



UNIVERSIDADE ESTADUAL DE CAMPINAS - UNICAMP
FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO - FEEC
DEPARTAMENTO DE SISTEMAS DE CONTROLE E ENERGIA - DSCE
LABORATÓRIO DE SISTEMAS MODULARES ROBÓTICOS - LSMR



ESTUDOS E IMPLEMENTAÇÕES DE DINÂMICA CAÓTICA UTILIZANDO DISPOSITIVOS ANALÓGICOS RECONFIGURÁVEIS

Filipe Ieda Fazanaro

Dissertação submetida à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas, como parte dos requisitos exigidos para obtenção do título de Mestre em Engenharia Elétrica. **Área de concentração:** Automação

Banca Examinadora:

Prof. Dr. Marconi Kolm Madrid (Orientador) - DSCE/FEEC/UNICAMP

Prof. Dr. José Raimundo de Oliveira - DCA/FEEC/UNICAMP

Prof. Dr. Reinaldo Gonçalves Nogueira - EEEEC/UFG

Prof. Dr. Romis Ribeiro de Faissol Attux - DCA/FEEC/UNICAMP

Campinas, SP
12 de dezembro de 2007

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA - BAE - UNICAMP

F297e Fazanaro, Filipe Ieda
Estudos e implementações de dinâmica caótica utilizando dispositivos analógicos reconfiguráveis / Filipe Ieda Fazanaro. – Campinas, SP: [s.n.], 2007.

Orientadores: Marconi Kolm Madrid.
Dissertação (mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Circuitos elétricos não lineares. 2. Comportamento caóticos nos sistemas. 3. Análise de séries temporais - Processamento de dados. 4. Teoria de Controle. 5. Robótica I. Madrid, Marconi Kolm. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título

Título em Inglês:	Studies and implementations of chaotic dynamics using reconfigurable analog devices
Palavras-chave em Inglês:	Chaotic dynamics, Field programmable analog array, Hardware reconfigurable implementation, Time-series
Área de concentração:	Automação
Titulação:	Mestre em Engenharia Elétrica
Banca Examinadora:	José Raimundo de Oliveira, Reinaldo Gonçalves Nogueira, Romis Ribeiro de Faissol Attux
Data da defesa:	12/12/2007
Programa de Pós-Graduação:	Engenharia Elétrica

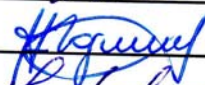
COMISSÃO JULGADORA - TESE DE MESTRADO

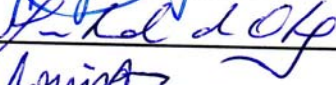
Candidato: Filipe Ieda Fazanaro

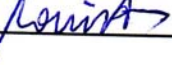
Data da Defesa: 12 de dezembro de 2007

Título da Tese: "Estudos e Implementações de Dinâmica Caótica Utilizando Dispositivos Analógicos Reconfiguráveis"

Prof. Dr. Marconi Kolm Madrid (Presidente): 

Prof. Dr. Reinaldo Gonçalves Nogueira: 

Prof. Dr. José Raimundo de Oliveira: 

Prof. Dr. Romis Ribeiro de Faissol Attux: 

Resumo

Fazanaro, F. I. (2007), Estudos e Implementações de Dinâmica Caótica utilizando Dispositivos Analógicos Reconfiguráveis, Dissertação de Mestrado, FEEC - UNICAMP.

Este trabalho teve como principal objetivo estudar a tecnologia baseada em dispositivos *Field Programmable Analog Arrays* (FPAAs) e identificar os benefícios quanto ao seu uso em aplicações de identificação de fenômenos inerentes aos sistemas dinâmicos não-lineares, tais como bifurcações e caos. Esses dispositivos permitem que diferentes tipos de circuitos possam ser implementados sem a necessidade de alteração da topologia do circuito, ou seja, existe a possibilidade de que os sistemas possam ser reconfigurados em tempo de execução à medida que novas alterações sejam necessárias. Com base na Teoria do Caos e na Teoria de Sistemas de Controle, foi implementado o sistema conhecido como Circuito de *Chua*, que serviu para demonstrar os ganhos que se podem obter com o uso da abordagem proposta quando aplicada ao estudo de sistemas dinâmicos operando no caos em relação às técnicas consideradas mais convencionais. Resultados obtidos pela análise de séries temporais de sinais adquiridos, comprovam a grande eficiência dessa abordagem quanto ao tempo de desenvolvimento e ao tempo para a obtenção dos resultados em comparação com implementações de modelos dinâmicos bastante conhecidos na literatura em relação às implementações dos mesmos em computadores digitais.

Palavras-chave: Dinâmica Caótica, *Field Programmable Analog Array*, Implementação em *Hardware* Reconfigurável, Séries Temporais.

Abstract

Fazanaro, F. I. (2007), Studies and implementations of chaotic dynamics using reconfigurable analog devices, Master's Thesis, FEEC - UNICAMP.

This work had as main objective to study the technology based on Field Programmable Analog Arrays (FPAAs) devices and to identify the benefits to use these devices in applications of identification of inherent phenomena to the nonlinear dynamic systems as bifurcations and chaos. These devices allow that different types of circuits can be implemented without the necessity of alteration of the topology of the circuit, that is, the systems implemented in the FPAA can be reconfigured in execution when new alterations are necessary. On the basis of the Chaos Theory and in the Control Systems Theory, was implemented the system known as Chua's Circuit which served to demonstrate the profits that can be gotten with the use of the boarding proposal when applied to the study of dynamic systems operating in chaos in relation to the considered techniques conventional. Gotten results, for the analysis of time series of acquired signals, prove the great efficiency of this boarding in the time of development and the time for obtain the results when comparing implementations of dynamic models sufficiently known in literature in relation with the implementations of the same ones in digital computers.

Keywords: Chaotic Dynamic, Field Programmable Analog Array, Hardware Reconfigurable Implementation, Time-Series

Agradecimentos

Agradeço às pessoas que de alguma forma contribuíram para a realização deste trabalho. Em especial, quero agradecer:

Ao meu orientador...

... Professor Dr. Marconi Kolm Madrid, pela excelente orientação, amizade, críticas e sugestões, incentivo, companheirismo e por acreditar na minha capacidade intelectual e no meu trabalho.

Aos Professores...

... Dr. José Raimundo de Oliveira e Dr. José Antenor Pomílio por cederem alguns dos equipamentos e placas utilizadas ao longo desse trabalho;

... Dr. José Raimundo de Oliveira, Dr. Reinaldo Gonçalves Nogueira e Dr. Romis Ribeiro de Faissol Attux por aceitarem participar da banca de avaliação.

Aos meus pais...

... Antonio Carlos e Maria Elisa pelo apoio, pelo incentivo e pela paciência.

À minha namorada...

... Mirela pelo carinho, pelo amor e por todo o tempo em que estamos juntos.

Aos meus tios e avós...

... Zaíra e Eduardo (*in memoriam*), João José e Cida, Lucy e José Ieda (*in memoriam*), Irineu Fazanaro, por acreditarem em mim e por me incentivarem principalmente nas horas difíceis.

Aos meus amigos

... Luiz Eduardo Guardia Filho e sua esposa Priscila e Maurício Soares Leão pela amizade e pelos constantes apoios e incentivos;

... André Luiz Delai pela amizade, pelas conversas e idéias e, principalmente, por me ensinar a jogar Half-Life 2:Deathmatch;

... Diogo Coutinho Soriano, pelas idéias, pelos incentivos, pelo vasto conhecimento em quadri-nhos e por me mostrar que, em pesquisa, ser teimoso ajuda;

... José Augusto Pires, Alexandre Montoya Bueloni e Letícia Maria Usberti por todos os bons momentos que passamos juntos, pelos constantes incentivos e pelo carinho que vocês tem por mim.

Aos colegas do LSMR...

... Héder Garcia, Fabrício Nicolato, Mário Jungbeck e outros mais que realizam e realizaram trabalhos no LSMR.

Aos ex-integrantes da República Winstom...

... Roger e Leonardo Novaes do Nascimento e Marcelo Suzart de Pádua, pelo tempo em que moramos juntos, pelas conversas e pelas horas que passamos jogando, em rede, Counter Strike (1.6 e Source) e Diablo 2.

Aos integrantes do Clube dos 13...

... Roger e Leonardo Novaes do Nascimento, Marcelo Suzart de Pádua, Rubens Hideo Korogui, Guilherme Cincoetti, Rafael de Castro Martins e todos os outros, pela amizade e pelo incentivo.

Aos funcionários...

... da FEEC pela ajuda em vários momentos e aos responsáveis pela base de periódicos da Biblioteca da Área de Engenharia, em especial à Juliana Ravaschio, por me ajudar na aquisição de diversos artigos, importantes para o desenvolvimento deste trabalho de mestrado.

Às bandas...

... Metallica, KoRn, Incubus, Deftones, Breaking Benjamin, U2, Papa Roach, Limp Bizkit e Skank por produzirem música de qualidade que embalsamaram esse trabalho.

À CAPES...

... pelo apoio financeiro.

*Para os meus pais e familiares, minha namorada Mirela
e em memória daqueles que nos deixaram.*

“O desenvolvimento de componentes analógicos, tradicionalmente, foi realizado em torno de “truques” em baixo nível de abstração (arte ou magia negra?) que envolvem *layout* e seleção de parâmetros de transistores e, dessa maneira, tornando virtualmente impossível a utilização de altos níveis de abstração (...)”

Adaptado de Carloni *et al.* (2002)

“Não existe nenhum caminho lógico para o descobrimento das leis elementares. O único caminho é o da intuição”.

Albert Einstein *em* (Rohden, 2007)

“Eu penso 99 vezes e nada descubro. Deixo de pensar e eis que a verdade é revelada”.

Albert Einstein *em* (Rohden, 2007)

Sumário

Resumo	iv
<i>Abstract</i>	v
Agradecimentos	vi
Dedicatória	ix
Epígrafe	x
Sumário	xi
Lista de Abreviaturas	xiv
Lista de Figuras	xvi
Lista de Símbolos	xix
Lista de Tabelas	xx
1 Introdução	1
1.1 Histórico do LSMR	4
1.2 Motivações e Objetivos	6
1.3 Organização deste Trabalho	8

2	Introdução ao FPAA	10
2.1	Estrutura do Dispositivo Utilizado	11
2.1.1	Descrição dos Blocos Analógicos Configuráveis	13
2.2	Tecnologia de Fabricação	15
2.3	Princípios da Reconfiguração Dinâmica	17
2.3.1	Sobre os Protocolos de Configuração	17
3	Comportamento Caótico em Sistemas Dinâmicos	20
3.1	Considerações Iniciais	20
3.2	Resposta Temporal	21
3.3	Plano de Fase e Espaço de Estados	23
3.4	Mapas de <i>Poincaré</i>	25
3.5	Diagramas de Bifurcação	28
3.6	Expoentes de <i>Lyapunov</i>	30
3.6.1	Conceitos Iniciais	30
3.6.2	Aplicação aos Modelos Dinâmicos	34
4	Séries Temporais	38
4.1	Considerações Iniciais	38
4.2	Séries Temporais	38
4.3	Reconstrução do Atrator	39
4.4	Obtenção dos Expoentes de <i>Lyapunov</i>	42
5	Ambiente de Desenvolvimento	45
5.1	Considerações Iniciais	45
5.2	Sobre o SIMULINK	45
5.3	As <i>System-Functions</i>	46

5.4	XPC TARGET	47
6	Resultados Experimentais	50
6.1	Considerações Iniciais	50
6.2	Análises Teóricas e Implementações Computacionais	50
6.3	Procedimentos Experimentais	54
6.3.1	Considerações sobre a Implementação em <i>Hardware</i>	54
6.3.2	Descrição da Bancada e Resultados Experimentais	56
6.3.3	Reconstrução do Atrator	59
6.4	Análises Quantitativas dos Resultados	60
7	Conclusões e Perspectivas Futuras	64
7.1	Conclusões	64
7.2	Perspectivas Futuras	65
A	Algoritmo do Máximo Expoente	66
A.1	Considerações Iniciais	66
A.2	Metodologia do Máximo Expoente	66
	Referências Bibliográficas	70

Lista de Abreviaturas

ADC *Analog-to-Digital Conversor;*

ASIC *Application-Specific Integrated Circuit;*

CAB *Configurable Analog Block;*

CAM *Configurable Analog Module;*

CNN *Cellular Neural Network;*

E/S *Entrada/Saída;*

DEP *Densidade Espectral de Potência;*

EEPROM *Electrically-Erasable Programmable Read-Only Memory;*

EPROM *Erasable Programmable Read-Only Memory;*

FPAA *Field Programmable Analog Array;*

FPGA *Field Programmable Gate Array;*

GUI *Graphical User Interface;*

I/O *Input/Output;*

ISA *Industry Standard Architecture;*

LE *Logic Element;*

LSMR *Laboratório de Sistemas Modulares Robóticos;*

LUT *Look-Up Table;*

MSB *Most Significant Bit;*

PSoC *Programmable System on-Chip;*

PCAs *Programmable Capacitor Arrays;*

PCI *Placa de Circuito Impresso;*

PWM *Pulse Width Modulation;*

RAM *Random Access Memory;*

SAR *Successive Approximation Register;*

SPI *Serial Protocol Interface;*

SRAM *Static Random Access Memory;*

VLSI *Very-Large-Scale Integration;*

Lista de Figuras

1.1	Primeiras estruturas robóticas desenvolvidas no LSMR.	5
1.2	Pêndulo acionado durante experimentação no LSMR.	5
1.3	Estrutura do robô COBRA.	6
1.4	Diagrama ilustrativo do funcionamento do pêndulo com restrições ao movimento.	7
1.5	<i>Kit</i> de desenvolvimento EXCALIBUR.	7
1.6	Placa de desenvolvimento baseada em tecnologia FPGA.	8
2.1	Estrutura interna simplificada do FPAA utilizado.	12
2.2	Ambiente de desenvolvimento do <i>software</i> ANADIGMDESIGNER® EDA.	12
2.3	Placa de desenvolvimento empregada nos estudos.	13
2.4	Estrutura interna de um CAB.	14
2.5	Ilustração da técnica de chaveamento de capacitores.	15
2.6	Possibilidade de obtenção de resistência equivalente negativa pela utilização da técnica de chaveamento de capacitores.	16
2.7	Seleção dos parâmetros da CAM habilitadas para a reconfiguração.	18
2.8	Esquema do procedimento de reconfiguração dinâmica.	19
3.1	Comparação de complexidade da resposta temporal entre um sinal caótico e um sinal periódico.	22
3.2	Verificação da sensibilidade às condições iniciais em um sistema caótico.	22

