexas, como por exemplo ( A.13 ), onde as polarizações a serem excitadas são  $y_a, y_b, y_c$  e  $y_d$  e a resposta é verdadeira se os nós ‘g’ e ‘f’ ficarem activos em simultâneo.

$$? \Leftarrow g(h(a,b),c) \wedge f(c,d) \quad (A.13)$$

Para questões que contenham variáveis, como por exemplo ( A.14 ), a questão é descobrir as instanciações de  $X$ , caso exista alguma, que façam de ( A.14 ) uma afirmação verdadeira.

$$? \Leftarrow m(d,X) \wedge p(X,c) \quad (A.14)$$

O produto das forças de conexão  $W_{mdX}W_{pXc} = \alpha_X$  definem um vector  $\alpha_X$  no espaço dos átomos e as entradas não nulas do vector são as instanciações que satisfazem a questão.

Um esquema semelhante é válido para questões que envolvem varias variáveis. Por exemplo em ( A.15 ) define-se um tensor  $\beta_{XY} = W_{gha}W_{hXc}W_{fcY}$ , onde as entradas não nulas são as respostas positivas à questão.

$$? \Leftarrow g(h(X,c),a) \wedge f(c,Y) \quad (A.15)$$

Com todas as forças de conexão estabelecidas na rede através da dinâmica ( A.9 ) – *forward chaining* – é possível observar as entradas não nulas de  $\alpha_X$  e  $\beta_{XY}$ . Alternativamente é possível considerar as forças de conexão como conhecimento potencial e gerar dinamicamente as entradas não nulas dos tensores ( A.16 ), com ( A.17 ).

$$\frac{d\alpha_X}{dt} = -\frac{\partial V_2}{\partial \alpha_X}$$

$$\frac{d\beta_{XY}}{dt} = -\frac{\partial V_3}{\partial \beta_{XY}}$$
( A.16 )

$$V_2 = \sum_X (W_{mdX} W_{pXc} - \alpha_X)^2$$

$$V_3 = \sum_{XY} (W_{gha} W_{hXc} W_{fcY} - \beta_{XY})^2$$
( A.17 )

Verificando a estrutura da rede e as forças de conexão como conhecimento potencial (em oposição a um conhecimento explícito), a dinâmica dos tensores de questões, definidos anteriormente, desempenha um papel similar ao processo de *backward chaining* na lógica de programação. Tal como em ( A.10 ) ( A.11 ) a generalização desta dinâmica, no espaço dos tensores de questões, para questões gerais é directa.

Na Tabela A.I apresenta-se um resumo de toda tradução entre o sistema lógico e a rede neuronal.

**Tabela A.I. Tabela de transição entre o sistema lógico e a rede neuronal.**

Sistema Lógico	Rede Neuronal
- Constante	- Nó simples
- Proposição atómica ( <i>n</i> argumentos)	- Nó com sinapses de ordem <i>n</i>
- Regras	- Constrangimentos nas intensidades sinápticas
- Aplicação de regras ( <i>forward chaining</i> )	- Dinâmica das sinapses
- Questões ( <i>backward chaining</i> )	- Activação dos nós / Dinâmica dos tensores de questões

Como exemplo simples de aplicação apresenta-se o problema 148 presente em (Coelho e Cotta, 1988), originalmente apresentado em “The Thinking Computer” por Bernrad Raphael. Considerem-se os seguintes *functor*:

<i>número</i> ( $X,N$ )	quer dizer que a pessoa $X$ pode ser contactada telefonicamente no número $N$
<i>visita</i> ( $X,Y$ )	quer dizer que a pessoa $X$ está a visitar a pessoa $Y$
<i>está</i> ( $X,Y$ )	quer dizer que a pessoa $X$ está em casa da pessoa $Y$
<i>telefone</i> ( $X,N$ )	quer dizer que a pessoa $X$ tem o número de telefone $N$

Adicionalmente considerem-se as cláusulas:

$$\begin{aligned} \textit{está} (X,Z) &\Leftarrow \textit{visita} (X,Y) \wedge \textit{está} (Y,Z) \\ \textit{número} (U,N) &\Leftarrow \textit{está} (U,V) \wedge \textit{número} (V,N) \end{aligned}$$

O objectivo é contactar telefonicamente uma dada pessoa, possuindo uma base de dados com números de telefone e sabendo quem visita quem e onde se encontra a pessoa visitada. Neste contexto considere-se uma base de conhecimento constituída pelos seguintes factos:

*número* (Coleman, '100001')  
*número* (Gordon, '100002')  
*número* (Wagner, '100003')  
*número* (Smith, '100004')  
*visita* (Coleman, Wagner)  
*está* (Wagner, Gordon)

As cláusulas neste exemplo definem o potencial ( A.18 ).