3.3	Atrator estranho obtido a partir do sistema de Lorenz (3.1).	23
3.4	Atratores estranhos obtidos a partir do sistema de <i>Chen</i> (3.2).	24
3.5	Atrator estranho obtido a partir do sistema de <i>Rössler</i> (3.3).	25
3.6	Mapa de <i>Poincaré</i> para a situação de um sinal periódico.	26
3.7	Mapa de <i>Poincaré</i> para a situação de um sinal caótico.	26
3.8	Seções de <i>Poincaré</i> , plano (x, y) , para sistemas com comportamento caótico. . .	28
3.9	Diagrama de bifurcação para o sistema de <i>Rössler</i> (3.4).	29
3.10	Evolução do elemento de volume particularizado para um caso bidimensional. . .	31
3.11	Ilustração do conceito de divergência das trajetórias para um sistema dinâmico. .	32
3.12	Cálculo dos expoentes de <i>Lyapunov</i> ao longo de uma trajetória fiducial.	33
3.13	Espectro de <i>Lyapunov</i> para o sistema de <i>Lorenz</i> (3.15).	36
4.1	Atratores reconstruídos a partir da série temporal para o modelo de <i>Lorenz</i> (3.1). .	41
4.2	Ilustração do procedimento de simulação das trajetórias fiducial e secundária. . .	43
4.3	Ilustração do procedimento de simulação das trajetórias fiducial e secundária (continuação).	44
4.4	Ilustração do procedimento de escolha dos pontos durante o cálculo do maior expoente de <i>Lyapunov</i>	44
5.1	Modelo (6.5) implementado utilizando-se <i>s-function</i> e a máscara desenvolvida para a alteração dos parâmetros do modelo.	47
5.2	Diagrama do funcionamento do xPC TARGET utilizando o protocolo de comu- nicação TCP/IP.	48
5.3	Placa de aquisição de dados utilizada.	49
6.1	Representação do Circuito de <i>Chua</i> em sua forma original e a característica tensão $\times$ corrente mais usual do Diodo de <i>Chua</i> , linear por partes.	51
6.2	Atrator e resposta temporal obtidos pela implementação do modelo (6.5) pela utilização da ferramenta <i>s-function</i>	53

6.3	Circuito de <i>Chua</i> (6.5) implementado no FPAA.	55
6.4	Ilustração da configuração da bancada experimental.	56
6.5	Resultados qualitativos, obtidos via osciloscópio digital com retenção de 6.95s, para o modelo do Sistema de <i>Chua</i> implementado no FPAA.	58
6.6	Resultados qualitativos, obtidos via osciloscópio digital com retenção de 6.95s, para o modelo do Sistema de <i>Chua</i> implementado no FPAA (continuação). . . .	59
6.7	Resposta temporal das variáveis de estado x e y e atrator para o Circuito de <i>Chua</i> representado pelo sistema (6.5).	59
6.8	Atratores reconstruídos a partir da série temporal.	60
6.9	Alteração do parâmetro EVOLV.	63
6.10	Alteração do parâmetro SCALMX.	63

Lista de Símbolos

- λ_i : Valor médio do expoente de *Lyapunov*. Por exemplo, para sistemas operando em caos com 3 variáveis de estado, quando $i = 1$ tem-se o expoente positivo, $i = 2$ representa o expoente nulo e, para $i = 3$, tem-se o expoente negativo;
- $L(t_i)$: Distância inicial entre a trajetória fiducial e a trajetória secundária/perturbada no instante de tempo t_i . Essa distância é utilizada no cálculo dos expoentes de *Lyapunov*;
- $L'(t_i)$: Distância entre a trajetória fiducial e a trajetória secundária/perturbada no instante de tempo t_{i+1} . Essa distância é utilizada no cálculo dos expoentes de *Lyapunov*;
- d_A : Dimensão do atrator original. Usualmente pode ser obtida a partir do cálculo dos expoentes de *Lyapunov*;
- d_E : Dimensão na qual o atrator será reconstruído;
- $x(t_j)$: Valor da variável de estado x adquirida no instante de tempo t_j ;
- X : Vetor série temporal referente à variável de estado x ;
- $z(j)$: Vetor que representa a j -ésima linha da matriz de trajetória do espaço de estados reconstruído;
- Z : Matriz de trajetória do espaço de estados reconstruído;
- τ : Tempo de atraso de *Takens*;

Lista de Tabelas

3.1	Combinações possíveis dos sinais dos expoentes de <i>Lyapunov</i> para um sistema dinâmico de 4 ^a ordem.	35
6.1	Parâmetros para as CAMs utilizadas na implementação do modelo de <i>Chua</i> . . .	55
6.2	Expoentes de <i>Lyapunov</i> obtidos a partir das séries experimentais.	62

Capítulo 1

Introdução

O estudo da dinâmica de determinados sistemas permite que ações sejam realizadas sobre os mesmos com o propósito de analisar, compreender, diagnosticar, prever e controlar sinais específicos. Além disso, perante essas necessidades, torna-se importante e necessária a classificação dos sistemas para que as ferramentas de análise a serem utilizadas sejam escolhidas de forma a melhor se adequarem ao seu comportamento, seja ele de dinâmica periódica ou caótica, evitando que as conclusões realizadas distorçam a realidade (Parker e Chua, 1989; Kinsner, 2006).

Dentre os comportamentos dinâmicos apresentados, destacam-se o de sistemas ditos caóticos, os quais, usualmente, podem ser caracterizados pela extrema sensibilidade às condições iniciais de suas variáveis de estado. Esse fato é evidenciado na situação em que o modelo dinâmico apresenta um grande número de variáveis de estado, quando a previsão do comportamento do sistema dentro de um longo intervalo de tempo passa a ser uma tarefa bastante complexa e que demanda, além de grande capacidade computacional, muito tempo para a obtenção das conclusões sobre os resultados (Nogueira, 2001). Por exemplo, em sistemas robóticos conservativos, mais especificamente, na sub-área responsável pelo estudo de robôs móveis do tipo bípedes, um dos grandes desafios reside nas dificuldades relacionadas em obter-se o modelo matemático e, conseqüentemente, realizar análises globais de estabilidade. Contudo, trabalhos recentes presentes na literatura especializada exibem progressos nessa área e verificam a existência de caos, estudado quantitativamente via expoentes e dimensão de *Lyapunov*, associado ao movimento do robô (Kaygisiz *et al.*, 2006).

Outro exemplo bastante importante é a atmosfera terrestre, que consiste em um sis-

tema que possui uma variedade considerável de processos físicos¹ representados por diversas variáveis de estado, sendo que muitos dos processos são pouco conhecidos individualmente, embora a sua interação determine o clima, seja em escala regional ou global. Fica evidente que a dificuldade em estudar a atmosfera terrestre e realizar previsões climáticas surge em virtude de que essas iterações possuem muitos mecanismos de realimentação que agem amplificando ou amortecendo pequenas perturbações iniciais (Vianello e Alves, 1991). Então, como o sistema climático terrestre é altamente não-linear, pode ser considerado um sistema caótico (Corti *et al.*, 1999; Orrel, 2002) e, por essa razão, torna-se um verdadeiro desafio a uma completa descrição quantitativa. A característica caótica da atmosfera terrestre é utilizada pela meteorologia para explicar a dificuldade em realizar previsões comportamentais climáticas para períodos de tempo muito longos (por exemplo, mais de 15 dias) e a necessidade de utilização de grande capacidade computacional para a realização de análises sobre o sistema (Guardia Filho, 2007).

Além da robótica e da meteorologia, muitas outras áreas científicas beneficiam-se de trabalhos publicados na literatura sob a temática “Teoria do Caos”. Em ciências sociais, empregam-se conceitos dessa teoria para estudar o comportamento humano (Guess e Sailor, 1993). Em medicina, são utilizados os princípios da dinâmica não-linear para melhorar os cuidados de pessoas com diabetes (Kroll, 1999; Holt, 2002), para o estudo de células cardíacas (Cai *et al.*, 1993; Cai *et al.*, 1994; Winslow *et al.*, 1995) e no estudo de sinais de EEG (Principe e Lo, 1991; Quiroga, 1998). Em engenharia, pode-se citar o estudo do caos realizado em sistemas de comunicação (Huang *et al.*, 2005) e em circuitos elétricos não-lineares (Carroll e Pecora, 1991; Kiers *et al.*, 2004; Femat *et al.*, 2005).

Em se tratando de circuitos elétricos não-lineares, o estudo do comportamento caótico pode ser realizado, de maneira razoavelmente simples, em circuitos construídos a partir de componentes discretos básicos tais como resistores, capacitores, indutores, amplificadores operacionais e diodos, e projetados para resolverem sistemas de equações diferenciais. Um caso particular de circuito elétrico não-linear bastante importante neste contexto é o Circuito de *Chua*, o qual ficou conhecido como um paradigma referencial no estudo de sistemas caóticos em virtude de ser possível observar uma vasta família de atratores estranhos, bem como bifurcações e rotas para o caos, pelo simples ajuste dos parâmetros dos seus componentes (Chua, 1993; Shil’nikov, 1993; Liu *et al.*, 2007).

Devido a essas características, o Circuito de *Chua* tem sido tema de pesquisa em

¹Podem-se citar, dentre outros, a radiação solar, as propriedades óticas da atmosfera, a latitude, a umidade relativa, gradiente de pressão e de temperatura.

diversos trabalhos científicos. Arena, Baglio, Fortuna e Manganaro (1995) apresentam a análise teórica para comprovar o comportamento caótico intrínseco desse circuito. Além disso, mostrou-se uma metodologia de análise baseada na Teoria de Redes Neurais (*Cellular Neural Networks* - CNNs). Uma rede neural foi, então, implementada através do uso de componentes discretos tais como resistores, capacitores e amplificadores operacionais e, conseqüentemente, pode ser implementada em qualquer outro tipo de tecnologia emergente que simula, emula ou realiza tais componentes.

As CNNs (Haykin, 2000; Islama e Muraseb, 2005; Arena *et al.*, 2005), no contexto de computação analógica de dados, são os exemplos mais famosos de processos analógicos que podem ser utilizadas em muitas aplicações, desde processamento de imagens até emulação de sistemas complexos. Contudo, a computação analógica possui um grande problema relacionado à enorme dificuldade em realizar programações (Caponetto *et al.*, 2005) e, conseqüentemente, novas tecnologias estão sendo estudadas para facilitar o desenvolvimento dessas aplicações. Uma tecnologia bastante promissora no contexto de desenvolvimento analógico de sistemas é a dos *Field Programmable Analog Arrays* (FPAAs²) (Andrade Jr. *et al.*, 2005) ainda em estágios iniciais de desenvolvimento em comparação ao que ocorre com os Dispositivos Lógicos Programáveis (PLD) e os *Field Programmable Gate Arrays* (FPGAs), já bastante desenvolvidos (Brown e Rose, 1996) e utilizados em muitas aplicações de processos de alta tecnologia tais como processamento de imagens (Kolodko e Vlacic, 2003), inteligência artificial (Tang e Yip, 2004), e, também, em sistemas robóticos conservativos (Hoshi *et al.*, 2004).

Os FPAAs³ são dispositivos analógicos reconfiguráveis, ou seja, permitem a reconfiguração de partes ou de todo o circuito implementado, tal como ocorre nos FPGAs (Guardia Filho, 2005). Justamente pelo fato de serem dispositivos analógicos, os FPAAs podem ser utilizados nos estudos e implementações de modelos dinâmicos de tempo contínuo que podem operar em caos (Caponetto *et al.*, 2005), em conjunto com ou em substituição aos circuitos eletrônicos compostos por componentes discretos. As vantagens da utilização dos FPAAs são bem conhecidas pela possibilidade de realizar-se modificações no circuito em tempo de execução e, além disso, possibilitar obtenção dos resultados com grande precisão (Kiers *et al.*, 2004), e com muita rapidez em comparação aos tempos envolvidos na implementação de modelos equivalentes em computadores digitais.

Com a implementação dos modelos em *hardware*, passa a ser interessante realizar

²Existem outras tecnologias baseadas em dispositivos analógicos reconfiguráveis tais como os PSoC da CYPRESS cujo grande diferencial é possuir um módulo digital encapsulado e trabalhando em conjunto com os módulos analógicos. Maiores informações em <http://www.cypress.com>.

³Maiores detalhes sobre essa tecnologia serão apresentados no Capítulo 2.

aquisições dos sinais provenientes dos dispositivos FPAAs objetivando-se analisar, qualitativa e quantitativamente, tais modelos para comprovação, através de ferramentas específicas de análise comportamental, como, por exemplo, periodicidade e caos através da utilização de planos e espaços de fase, planos de *Poincaré* e expoentes de *Lyapunov*. Assim, tem-se melhores condições para realizar provisões sobre comportamentos futuros dos sistemas sujeitos a variações de parâmetros, definir faixas de operações seguras e sem instabilidades, ou simplesmente conhecer com maior detalhamento suas possibilidades de operação. Geralmente, essa aquisição de dados é realizada de forma que se obtenham séries temporais de um ou mais sinais do sistema, sinais que representam as suas variáveis de estado. Conseqüentemente, princípios teóricos da teoria de séries temporais tornam-se necessários nesse contexto (Abarbanel, 1996; Sprott, 2003).

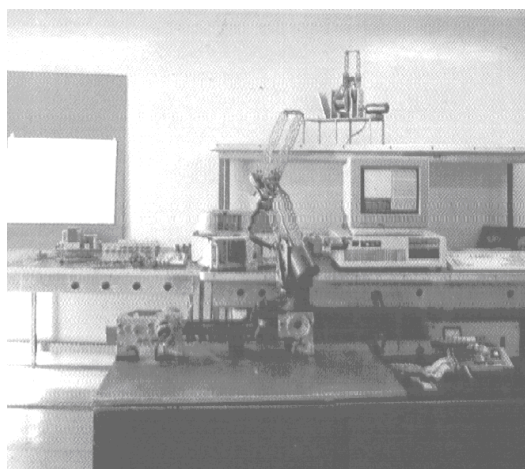
Com o que foi apresentado, ao longo desse capítulo será contextualizado o ambiente no qual esse trabalho de dissertação foi desenvolvido e serão expostos seus objetivos e a sua organização com o intuito de motivar a leitura e, dessa forma, oferecendo maior facilidade para a sua compreensão.

1.1 Histórico do LSMR

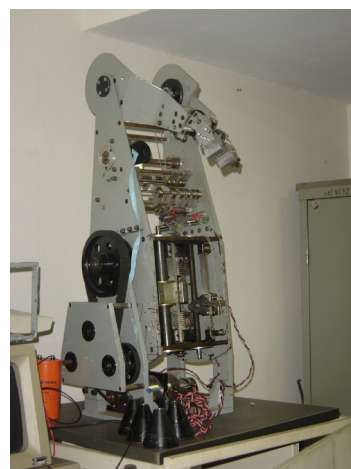
O Laboratório de Sistemas Modulares Robóticos (LSMR) foi criado há cerca de 23 anos, e desde então muitas linhas de pesquisa vêm sendo desenvolvidas. Dentre os temas, podem-se citar o estudo de estruturas robóticas, plataformas de controle baseadas em *hardware* reconfiguráveis e técnicas de análise do comportamento caótico.

No LSMR, foram projetados e construídos 4 protótipos de sistemas robóticos. O primeiro, com 4 graus de liberdade, denominado JECA I, Figura 1.1(a), possuía todo o sistema de controle implementado utilizando, por razões tecnológicas, microprocessadores de 8 *bits* sendo que, para o controle e acionamento das juntas, foi utilizada uma arquitetura monoprocessada (Madrid, 1988). Já o segundo, um robô construído com 5 graus de liberdade mais garra de 2 dedos denominado JECA II, Figura 1.1(b), teve todo o seu sistema de controle projetado e implementado utilizando uma arquitetura paralela microprocessada (mestre/escravos) (Madrid, 1994). Nesse trabalho foram utilizados microprocessadores de 8 *bits*, sendo que cada junta tinha o seu próprio controle de acionamento.

O terceiro módulo mecânico desenvolvido foi um pêndulo acionado composto de 2 eixos e engrenagens do tipo polias e correias sincronizadas, Figura 1.2. O pêndulo foi desenvolvido durante o trabalho realizado por de Souza (2000), quando foi proposta uma técnica



(a) JECA I



(b) JECA II

Figura 1.1: Primeiras estruturas robóticas desenvolvidas no LSMR.

que empregava algoritmos genéticos e teoria de conjuntos nebulosos integrados, objetivando o desenvolvimento automático de controladores de alto desempenho para servo-mecanismos tipo elo-acionado ou módulos de junta robótica. Todos os algoritmos foram implementados em linguagem C, e os procedimentos de aquisição de leitura do *encoder* e de geração do sinal de PWM (*Pulse Width Modulation*) foram realizados por *hardware* externo e enviados a um PC através de uma placa de comunicação de dados baseada no barramento ISA (*Industry Standard Architecture*). Utilizando esse mesmo protótipo, Jungbeck (2001) propôs técnicas de controle baseadas em redes neurais. Mais recentemente, o pêndulo foi utilizado na implementação de controladores evoluídos utilizando-se técnicas de algoritmos e *hardware* evolutivos (Delai, 2008).

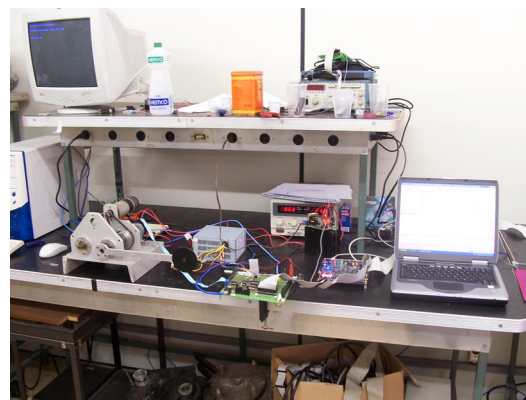
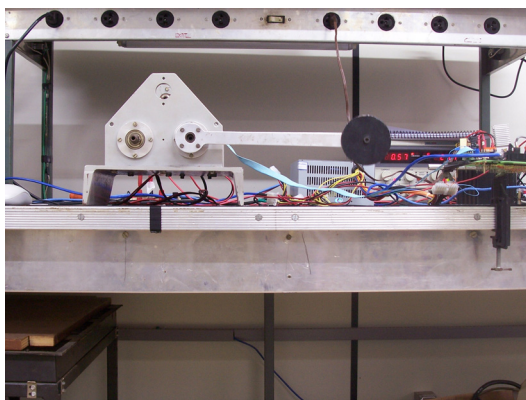


Figura 1.2: Pêndulo acionado durante experimentação no LSMR.

O quarto e mais recente robô desenvolvido no LSMR foi denominado COBRA, Figura 1.3. Essa estrutura robótica foi desenvolvida durante o trabalho de Nicolato (2007) com

o intuito de realizar implementações práticas da metodologia de controle baseada em buscas heurísticas aplicadas a robôs de estrutura redundante (Spong e Vidyasagar, 1989).

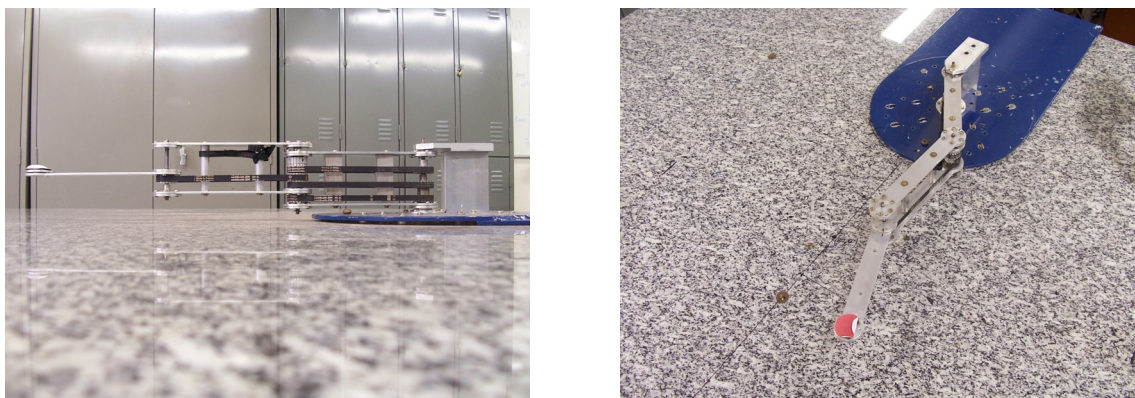


Figura 1.3: Estrutura do robô COBRA.

No contexto de estudos de plataformas de desenvolvimento baseadas em *hardware* reconfigurável, foram realizados os trabalhos de Nicolato (2002) e de Guardia Filho (2005), os quais são mais detalhados na Seção 1.2.

Outra linha de pesquisa desenvolvida no LSMR diz respeito ao estudo de técnicas de análise comportamental de sistemas não-lineares. O comportamento caótico desperta particular interesse, visto que só é possível identificá-lo em sistemas cuja dinâmica seja não-linear, como os que ocorrem no modelamento de robôs (Spong e Vidyasagar, 1989). Como o pêndulo é uma estrutura que pode ser interpretada como um elo robótico, Nogueira (2001) desenvolveu o seu trabalho a partir de um problema bastante importante em robótica, que é o caso da análise de restrições ao movimento, Figura 1.4. A metodologia proposta baseou-se em simulações computacionais do modelo dinâmico do sistema, implementado com o auxílio do ambiente de desenvolvimento MATLAB. Além disso, como tratou-se de análises de um sistema dinâmico com comportamento caótico, foi possível a verificação de diversas dificuldades para a realização das simulações tais como tempo elevado para a obtenção de resultados.

1.2 Motivações e Objetivos

Um dos objetivos desse trabalho consistiu em trazer para o LSMR a tecnologia baseada em *Field Programmable Analog Arrays* (FPAA) e estudar o seu funcionamento e as possibilidades de sua utilização em contextos acadêmicos e industriais. O procedimento inicial é análogo ao que foi realizado em (Nicolato, 2002), quando foram realizados os primeiros estu-

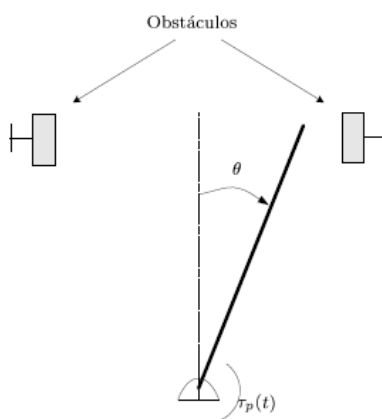


Figura 1.4: Diagrama ilustrativo do funcionamento do pêndulo com restrições ao movimento.

dos⁴ da tecnologia baseada em *Field Programmable Gate Arrays* (FPGA) através da utilização de um *kit* de desenvolvimento EXCALIBUR da ALTERA⁵, ilustrado na Figura 1.5. Os objetivos consistiam em analisar as possibilidades da utilização de FGPAs no âmbito da robótica.



Figura 1.5: *Kit* de desenvolvimento EXCALIBUR.

Dessa forma, pelas possibilidades envolvidas na utilização dos FPGAs em robótica, Guardia Filho (2005) desenvolveu todo o projeto de uma plataforma capaz de realizar o controle de máquinas robóticas em tempo real, Figura 1.6, utilizando técnicas de processamento paralelo. O núcleo dessa placa de circuito impresso (PCI) é um FPGA da família CYCLONE⁶. Foi previsto, também, que essa plataforma pudesse ser utilizada em todos os níveis hierárquicos de produção, sendo esses níveis constituídos por supervisão, tarefas, trajetória e servo-mecanismos,

⁴O estudo da tecnologia FPGA desenvolvido pelo autor é um dos primeiros realizados tanto no LSMR quanto na FEEC, UNICAMP.

⁵Maiores informações podem ser encontradas em <http://www.altera.com>.

⁶Maiores informações em <http://www.altera.com/products/devices/cyclone/cyc-index.jsp>.

todos envolvidos em plantas baseadas em controle automático.

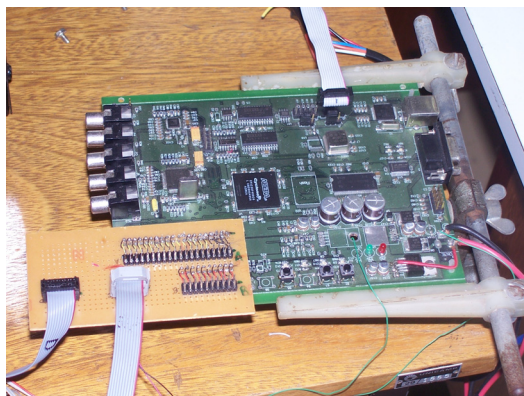


Figura 1.6: Placa de desenvolvimento baseada em tecnologia FPGA.

Portanto, com base nos estudos iniciais realizados durante esse trabalho de mestrado que está sendo apresentado para a tecnologia FPAA, espera-se que seja possível o desenvolvimento de uma plataforma híbrida formada por dispositivos FPGA e FPAA operando em conjunto e de maneira automatizada (Guardia Filho, 2007).

O segundo objetivo consistiu em realizar um estudo de caso para verificar as possibilidades de aplicações dos FPAAs em sistemas dinâmicos não-lineares e identificar as vantagens e as limitações de sua utilização. O estudo de caso realizado baseou-se em implementações de modelos matemáticos de sistemas dinâmicos, em especial, circuitos elétricos não-lineares bastante conhecidos na literatura (Chua, 1993; Shil'nikov, 1993; Kinsner, 2006). Finalmente, uma metodologia de análise quantitativa baseada no cálculo dos expoentes de *Lyapunov* é proposta a partir de séries temporais (Abarbanel, 1996; Sprott, 2003) formadas pela aquisição dos sinais do modelo implementado em FPAA.

1.3 Organização deste Trabalho

Para auxiliar durante a leitura deste trabalho de dissertação, o texto foi organizado da seguinte maneira:

Capítulo 1: capítulo atual em que contextualiza-se esse trabalho;

Capítulo 2: são apresentadas características do dispositivo FPAA utilizado no trabalho, bem como o *kit* de desenvolvimento disponível no LSMR e empregado nos experimentos realizados;

Capítulo 3: são apresentados conceitos importantes para o estudo de sistemas caóticos bem como discussões sobre os expoentes de *Lyapunov* e alguns modelos dinâmicos bastante estudados na literatura especializada;

Capítulo 4: discussão sobre as séries temporais, principais características e as justificativas sobre o seu estudo no contexto dessa dissertação;

Capítulo 5: nesse capítulo são apresentadas características dos ambientes de desenvolvimento MATLAB e SIMULINK utilizados para estudos dos modelos dinâmicos;

Capítulo 6: são apresentados os resultados experimentais obtidos;

Capítulo 7: são apresentadas as conclusões desse trabalho e perspectivas futuras.

Capítulo 2

Introdução ao FPAA

Field Programmable Analog Arrays (FPAAs) são dispositivos analógicos reconfiguráveis do tipo VLSI construídos a partir de amplificadores operacionais, capacitores, chaves e memórias SRAM, todos integrados em um mesmo encapsulamento (Baccigalupi e Liccardo, 2007). Os FPAAs permitem reconfiguração dinâmica não somente durante a fase de desenvolvimento de um determinado sistema mas também durante o seu funcionamento, ou seja, em tempo de execução (*on-the-fly*), possibilitando, ainda, a reconfiguração de parte e/ou de todo o sistema implementado, mantendo-se a sua topologia. Assim como os seus equivalentes digitais, os FPGAs (Brown e Rose, 1996; Guardia Filho, 2005), os FPAAs fornecem rápida prototipagem, grande flexibilidade, possibilidade de implementação de muitas configurações em um único dispositivo, além de baixos custos de desenvolvimento. Dessa maneira, incorporam propriedades interessantes para aplicações em controle e em sistemas de instrumentação (Andrade Jr. *et al.*, 2005).

Sob o tema instrumentação, Callegari *et al.* (2006) propõem uma metodologia de prototipagem rápida de circuitos de interfaceamento de sensores do tipo capacitivos (Merendino *et al.*, 2005) que explora a utilização de FPAAs no desenvolvimento de estágios analógicos de condicionamento de sinais. A metodologia apresentada objetiva a implementação de arquiteturas de amplificadores de carga em FPAA de tempo discreto, os quais são construídos utilizando-se técnicas de chaveamento de capacitores (Laknaur *et al.*, 2006). Além disso, também foi observado que a principal vantagem na utilização de FPAAs consiste na possibilidade de calibração do sistema via *software* aliada à capacidade de reconfiguração em tempo de execução. Os resultados obtidos pelos autores foram considerados bons quando comparados aos resultados decorrentes da utilização de outras arquiteturas, como as ASICs (Guardia Filho, 2005).

Devido à recente popularização de dispositivos analógicos programáveis, torna-se necessário o desenvolvimento de técnicas de testes de produção dos FPAAs que possibilitem maior capacidade e melhor utilização dos dispositivos, diminuição de custos de fabricação e de testes, além da baixa susceptibilidade às falhas dos sistemas (Caponetto *et al.*, 2005; Terry *et al.*, 2006; Baccigalupi e Liccardo, 2007). Dessa maneira, é conveniente analisar e testar cada um dos módulos funcionais que compõem um FPAA, tais como Blocos Analógicos Configuráveis, dispositivos de E/S e memórias de configuração, e, conseqüentemente, desenvolver metodologias específicas de testes (Andrade Jr. *et al.*, 2005).