$$V_4 = \frac{1}{2} \sum_{XYZ} \chi(W_{\textit{visita}XY} W_{\textit{está}YZ} - W_{\textit{está}XZ}) + \frac{1}{2} \sum_{UVN} \chi(W_{\textit{está}UN} W_{\textit{número}VN} - W_{\textit{número}UN}) \quad (\text{A.18})$$

A evolução das forças de conexão, de acordo com a dinâmica de gradiente descendente, origina a rede neuronal apresentada na Figura A.4, onde apenas as conexões não nulas são apresentadas. As linhas a contínuo representam as forças de conexão dos factos da base de conhecimento e as linhas a tracejado representam as forças de conexão das cláusulas utilizadas para construir o potencial  $V_4$ .

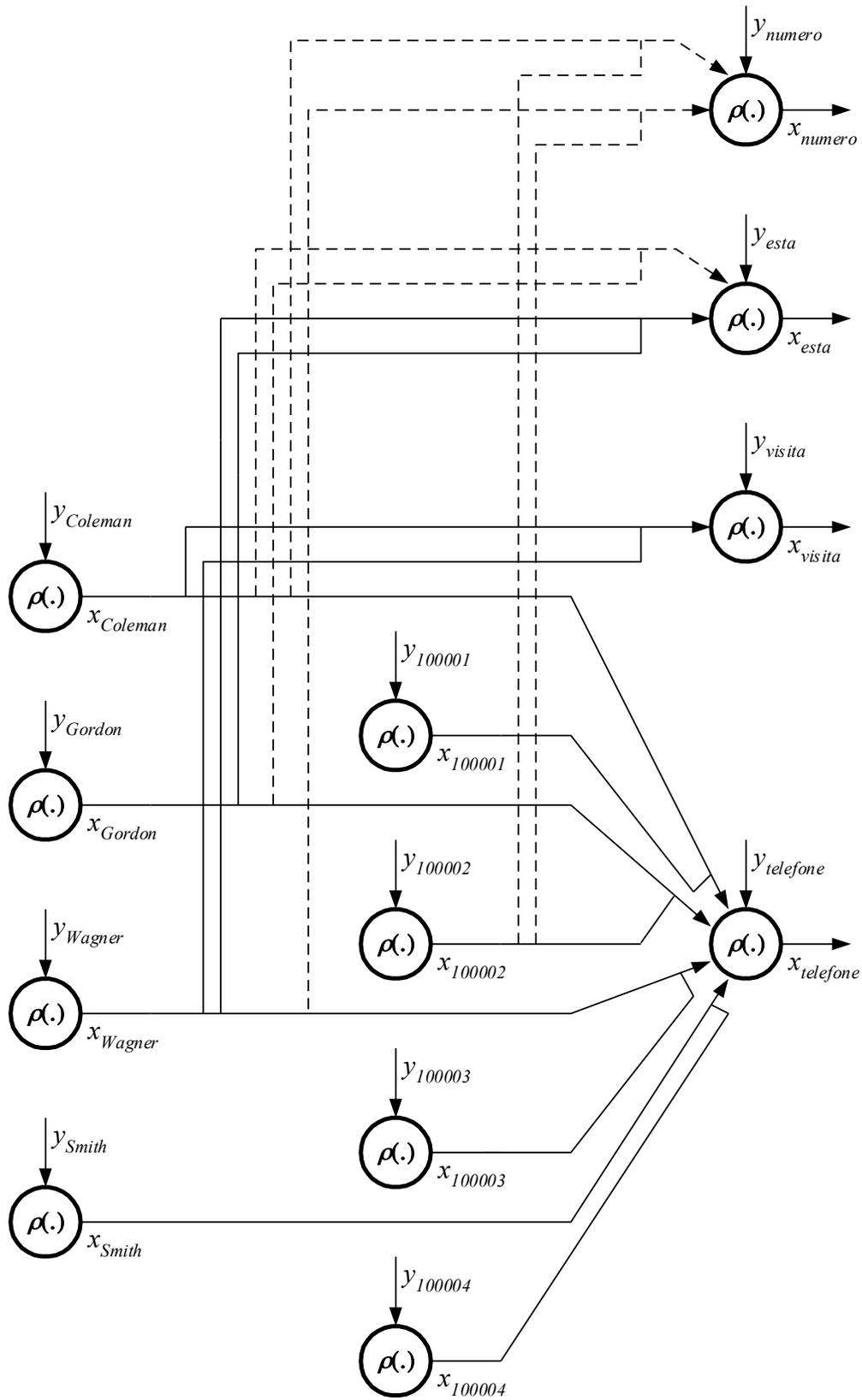


Figura A.4. Representação neuronal de um problema lógico.