2.1 Estrutura do Dispositivo Utilizado

Os FPAAs AN221E04 da ANADIGM (Anadigm, 2007) são constituídos por uma matriz 2×2 em que cada um dos elementos armazena um Bloco Analógico Reconfigurável (*Configurable Analog Block* - CAB). Os CABS estão imersos em uma rede de interconexões (*Interconnect Resources*) que possibilita a interligação desses elementos entre si e a interligação entre os CABS e todos os outros componentes que constituem o FPAA. Dentre esses componentes destacam-se as memórias de configuração do tipo SRAM, uma para cada CAB, específicas para o armazenamento dos dados de configuração de cada um dos blocos. Além das memórias, existe uma interface analógica de Entrada/Saída (E/S) composta por 4 células configuráveis de E/S (*Configurable I/O Cells*) e outras 2 células dedicadas exclusivamente para interface de saída. Na Figura 2.1 tem-se uma ilustração da estrutura interna do FPAA.

Os dispositivos AN221E04 possibilitam a implementação de Conversores Analógico-Digitais (Conversor AD) de 8 *bits*. Esse conversor pode comunicar-se com o exterior do *chip* através das células de saída. Caso seja utilizada uma das duas células dedicadas de saída, é necessário configurá-la para operar em modo digital. A saída de dados do Conversor AD é realizada como uma seqüência serial de 8 *bits*, sendo iniciada com o *bit* mais significativo (*Most Significant Bit* - MSB). Para a implementação do conversor, pode-se utilizar o Módulo Analógico Configurável (*Configurable Analog Module* - CAM) SAR-ADC disponível na biblioteca padrão do *software* de desenvolvimento ANADIGMDESIGNER[®] EDA, Figura 2.2.

Para a realização de todos os experimentos desenvolvidos durante esse trabalho, foi utilizado um *kit* de desenvolvimento da ANADIGM modelo AN221K04 - ANADIGMVORTEX, Figura 2.3, baseado no FPAA AN221E04. As principais características desse *kit* consistem em:

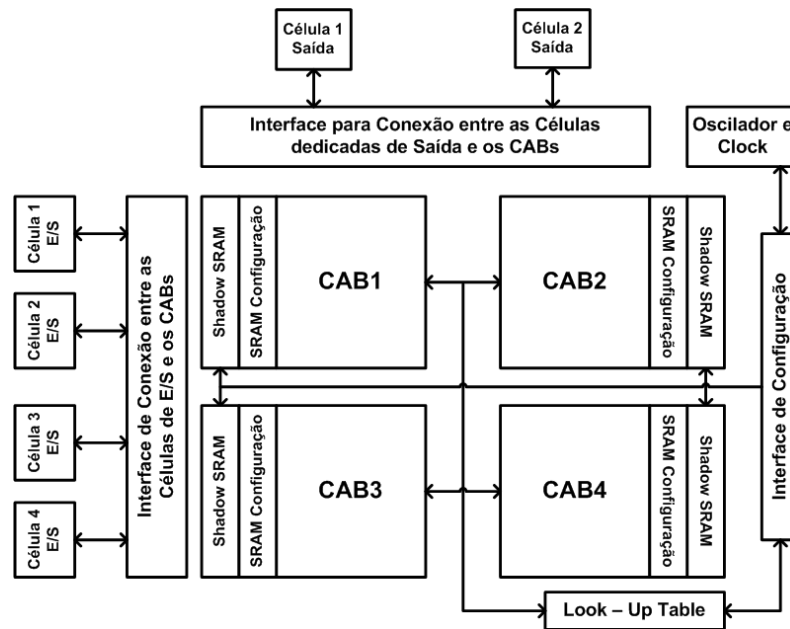
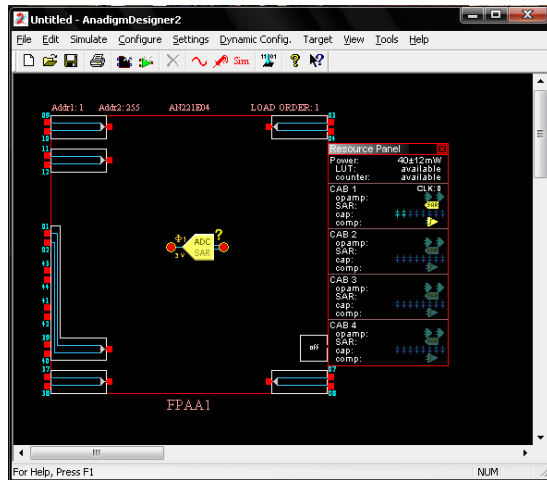
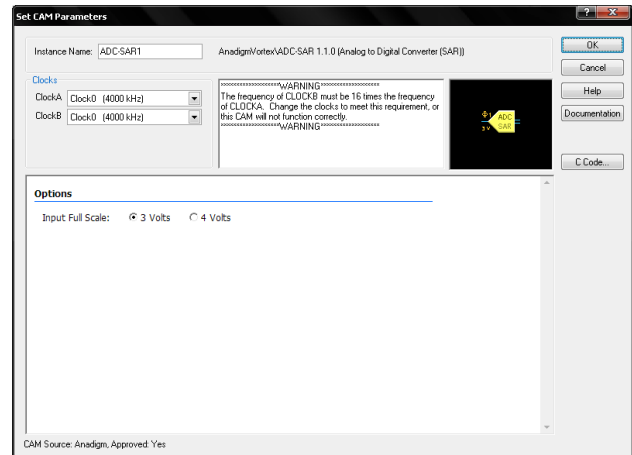


Figura 2.1: Estrutura interna simplificada do FPAA utilizado.



(a) CAM do Conversor AD.



(b) Janela de configuração do SAR-ADC.

Figura 2.2: Ambiente de desenvolvimento do *software* ANADIGMDESIGNER® EDA.

- Uma porta de comunicação serial RS232 que converte os dados enviados pelo *software* de desenvolvimento para a configuração do dispositivo FPAA pela interface SPI (*Serial Protocol Interface*);
- Pinos de E/S analógicos;
- 2 amplificadores operacionais que podem ser utilizados como *buffer*, como elevadores/a-

- tenuadores de tensão, para filtragem e/ou transformação de sinais diferenciais em *single*;
- d. Pequena área para montagem de circuitos externos e soquete específico para a inserção de memória tipo EPROM, o que possibilitaria a configuração de maneira automática quando a placa fosse inicializada.

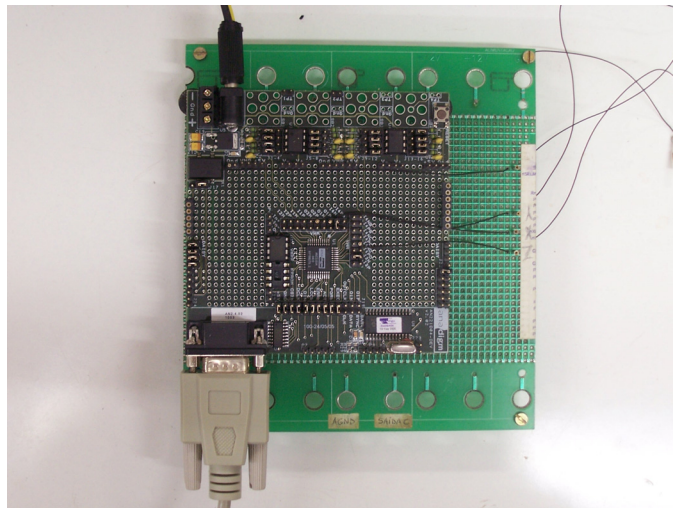


Figura 2.3: Placa de desenvolvimento empregada nos estudos.

O dispositivo AN221E04 pode trabalhar tanto com sinais de relógio externos (*external clock*) quanto com um relógio gerado a partir de um oscilador interno e um cristal externo. Nessa situação, o oscilador interno do FPAA automaticamente detecta o cristal externo e, em seguida, produz o sinal de relógio a ser utilizado tanto pela lógica de configuração quanto pelas partes analógicas do dispositivo. Este FPAA aceita cristais externos que operem na faixa de 12 MHz a 24 MHz, sendo que 16 MHz é considerado o valor ótimo (Anadigm, 2007).

2.1.1 Descrição dos Blocos Analógicos Configuráveis

Conforme apresentado na Figura 2.1, o dispositivo FPAA AN221E04 possui 4 Blocos Analógicos Configuráveis (CABS) onde são implementados todos os Módulos Analógicos Configuráveis (CAM), ou seja, os CABS são estruturas primárias equivalentes aos Elementos Lógicos (*Logic Elements* - LE) encontrados nos FPGAs (Guardia Filho, 2005). Na Figura 2.4 tem-se o diagrama simplificado de como os CABS são constituídos (Anadigm, 2007)

Na entrada de cada CAB existe uma matriz de chaves estáticas que recebe sinais de outros CABS, bem como sinais realimentados dos 2 amplificadores operacionais e do compara-

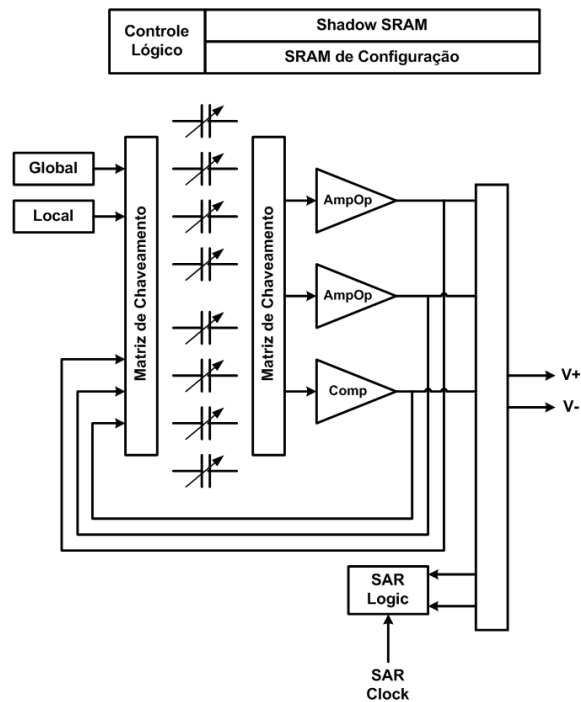


Figura 2.4: Estrutura interna de um CAB.

dor. A escolha de qual desses sinais será utilizado é feita através da memória de Configuração SRAM.

Ligada à essa matriz de chaves encontra-se um banco de 8 capacitores programáveis. Cada um desses capacitores representa um banco capacitivo formado por muitos capacitores de pequeno valor e idênticos uns aos outros. Para o correto funcionamento do FPAA, o valor relativo entre os pequenos capacitores varia entre 0 e 255 unidades de capacitância dentro de uma tolerância de 0.1%. O chaveamento desses capacitores é controlado pela memória de Configuração SRAM (Anadigm, 2007).

Dentro do CAB existe uma segunda matriz de chaves responsável por estabelecer a topologia do circuito a ser implementado no FPAA e realizar conexões apropriadas. No núcleo do CAB existem dois amplificadores operacionais e um comparador cujas saídas podem ser enviadas tanto para outros CABS como realimentadas para a primeira matriz de chaves, possibilitando, dessa maneira, a construção de circuitos e malhas fechadas de controle.

Outra estrutura presente no CAB é o Registrador de Aproximações Sucessivas (*Successive Approximation Register* - SAR), o qual, quando habilitado, utiliza o comparador para a implementação de um Conversor Analógico-Digital (*Analog-to-Digital Converter* - ADC) de 8 bits. Realimentando a saída do SAR-ADC para o próprio CAB ou para a *Look-Up Table*

(LUT), podem-se criar funções analógicas não-lineares tais como multiplicadores de tensão, linearização e controle automático de ganho.

2.2 Tecnologia de Fabricação

Internamente aos CABS, todo o tratamento de sinais é realizado através de circuitos baseados em chaveamento de capacitores, processo inerente ao funcionamento do FPAA AN221E04. A técnica baseada em chaveamento de capacitores consiste em implementar resistências equivalentes abrindo-se e fechando-se as entradas e as saídas do capacitor em determinada frequência, da qual conseqüentemente, o valor da resistência equivalente obtida é dependente (Terry *et al.*, 2006; Baccigalupi e Liccardo, 2007). Na Figura 2.5 é ilustrada a metodologia de chaveamento de capacitor.

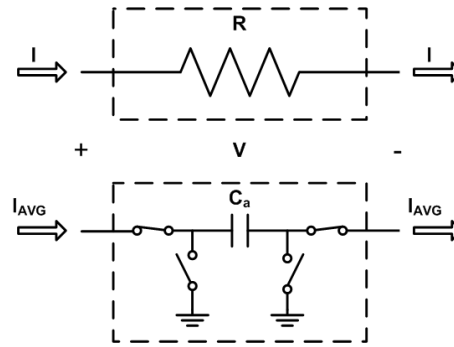


Figura 2.5: Ilustração da técnica de chaveamento de capacitores.

Pode-se comprovar matematicamente a metodologia de chaveamento em questão. Supondo que R e C_a possuam a mesma diferença de potencial V aplicada em seus terminais, como mostrado na Figura 2.5, ou seja,

$$R = \frac{V}{I} \quad (2.1)$$

$$I_{AVG} = \frac{C_a V}{T} \quad (2.2)$$

onde I e I_{AVG} correspondem a corrente que passa pelo resistor e a corrente média que passa pelo capacitor, respectivamente, e C_a é a carga no capacitor ao longo do tempo T considerado. Substituindo (2.2) em (2.1), tem-se

$$R_{eq} = \frac{1}{fC_a} \quad (2.3)$$

onde $f = T^{-1}$ corresponde à frequência com que as chaves da entrada e da saída do capacitor são abertas e fechadas e R_{eq} é a resistência equivalente do sistema.

Uma característica importante dessa técnica é que, dependendo da disposição das chaves, é possível obterem-se valores negativos de resistência, Figura 2.6,

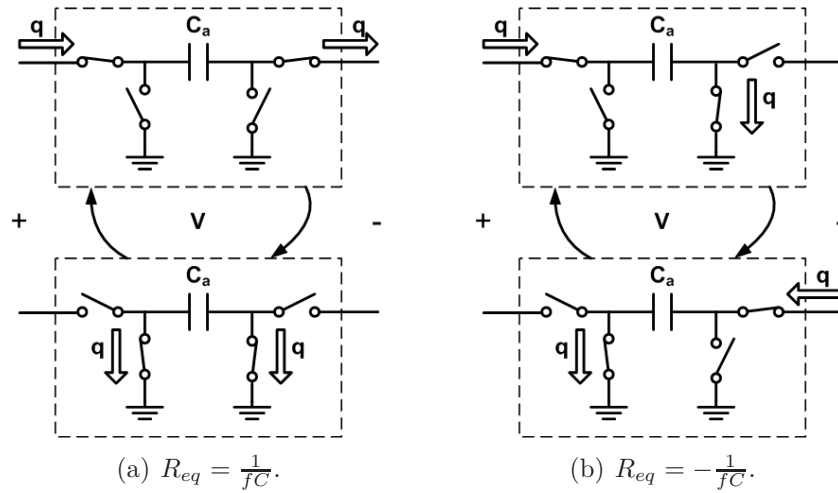


Figura 2.6: Possibilidade de obtenção de resistência equivalente negativa pela utilização da técnica de chaveamento de capacitores.

Do ponto de vista macroscópico, o valor da resistência equivalente é controlado pela frequência de chaveamento, a qual é limitada pelo *clock* do sistema. No caso do *kit*, essa frequência possui valor máximo de 4 MHz. Outra maneira de alterar o valor da resistência seria manipular o valor da capacitância propriamente dita (Terry *et al.*, 2006), algo que poderia ser realizado somente durante o processo de fabricação do *chip* FPAA.

Contudo, apesar das possibilidades envolvidas, a técnica de chaveamento de capacitores possui algumas dificuldades de implementação. É intrínseco ao método que o processamento dos sinais é realizado discretamente e, dessa forma, devem ser adotados filtros para correções de problemas (Terry *et al.*, 2006). Baccigalupi e Liccardo (2007) desenvolveram um trabalho com o objetivo de realizar testes e analisar outros problemas que podem ocorrer, tais como capacitância parasita. A técnica proposta também engloba métodos para tentativa de correções dessas capacitâncias sendo os resultados obtidos pelos autores considerados bastante promissores quando em comparação com outras metodologias.

2.3 Princípios da Reconfiguração Dinâmica

A reconfiguração dinâmica é uma característica que torna o FPAA uma ferramenta bastante importante durante o desenvolvimento e testes de circuitos, por permitir a obtenção de um grande número de circuitos sem a necessidade de montagem física dos mesmos. Em particular, os dispositivos AN221E04 permitem não somente essa facilidade de implementação de diversas topologias de circuitos, mas também possibilitam que determinados parâmetros, por exemplo um ganho ou uma frequência de corte de uma determinada CAM, possam ser alterados em tempo de execução do sistema (*on-the-fly*) sem a necessidade de interromper o seu funcionamento (Caponetto *et al.*, 2005; Terry *et al.*, 2006; Baccigalupi e Liccardo, 2007).

Contudo, apesar de ser uma ferramenta bastante versátil, a reconfiguração dinâmica não pode ser realizada diretamente com a utilização do ambiente de desenvolvimento ANADIGMDESIGNER[®] EDA. Esse ambiente gera as bibliotecas de funções para a linguagem de programação C/C++ após o sistema a ser implementado no FPAA ter sido finalizado, possibilitando, assim, que o processo de reconfiguração dinâmica seja realizado de maneira automatizada por um processador mestre, como por exemplo um microprocessador (Terry *et al.*, 2006; Baccigalupi e Liccardo, 2007), ou um FPGA (Znamirovski *et al.*, 2002; Znamirovski *et al.*, 2004).

2.3.1 Sobre os Protocolos de Configuração

Vale ressaltar alguns pontos sobre a reconfiguração. Primeiramente, a topologia do sistema em execução implementado no FPAA é mantida. A reconfiguração propriamente dita ocorre apenas naquelas CAMs especificadas pelo projetista. Além disso, somente aquelas características habilitadas para a reconfiguração irão sofrer alterações, como mostra a Figura 2.7. Esse procedimento é necessário, pois funções específicas de reconfiguração são geradas pelo ambiente de desenvolvimento ANADIGMDESIGNER[®] EDA.

Outro ponto importante refere-se aos tempos envolvidos no procedimento de reconfiguração. Na inicialização, deve-se enviar ao FPAA a configuração primária responsável por definir nível alto em posições de memória específicas da *Shadow* SRAM, a qual, inicialmente, está completamente preenchida com zeros. O tempo total estimado para esse procedimento inicial é da ordem de 30 ms. Após esse tempo, o FPAA fica apto a receber os dados da configuração primária propriamente dita, sendo que, para cada *bit* transmitido são necessários $10\mu s$ (Baccigalupi e Liccardo, 2007).

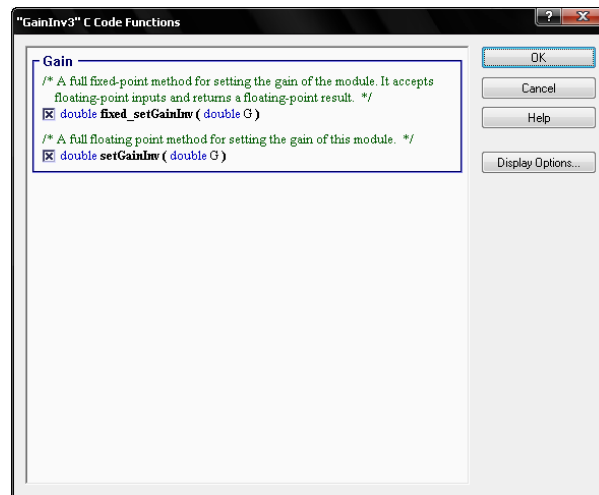


Figura 2.7: Seleção dos parâmetros da CAM habilitadas para a reconfiguração.

Independentemente do processador mestre adotado, microprocessador ou FPGA, o método para a realização da configuração dinâmica segue o mesmo padrão. Como foi ilustrado na Figura 2.1, cada um dos CABS possui memórias SRAM exclusivas, sendo subdivididas em SRAM de Configuração (*Configuration SRAM*) e *Shadow SRAM*. A memória *Shadow* é responsável por receber os dados de configuração do dispositivo sem que ocorram interferências do modelo já implementado e em execução. Após finalizado o processo de gravação da *Shadow SRAM*, em um único pulso de relógio (*clock*), todo o seu conteúdo é passado para a SRAM de Configuração, a qual é responsável pelo controle do comportamento de todos os CAMs implementados, Figura 2.8.

Como mencionado, a reconfiguração dinâmica em tempo de execução permite a alteração de determinados parâmetros dos CAMs sem modificações topológicas do circuito. O ambiente de desenvolvimento ANADIGMDESIGNER[®] EDA permite a relação analítica entre as características dos CAMs implementados e as capacitâncias envolvidas, isto é, as equações necessárias para a computação dos valores de capacitância de modo a obter o comportamento desejado do circuito. Dessa maneira, o protocolo de configuração do FPAA AN221E04 é organizado em blocos de dados, sendo que cada um deles começa com um *byte* de sincronização de comunicação. Em seguida, os próximos 2 *bytes* identificam qual FPAA será reconfigurado, no caso de existir mais de um desses dispositivos. No próximo passo, são enviados 5 *bytes*, para cada capacitância a ser alterada, sendo representados por (Baccigalupi e Liccardo, 2007):

1. Endereço de memória do capacitor;
2. Banco de memória do capacitor;

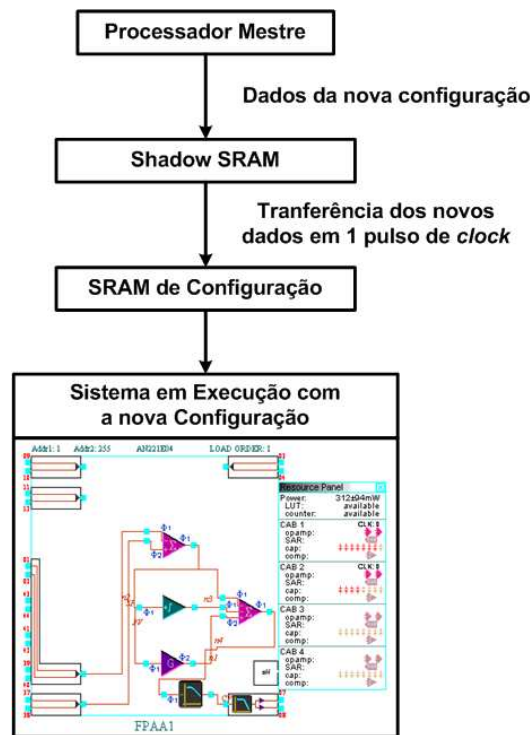


Figura 2.8: Esquema do procedimento de reconfiguração dinâmica.

3. Contador de *bytes* (quantos *bytes* contém a informação referente ao valor da capacitância);
4. Valor da capacitância (entre 1 e 255 unidades de capacitância);
5. *Byte* de checagem de erro.

É importante notar que, comparado com a configuração primária, a reconfiguração dinâmica é mais rápida de ser transmitida, já que uma menor quantidade de dados é necessária (Baccigalupi e Liccardo, 2007).

Portanto, com esses tipos de dispositivos e com estas características e possibilidades, abre-se a possibilidade para a realização de circuitos de computação analógica pela integração com circuitos de computação digital. Conseqüentemente, podem tornar muito mais eficientes os sistemas de análise e controle de processos, principalmente aqueles que possuem tempos críticos com restrição nas ações, dependentes da solução de sistemas de equações matemáticas complexas e algoritmos de implementação sofisticados. Alguns exemplos podem ser citados tais como os casos em robótica e em sistemas não-lineares que possam ter dinâmica complexa, com mudanças qualitativas em função de variações de parâmetros como aqueles que podem apresentar dinâmica caótica.

Capítulo 3

Comportamento Caótico em Sistemas Dinâmicos

3.1 Considerações Iniciais

As origens da Teoria do Caos dificilmente podem ser datadas com precisão ou atribuídas a um único indivíduo. É mais razoável dizer que essa teoria nasceu de um conjunto de observações, teoremas e descobertas nas mais diversas áreas científicas, como resultado do trabalho de muitos pesquisadores. Dentre os vários estudiosos considerados precursores, destaca-se o matemático russo Aleksandr Mikhailovich Lyapunov¹, cujos estudos foram desenvolvidos no contexto de sistemas dinâmicos não-lineares (Banbrook, 1996). Pode-se também citar o francês Jules Henri Poincaré², matemático e físico, sendo o primeiro a considerar a possibilidade de caos em sistemas determinísticos. Além de Lyapunov e de Poincaré, Edward Norton Lorenz, meteorologista interessado em estudar e prever o clima, observou certos comportamentos em sistemas que se tornaram símbolos de sistemas caóticos, e que ficaram conhecidos como Atratores Estranhos ou Atratores de *Lorenz* (Lorenz, 1963; Matsumoto *et al.*, 1985; Chua *et al.*, 1986).

Sob esse contexto, nas próximas seções são apresentadas as principais características inerentes ao comportamento dinâmico caótico, juntamente com algumas ferramentas de análise, bastante conhecidas na literatura (Parker e Chua, 1989; Fiedler-Ferrara e do Prado, 1994; Baker e Gollub, 1996; Nogueira, 2001), que possibilitam identificar tais comportamentos em sistemas dinâmicos não-lineares.

¹Maiores informações em http://en.wikipedia.org/wiki/Aleksandr_Lyapunov.

²Maiores informações em http://en.wikipedia.org/wiki/Henri_Poincar%C3%A9.

3.2 Resposta Temporal

Um sistema dinâmico, linear ou não-linear, pode ser estudado a partir do comportamento de suas variáveis de estado ao longo do tempo. Em especial, sistemas caóticos possuem trajetórias bastante complexas, imprevisíveis e podem não retornar a um determinado ponto anteriormente visitado (Nogueira, 2001).

Outra característica bastante importante dos sistemas caóticos que pode ser observada com o auxílio da resposta temporal do sistema é a alta sensibilidade às variações das condições iniciais. O conceito envolvido objetiva obter, após um determinado intervalo de tempo, comportamentos diferentes para o mesmo sistema dinâmico iniciado em condições diferentes, escalares genericamente definidos como x_0 e $x_0 + \epsilon$, com ϵ infinitesimalmente pequeno (Fiedler-Ferrara e do Prado, 1994; Baker e Gollub, 1996).

A título de ilustração, considere o modelo dinâmico meteorológico dado pelo sistema de equações (3.1), originalmente apresentado por Lorenz (1963), o qual diz respeito à instabilidade de *Rayleigh-Bérnard* de um fluido localizado entre duas placas horizontais,

$$\begin{aligned}\dot{x} &= -\sigma(x - y) \\ \dot{y} &= rx - y - xz \\ \dot{z} &= xy - bz\end{aligned}\tag{3.1}$$

onde x é proporcional à intensidade do movimento de convecção do fluido, y é proporcional à diferença de temperatura entre as correntes de fluido ascendente e descendente e z representa o gradiente vertical de temperatura entre as placas (Fiedler-Ferrara e do Prado, 1994; Baker e Gollub, 1996). Os parâmetros σ , b e r são constantes do modelo e, em especial, para $\sigma = 10$ e $b = \frac{8}{3}$, valores de r maiores que 28 resultam em comportamento caótico (Lorenz, 1963).

Na Figura 3.1(b), tem-se a resposta temporal de uma função seno para frequência de oscilação de 60 Hz, e na Figura 3.1(a) observa-se a resposta temporal do sistema (3.1) operando em caos. Para a obtenção desses resultados, as condições iniciais de simulação do sistema de Lorenz (3.1) foram $x_0 = 0.0$, $y_0 = -0.01$ e $z_0 = 9.0$, método de integração *Runge-Kutta* de 4ª ordem com passo fixo de 0.001, e utilizando o MATLAB/SIMULINK.

Como foi apresentado anteriormente, sistemas caóticos são sensíveis a pequenas variações nas condições iniciais de suas variáveis de estado. Para o sistema (3.1), foram utilizadas

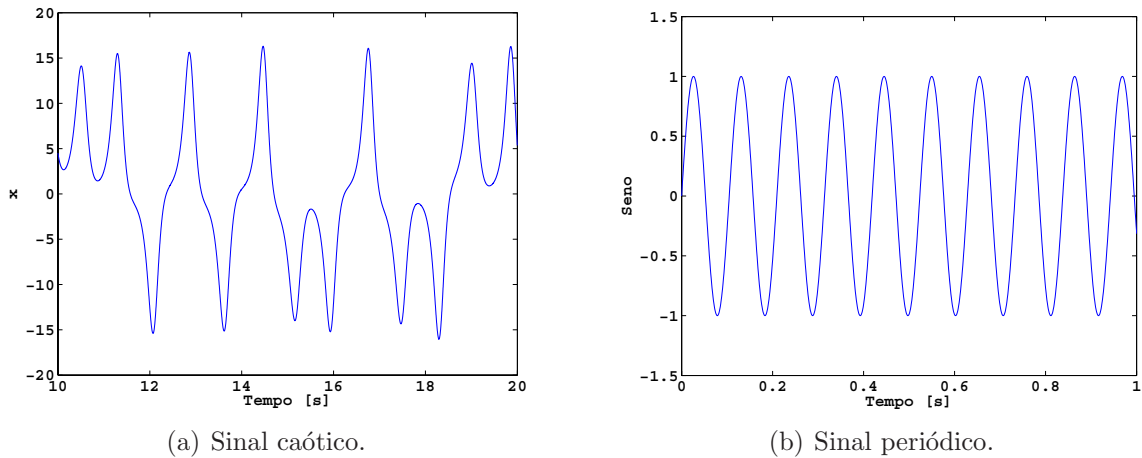


Figura 3.1: Comparação de complexidade da resposta temporal entre um sinal caótico e um sinal periódico.

as condições de simulação $y_0 = -0.01$, $z_0 = 9.0$ com $x_0 = 0.0$ na primeira simulação, e na segunda simulação, perturbou-se a última variável de estado para $x_0 = 0.001$. Em ambos os casos, adotou-se método de integração *Runge-Kutta* de 4ª ordem com passo fixo de 0.001. Dessa forma, foi possível construir as curvas apresentadas na Figura 3.2(a), que representa o fenômeno em questão e, na Figura 3.2(b), a evolução temporal do erro normalizado.

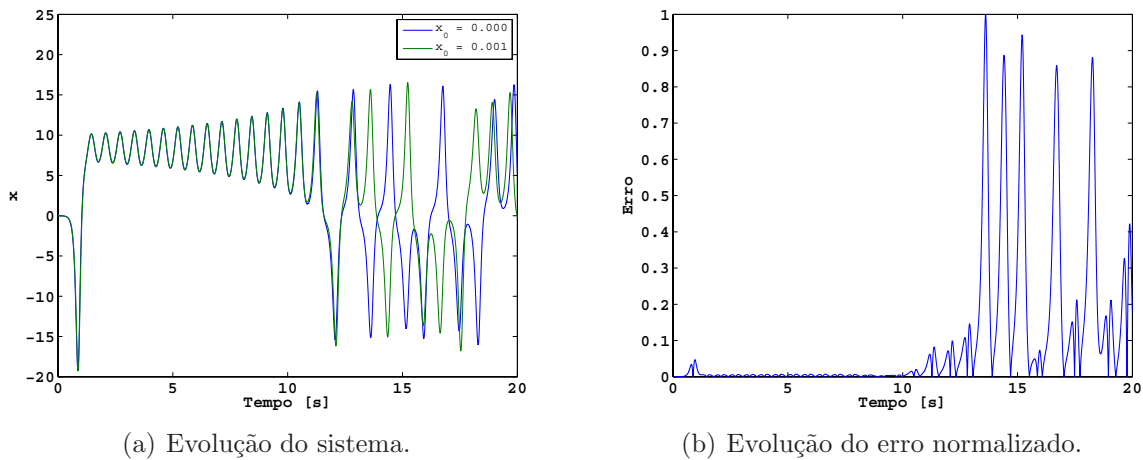


Figura 3.2: Verificação da sensibilidade às condições iniciais em um sistema caótico.

Pode-se perceber que, após um tempo relativamente curto de simulação (por volta de 10 segundos), o sistema perturbado passou a apresentar um comportamento imprevisível em relação ao comportamento obtido na primeira simulação, comprovando o efeito da sensibilidade às condições iniciais.

3.3 Plano de Fase e Espaço de Estados

Assim como a resposta temporal, os planos de fase e os espaços de estado consistem de ferramentas de análise qualitativas, ou seja, permitem apenas identificar se o sistema é ou não periódico. Essa identificação é possível pois sistemas periódicos possuem um plano de fase, ou seja, uma curva fechada denominada ciclo limite, e sistemas caóticos, na maioria dos casos, atratores estranhos (Nogueira, 2001).

Utilizando o sistema (3.1), foi possível reproduzir o atrator estranho apresentado por Lorenz (1963), Figura 3.3(a) e também o atrator estranho obtido no espaço de estados (x, y, z) , Figura 3.3(b).

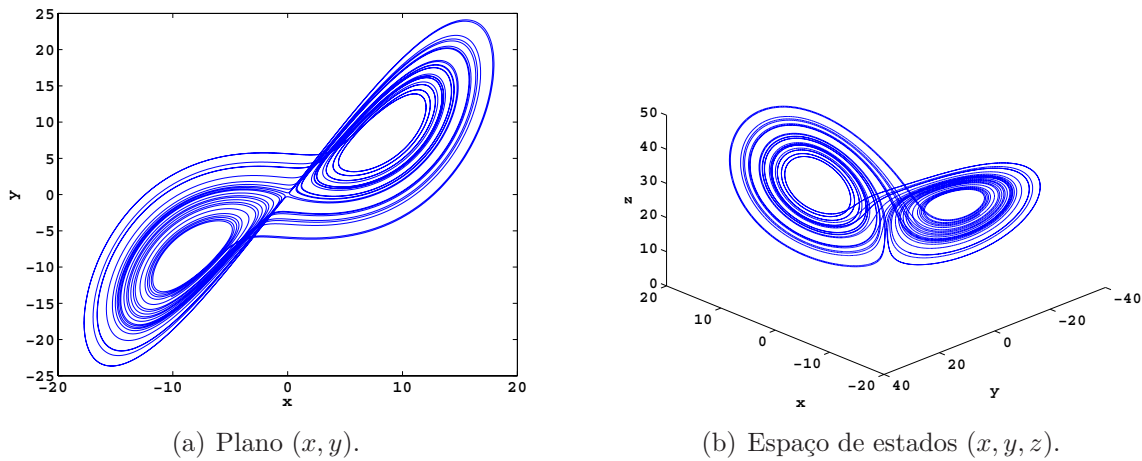


Figura 3.3: Atrator estranho obtido a partir do sistema de Lorenz (3.1).

Para a obtenção da Figura 3.3(a) e da Figura 3.3(b), o sistema (3.1) foi simulado para as condições iniciais $x_0 = 0.0$, $y_0 = -0.01$ e $z_0 = 9.0$, método de integração *Runge-Kutta* de 4ª ordem com passo fixo de 0.001 e tempo de simulação computacional de 200 segundos, o que, na realidade, correspondeu a mais de 30 minutos de simulação computacional contínua em um computador com processador AMD SEMPRON 2800+ (256 kbytes de memória *cache* L2, soquete de 754 pinos), disco rígido Samsung SATA II, 1 GB de memória RAM (*Random Access Memory*) e placa de vídeo *off-board* com processador gráfico dedicado da ATI com 256 MB de memória.

A forma do atrator apresentado na Figura 3.3(a) não é única pois depende do sistema em estudo. Por exemplo, para o modelo apresentado por Chen e Ueta (1999), representado pelo sistema de equações (3.2),

$$\begin{aligned}
\dot{x} &= a(y - x) \\
\dot{y} &= (c - a)x - xy + cy \\
\dot{z} &= xy - bz
\end{aligned} \tag{3.2}$$

pode-se obter o atrator estranho no plano de estados (x, y) , Figura 3.4(a), e o atrator no espaço de estados (x, y, z) , Figura 3.4(b). Particularmente, o sistema de *Chen* (3.2) e o sistema de *Lorenz* (3.1) compartilham de propriedades tais como simetria, bifurcações e topologia similares (Lü *et al.*, 2002). Para a obtenção desses atratores, utilizou-se método de integração *Runge-Kutta* de 4ª ordem com passo fixo de 0.001. As condições iniciais foram definidas como $x_0 = -10.0$, $y_0 = 0.0$ e $z_0 = 37.0$ e os outros parâmetros como $a = 35$, $b = 3$ e $c = 28$.

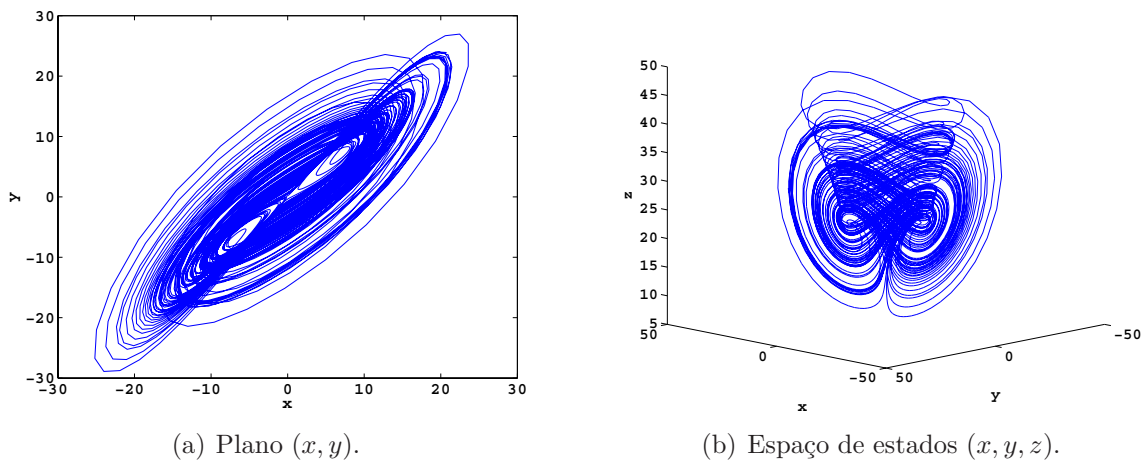


Figura 3.4: Atratores estranhos obtidos a partir do sistema de *Chen* (3.2).

Outro sistema interessante e bastante conhecido, que possui um atrator bastante peculiar, é o sistema de *Rössler* (3.3). Esse é um sistema dissipativo de 4ª ordem e, dependendo dos seus parâmetros, pode apresentar até 2 expoentes de *Lyapunov* positivos. Por essa razão, usualmente é denominado um sistema hiper-caótico (Sprott, 2003).

$$\begin{aligned}
\dot{x} &= -y - z \\
\dot{y} &= x + ay + w \\
\dot{z} &= b + xz \\
\dot{w} &= cw - dz
\end{aligned} \tag{3.3}$$

Para que o sistema (3.3) operasse em caos, os seus parâmetros foram definidos como sendo $a = 0.25$, $b = 3.0$, $c = 0.05$, $d = 0.5$, com condições iniciais dadas por $x_0 = -20$, $y_0 = 0.0$, $z_0 = 0.0$ e $w_0 = 15.0$ (Ramasubramanian e Sriram, 2000). Utilizando-se método de integração *Runge-Kutta* de 4ª ordem com passo fixo de 0.001, tem-se o atrator do plano de estados (x, y) apresentado na Figura 3.5.

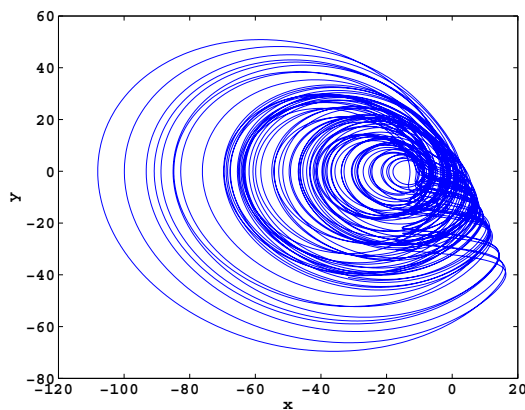


Figura 3.5: Atrator estranho obtido a partir do sistema de *Rössler* (3.3).

A análise por plano de fase e espaço de estados é bastante útil, pois possibilita qualificar o sistema em estudo. Contudo, é preciso tomar alguns cuidados para evitar que conclusões precipitadas sejam tomadas. Quando são estudados sistemas dinâmicos caóticos, deve-se tomar o cuidado de permitir que o sistema evolua durante um tempo suficientemente longo de tal maneira que o comportamento de transitório inicial seja desconsiderado, ou seja, o sistema deve estar em regime. Isso consiste em um problema bastante complexo no sentido de que esse tempo não pode ser generalizado para diferentes sistemas caóticos, e, além disso, o tempo de transitório pode ser bastante grande, geralmente implicando que o tempo real de simulação seja longo (Nogueira, 2001).

3.4 Mapas de *Poincaré*

Os Mapas de *Poincaré* são gráficos que intencionam oferecer informações sobre as periodicidades envolvidas nos sistemas dinâmicos e que são construídos a partir da amostragem de uma ou mais variáveis de estado do sistema. O procedimento de amostragem não é realizado no tempo, e sim a partir de outras variáveis do próprio sistema. Suponha que um determinado sistema dinâmico possua 3 variáveis de estado, x , y e z , ou seja, que se trata de um sistema imerso no espaço cartesiano $\mathbb{R}^3$. Um mapa pode ser construído a partir da aquisição dos valores

de x e/ou de y sempre que as mesmas cruzarem o hiperplano definido por $z = z_i$ previamente escolhido. Na Figura 3.6 tem-se uma ilustração para a situação em que um determinado sistema encontra-se em comportamento periódico, e na Figura 3.7 tem-se o mesmo sistema operando em caos. Percebe-se que o Mapa de *Poincaré* permite estimar a quantidade das componentes em frequência pela identificação das intersecções da trajetória do sistema com um hiperplano.

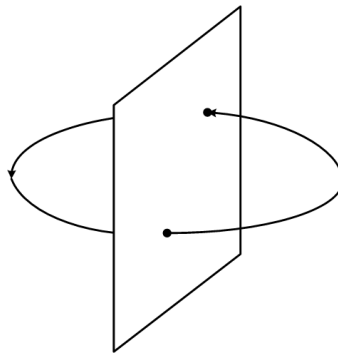


Figura 3.6: Mapa de *Poincaré* para a situação de um sinal periódico.

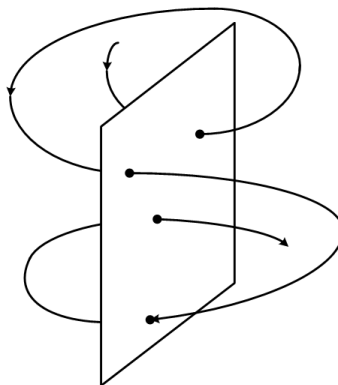


Figura 3.7: Mapa de *Poincaré* para a situação de um sinal caótico.

A título de observação, uma outra ferramenta de análise, conhecida como Densidade Espectral de Potência (DEP) pode ser utilizada em conjunto com os Mapas de *Poincaré* objetivando auxiliar nas identificações das componentes em frequência. Sistemas periódicos possuem uma DEP bem definida, com componentes em frequência distintas. No caso de sistemas caóticos, o espectro de frequência é cheio, ou seja, sinais caóticos são compostos por infinitas componentes em frequência. Durante muito tempo, a análise via utilização do espectro de frequências levou a conclusões erradas, visto que sinais caóticos eram considerados ruídos, como por exemplo, o ruído branco. Contudo, o que distingue um ruído branco de um sinal periódico ou do caos é o fato de que, no caos, algumas componentes em frequência possuem maior ener-

gia do que outras ao contrário do ruído branco em que todas as frequências possuem a mesma densidade energética (Fiedler-Ferrara e do Prado, 1994; Abarbanel, 1996; Nogueira, 2001).

Um dos grandes problemas na construção dos Mapas de *Poincaré* está justamente relacionado à definição do hiperplano o qual, teoricamente, deve possuir espessura nula. Na prática isso é extremamente complicado de se realizar, visto que qualquer que seja o sistema computacional utilizado, ou seja, independentemente do processador, do sistema operacional e do ambiente de desenvolvimento (nesse caso, o MATLAB), existem precisões numéricas envolvidas inerentes à máquina, o que dificulta ou mesmo impossibilita a obtenção de tal hiperplano de espessura nula.

Com o objetivo de contornar tais dificuldades, algumas soluções, dependendo da aplicação e dos estudos envolvidos, podem ser aplicadas, como a recorrência no tempo e recorrência na própria variável (Nogueira, 2001). Para essa dissertação, como o ambiente de trabalho utilizado é o MATLAB, foi possível utilizar o bloco da biblioteca padrão de componentes do SIMULINK denominado *hit crossing*. O algoritmo desse bloco permite que o próprio MATLAB ajuste o passo de integração do modelo durante a execução do sistema de tal maneira que a espessura do hiperplano seja a menor possível, evitando-se ao máximo que ocorram deformações nos mapas. Esse ajuste é realizado de maneira mais eficiente quando utiliza-se passo variável durante a integração numérica do sistema.

No caso de um sistema realmente operando em caos, pode-se prever que ocorra uma grande quantidade de intersecções da trajetória com o hiperplano de *Poincaré*. Para melhor visualização, na Figura 3.8(a), tem-se a seção de *Poincaré* para o sistema de *Chen* (3.2). Para a obtenção dessa seção, adotou-se integração numérica do tipo *ode113(Adams)* com passo variável de integração. Adotou-se, também, tolerância relativa e tolerância absoluta iguais a 10^{-10} e a seção sendo definida por $z = 22$. Na Figura 3.8(b), tem-se a superfície de *Poincaré* para o sistema de *Rössler* (3.3) utilizando-se as mesmas configurações de simulação e a seção foi definida como $w = 22$. Em ambos os casos, utilizaram-se as variáveis de estado x e y para a construção das seções.

Uma das grandes dificuldades na construção dos mapas de *Poincaré* está relacionada à grande quantidade de pontos que devem ser adquiridos para que as figuras obtidas apresentem uma resolução razoável. No caso do mapa obtido na Figura 3.8(b), foram necessários mais de 33000 pontos. Além disso, o tempo necessário para a aquisição desses mapas é longo em virtude de que o transitório deve ser desconsiderado, e, como foi comentado, esse tempo é função do tipo do sistema em estudo.

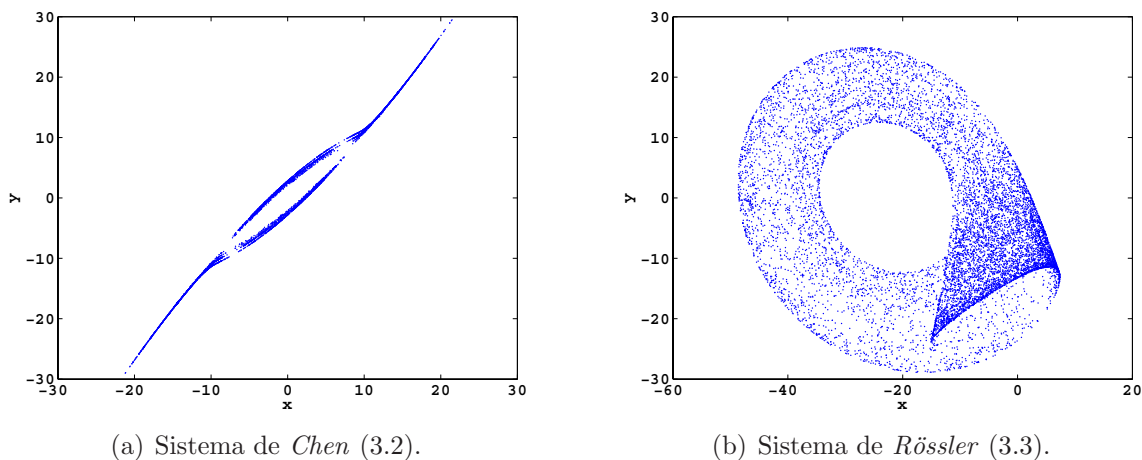


Figura 3.8: Seções de *Poincaré*, plano (x, y) , para sistemas com comportamento caótico.

3.5 Diagramas de Bifurcação

Diagramas de fase e seções de *Poincaré* provêm informações qualitativas em relação à dinâmica de um determinado sistema para valores específicos de seus parâmetros. Contudo, a dinâmica pode ser analisada a partir de uma visão mais global, especificando-se um intervalo de variação de valores de um determinado parâmetro do sistema. Dessa forma, é possível a identificação de como a qualidade do sistema varia conforme um determinado parâmetro é alterado, permitindo a identificação do comportamento periódico e, dependendo do sistema, do comportamento caótico (Baker e Gollub, 1996).

Sabendo-se que o aparecimento de caos em sistemas dinâmicos está intimamente ligado à ocorrência de algum tipo de bifurcação (Fiedler-Ferrara e do Prado, 1994), a variação de um parâmetro do sistema permite a construção de um diagrama (ou mapa) de bifurcação cuja finalidade consiste em mostrar o seu comportamento em função dessa variação, mantendo-se todos os outros parâmetros constantes.

A construção de um diagrama de bifurcação segue o mesmo conceito empregado na obtenção das seções de *Poincaré*, ou seja, no caso dos experimentos realizados, a definição de uma seção está intimamente relacionada ao valor da variável de estado amostrada. Consequentemente, para diferentes valores dessa variável, são obtidas seções de *Poincaré* distintas. Portanto, o diagrama de bifurcação pode ser obtido pela justaposição de diversas seções obtidas para cada valor de um determinado parâmetro (Baker e Gollub, 1996; Nogueira, 2001).

Como ilustração, na Figura 3.9 tem-se o diagrama de bifurcação obtido pela simulação do sistema original de *Rössler*, definido pelas equações (3.4), utilizando-se os parâmetros

$a = 0.2$, b variando entre 0.0 e 2.0 com passo de 0.001, e o parâmetro $c = 5.7$.

$$\begin{aligned}\dot{x} &= -y - z \\ \dot{y} &= x + ay \\ \dot{z} &= b + z(x - c)\end{aligned}\tag{3.4}$$

Na Figura 3.9(a), tem-se o diagrama de bifurcação para a variável de estado x e, na Figura 3.9(b), o diagrama para a variável de estado y . Em ambos os casos, o plano de *Poincaré* foi definido para $z = 1.0$, integração numérica `ode113(Adams)` com passo variável, tolerância relativa e tolerância absoluta iguais a 10^{-10} . As condições iniciais do sistema foram $x_0 = -9.0$, $y_0 = 0.0$ e $z_0 = 0.0$. Para a obtenção dessas figuras, foram necessárias 5 horas de simulação contínua em um computador com processador AMD SEMPRON 2800+ (256 kbytes de memória *cache* L2, soquete de 754 pinos), disco rígido Samsung SATA II, 1 GB de memória RAM (*Random Access Memory*) e placa de vídeo *off-board* com processador gráfico dedicado da ATI com 256 MB de memória.

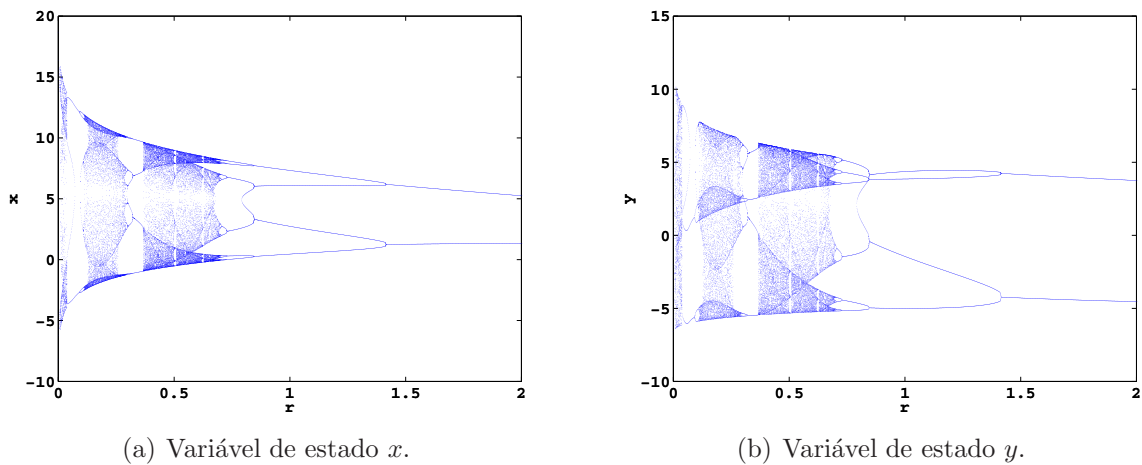


Figura 3.9: Diagrama de bifurcação para o sistema de *Rössler* (3.4).

Diante dos resultados obtidos, percebe-se que a obtenção de um diagrama de bifurcação é uma tarefa bastante complexa e custosa sob o ponto de vista computacional. Uma primeira dificuldade reside no fato de que muitos pontos são adquiridos, principalmente, quando o sistema está operando em caos, já que a manipulação dos dados requer bastante poder computacional. Isso pode ser facilmente constatado nos diagramas da Figura 3.9. Outra dificuldade está no fato de que é necessário que o passo de variação do parâmetro deve ser pequeno (no

mínimo, da ordem de 10^{-2}) em virtude da necessidade de precisão dos resultados. Finalmente, cada uma dessas seções deve ser construída levando-se em consideração o transitório do sistema em análise o que pode aumentar, consideravelmente, o tempo necessário para a obtenção dos resultados. No caso dos experimentos realizados por Nogueira (2001), alguns diagramas de bifurcação demoraram 10 dias para serem obtidos, tendo sido usada uma máquina relativamente potente de processamento digital para a época, um microcomputador com processador *Pentium* III com *clock* de 450 MHz. Portanto, a diminuição desse tempo é uma das justificativas em tentar-se realizar estudos de sistemas dinâmicos operando em caos a partir de implementações em *hardware*.

3.6 Expoentes de *Lyapunov*

3.6.1 Conceitos Iniciais

A classificação dos sistemas dinâmicos é uma parte crítica no processo de análise dos sinais medidos. Existem, então, características, ditas invariantes, independentes tanto das condições iniciais quanto do sistema de coordenadas sob o qual o atrator é observado (Abarbanel, 1996). Uma das características mais importantes é conhecida como expoente de *Lyapunov*, os quais, para o caso de sistemas dinâmicos caóticos, podem ser interpretados como sendo a taxa média exponencial de divergência (expoentes positivos) ou de convergência (expoentes negativos) de órbitas inicialmente próximas no espaço de fase, ou seja, esses expoentes quantificam a sensibilidade dos sistemas dinâmicos às condições iniciais (Wolf *et al.*, 1985). Além disso, os expoentes de *Lyapunov* podem ser utilizados para a determinação da estabilidade de comportamentos periódicos e caóticos bem como pontos de equilíbrio e instabilidades (Fiedler-Ferrara e do Prado, 1994; Nogueira, 2001).

Dentre as muitas maneiras de explorar, matematicamente, o conceito dos expoentes de *Lyapunov*, duas serão abordadas nesse texto. A primeira baseia-se em análise por elementos de volume (Fiedler-Ferrara e do Prado, 1994; Nogueira, 2001). Inicialmente, considera-se um sistema contínuo composto de n equações diferenciais ordinárias e um pequeno hiper-volume esférico de raio $\epsilon_0(x_0)$ formado por estados iniciais x_{ϵ_0} em torno de um ponto inicial x_0 de uma linha de fluxo, isto é,

$$|x_{\epsilon_0} - x_0| \leq \epsilon_0(x_0) \quad (3.5)$$

Conforme o tempo passa, o fluxo deforma essa hiper-esfera dando origem a um hiper-elipsóide com eixos principais $\epsilon_k(t)$, $k = 1, 2, 3, \dots, n$. Na Figura 3.10 tem-se uma ilustração da evolução do elemento de volume para um caso bidimensional.

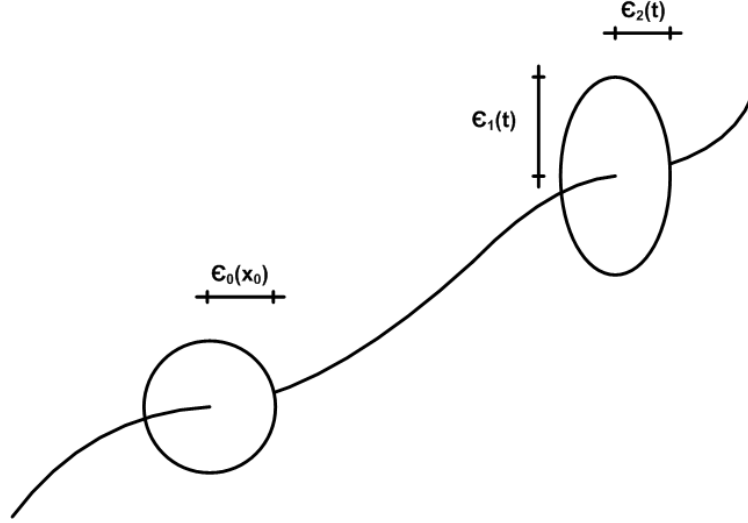


Figura 3.10: Evolução do elemento de volume particularizado para um caso bidimensional.

Dessa forma, os expoentes de *Lyapunov* medem a taxa de crescimento exponencial dos eixos principais e são definidos pela relação (3.6),

$$\lambda_i = \lim_{t \rightarrow \infty} \lim_{\epsilon_0(x_0) \rightarrow 0} \left(\frac{1}{t} \right) \ln \left(\frac{\epsilon_i(t)}{\epsilon_0(x_0)} \right) \quad (3.6)$$

com $i = 1, 2, 3, \dots, n$. A partir de manipulações matemáticas da equação (3.6), obtém-se a relação (3.7),

$$\epsilon_i(t) \sim \epsilon_0(x_0) \exp^{\lambda_i t} \quad (3.7)$$

de onde pode-se concluir que (Fiedler-Ferrara e do Prado, 1994; Nogueira, 2001):

1. A existência de expoentes de *Lyapunov* positivos define uma instabilidade orbital nas direções associadas, ou seja, uma expansão;
2. A existência de expoentes de *Lyapunov* negativos significa a existência de contração;
3. Para uma solução caótica associada a um atrator estranho, a dependência às condições iniciais implica na existência de pelo menos um expoente de *Lyapunov* positivo;

4. Para uma solução periódica, pode-se esperar que deslocamentos na direção perpendicular ao movimento diminuam com o tempo, enquanto que ao longo da trajetória eles não devem se alterar, correspondendo a um simples deslocamento do ponto inicial. Portanto, a partir da equação (3.7), segue que no caso de solução periódica, $\lambda_i < 0$ nas direções perpendiculares ao movimento e $\lambda_i = 0$ ao longo da trajetória.

Em um instante de tempo qualquer t , o elemento do hiper-volume no espaço de fases é dado por (3.8),

$$\delta V(t) = \prod_{i=1}^n \epsilon_i(t) \quad (3.8)$$

Substituindo-se (3.7) em (3.8), obtém-se a relação (3.9), a qual será melhor analisada na Sub-Seção 3.6.2,

$$\delta V(t) = \delta V(0) \exp \sum_{i=1}^n \lambda_i(t) \quad (3.9)$$

A segunda possibilidade de estudo do conceito dos expoentes de *Lyapunov* podem ser analisados considerando-se, inicialmente, duas trajetórias vizinhas divergentes descritas por equações diferenciais, assim como é ilustrado na Figura 3.11 (Wolf *et al.*, 1985; Kinsner, 2006).

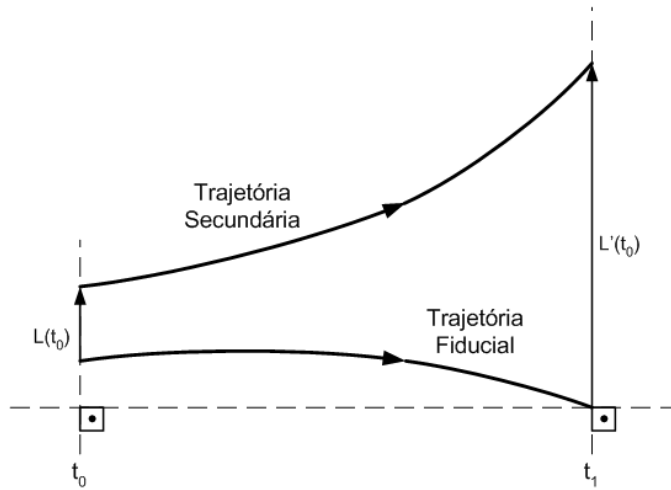


Figura 3.11: Ilustração do conceito de divergência das trajetórias para um sistema dinâmico.

Dessa forma, pode-se então descrever o crescimento da diferença $L(t_0)$ entre a trajetória de referência (fiducial) e a trajetória secundária (perturbada) durante um determinado intervalo de tempo $\Delta t = t_1 - t_0$ pela equação (3.10),

$$L'(t_0) = L(t_0) \exp^{(\lambda^{(0)} \Delta t)} \quad (3.10)$$

onde $\lambda^{(0)}$ corresponde ao expoente de *Lyapunov*, $L(t_0)$ representa a distância inicial entre as trajetórias e $L'(t_0)$ equivale à distância no instante de tempo t_1 . O expoente pode, então, ser calculado a partir da relação (3.11),

$$\lambda^{(0)} = \left(\frac{1}{\Delta t} \right) \ln \left(\frac{L'(t_0)}{L(t_0)} \right) \quad (3.11)$$

O cálculo de λ é compreendido facilmente. Contudo, são encontradas grandes dificuldades em relação à precisão dos resultados obtidos devido ao crescimento exponencial, inerente à sua própria definição. Portanto, adota-se o cálculo do expoente de *Lyapunov* de não apenas uma trajetória vizinha e sim de tantas quanto forem possíveis até que o final da trajetória fiducial seja atingido, ou seja, calcula-se o valor médio ao longo de toda a trajetória (Kinsner, 2006), Figura 3.12.

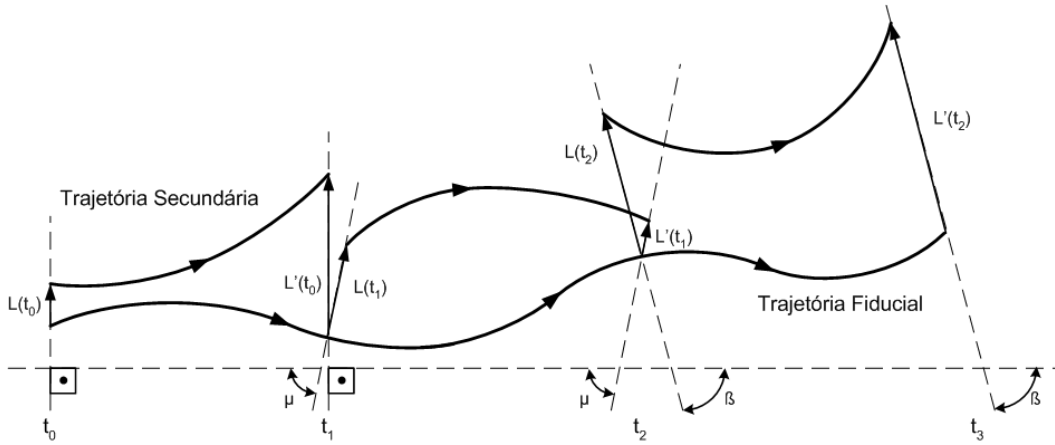


Figura 3.12: Cálculo dos expoentes de *Lyapunov* ao longo de uma trajetória fiducial.

Dessa forma, considerando-se N segmentos de trajetórias vizinhas, o valor médio do expoente de *Lyapunov*, λ_i , é obtido a partir da relação (3.12),

$$\lambda_i = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{j=0}^{N-1} \lambda^{(j)} \quad (3.12)$$

onde $\lambda^{(j)}$ corresponde ao cálculo do expoente calculado no segmento j considerado, por exemplo, no intervalo de tempo Δt . A utilização do valor médio do expoente de *Lyapunov* obtido ao

longo de N segmentos de trajetória permite que eventuais erros numéricos sejam menores quando comparados ao obtido pela relação (3.11) (Kinsner, 2006).

3.6.2 Aplicação aos Modelos Dinâmicos

Segundo Parker e Chua (1989), uma característica bastante importante de sistemas dinâmicos não-lineares, em especial sistemas caóticos que pode ser observada a partir da definição (3.12) consiste no fato de que pelo menos um dos expoentes de *Lyapunov* é positivo, representando, assim, o fenômeno de divergência entre as trajetórias, ou seja, a sensibilidade às condições iniciais. Outra interpretação possível permite dizer que essa qualidade de expoentes representa uma certa quantidade de informação gerada pelo sistema após um intervalo de tempo Δt , sendo que essa informação pode ser precisamente quantificada pelo uso de estudos específicos, tais como a entropia de *Kolmogorov-Sinai* (Fiedler-Ferrara e do Prado, 1994), ou seja, além da existência de expoentes de *Lyapunov* positivos permitir distinguir atratores estranhos de atratores não caóticos, tal existência permite também concluir que sistemas caóticos são fontes de informação (Abarbanel, 1996).

Em um sistema não-linear com d graus de liberdade, existem d expoentes de *Lyapunov* tais que, por definição, $\lambda_1 > \lambda_2 > \dots > \lambda_d$. Uma questão importante levantada pela equação (3.9) é que a soma de todos os expoentes deve ser negativa para que o hiper-volume no espaço de fases não seja divergente (Farmer *et al.*, 1983; Wolf *et al.*, 1985; Parker e Chua, 1989). O fato dessa soma ser negativa, ou seja,

$$\sum_{i=1}^d \lambda_i < 0 \quad (3.13)$$

significa dizer que os pontos da órbita do sistema não tendem a escapar para o infinito, apesar da instabilidade inerente do sistema. A órbita do sistema fica confinada em uma região compacta do espaço de estados, cujo volume é comprimido a uma taxa equivalente à soma de todos os λ_i , em outras palavras, substituindo (3.13) na equação (3.9), tem-se

$$\delta V(t) < \delta V(0) \quad (3.14)$$

Os valores positivos de λ_i são, então, responsáveis por expandir a órbita de tal forma a levá-la à instabilidade local. Os fenômenos de expansão, devido à instabilidade, e de contração,

devido à dissipação (que ocorre dentro de limites energéticos do sistema), formam um conjunto fundamental de processos responsável por formar estruturas fractais (Abarbanel, 1996; Sprott, 2003), cuja principal característica consiste na repetição da estrutura independentemente da escala em que é observada (Nogueira, 2001).

Em trabalhos como o de Parker e Chua (1989) e o de Abarbanel (1996), justifica-se que sistemas descritos por equações diferenciais que evoluem no tempo apresentam pelo menos um expoente nulo. Já para Sprott (2003), o expoente nulo representa a direção do fluxo do movimento. Dessa forma, para sistemas dinâmicos contínuos no tempo que apresentam comportamento caótico, são necessários, no mínimo, 3 expoentes de *Lyapunov*, cuja única combinação de sinais possível é dada por $(+, 0, -)$. Portanto, não é possível a verificação de caos em sistemas contínuos autônomos de 1ª e de 2ª ordem e em sistemas contínuos não-autônomos de 1ª ordem. Para o caso de sistemas de 4ª ordem, como é o caso do modelo de *Rössler* (3.3), existem várias combinações de sinais para os λ_i , sendo que algumas estão resumidas na Tabela 3.1 (Parker e Chua, 1989; Sprott, 2003):

λ_1	λ_2	λ_3	λ_4	Atrator
-	-	-	-	Ponto de equilíbrio
0	-	-	-	Ciclo limite
+	0	-	-	Atrator estranho (caótico)
+	+	0	-	Atrator estranho (hipercaos)
+	0	0	-	2-torus caótico

Tabela 3.1: Combinações possíveis dos sinais dos expoentes de *Lyapunov* para um sistema dinâmico de 4ª ordem.

Um estudo bastante importante, que em conjunto com os diagramas de bifurcação realmente comprova o comportamento caótico no sistema, consiste na construção do Espectro de *Lyapunov*. Esse espectro consiste em obter o valor de todos os λ_i do sistema variando-se um ou mais parâmetros específicos do modelo. Por exemplo, tem-se o modelo dinâmico (3.15),

$$\begin{aligned}
 \dot{x} &= -y^2 - z^2 - ax + aF \\
 \dot{y} &= xy - bxz - y + G \\
 \dot{z} &= bxy + xz - z
 \end{aligned}
 \tag{3.15}$$

onde, segundo os resultados apresentados por Lorenz (2005), o sistema (3.15) irá operar com comportamento caótico para valores de $G \leq 1.367$ quando os outros parâmetros do sistema são

definidos por $a = 0.25$, $b = 4.0$ e $F = 8.0$. O espectro de *Lyapunov*, Figura 3.13, para o modelo em questão, foi obtido variando-se o parâmetro G entre 0.98 e 1.38, em passos de 0.00005. As condições iniciais adotadas para a simulação são iguais a $x_0 = 1.0$, $y_0 = 0.0$ e $z_0 = -0.75$. O método de integração numérico utilizado foi o *Runge-Kutta* de 4ª ordem.

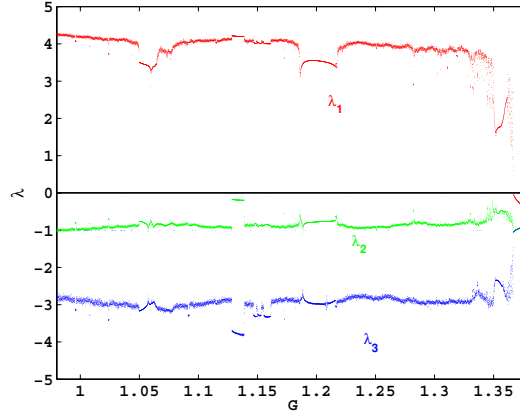


Figura 3.13: Espectro de *Lyapunov* para o sistema de *Lorenz* (3.15).

Apesar de a simulação realizada adotar todas as especificações apresentadas por Lorenz (2005) quando os resultados da Figura 3.13 são comparados aos resultados obtidos no trabalho original, pode-se perceber que existem distorções que podem ser interpretadas, por exemplo, como o ambiente de desenvolvimento utilizado pelo autor seja diferente do ambiente utilizado neste trabalho de dissertação de mestrado, o qual é discutido no Capítulo 5. Com isso, pode-se verificar que é bastante complexo o estudo de sistemas dinâmicos operando em caos, no sentido de que o ambiente computacional (*hardware* e *software*) em que o modelo é executado pode interferir bastante nos resultados obtidos, por exemplo, pela injeção de incertezas e truncamentos.

Outro grande problema encontrado na obtenção desse espectro está relacionado ao tempo. A partir da definição do λ_i (3.12), o valor correto do expoente a ser obtido somente deverá ser encontrado depois de muito tempo de operação do sistema (idealmente no infinito). Na prática, são adotados critérios de parada que melhor se adequem ao modelo em estudo, ou seja, para cada caso devem ser levadas em consideração características distintas, como por exemplo o tempo de transitório envolvido. Em especial, o transitório é um grande complicador no cálculo dos expoentes, já que, como foi observado anteriormente, ele deve ser desconsiderado para que as análises não sejam prejudicadas. No caso do modelo (3.15), para cada variação do parâmetro G foi levado em consideração um tempo de transitório de 100 segundos, sendo que esse valor foi estimado a partir de simulações realizadas isoladamente, antes de o procedi-

mento de construção do espectro ser iniciado. Dessa forma, foram necessárias quase 5 horas de simulação contínua para a obtenção dos resultados.

É importante ressaltar que a metodologia empregada no cálculo dos expoentes de *Lyapunov* baseou-se fortemente nos procedimentos desenvolvidos por Nogueira (2001) onde emprega-se a técnica da equação variacional em conjunto com a ortonormalização de *Gram-Schmidt*. Basicamente, a cada passo de integração, o sistema original é perturbado e o expoente é calculado a partir da trajetória percorrida por ele e pela trajetória fiducial. A ortonormalização é empregada para garantir que o sistema de coordenadas referente à medida $L'(t_i)$ respeite as orientações do sistema de coordenadas imediatamente anterior, ou seja, da medida $L(t_i)$. Dessa forma, previnem-se erros numéricos durante o emprego das expressões (3.10), (3.11) e (3.12) mas a simulação é comprometida no sentido de necessitar de mais poder computacional para a realização das integrações numéricas e, conseqüentemente, mais tempo de simulação.

No Capítulo 5 são apresentadas todas as ferramentas computacionais utilizadas para a realização das simulações apresentadas, não somente neste capítulo mas também em todas as outras simulações realizadas ao longo deste trabalho.

Capítulo 4

Séries Temporais

4.1 Considerações Iniciais

A análise de sinais amostrados ao longo do tempo permite identificar o comportamento da dinâmica dos sistemas em estudo. Dessa maneira, como o procedimento de aquisição produz uma série temporal, passa a ser interessante apresentar alguns dos princípios teóricos envolvidos no estudo das séries. Esse é, então, o principal objetivo deste capítulo.

4.2 Séries Temporais

Em diversas áreas científicas, tais como a engenharia, medicina e biologia, deseja-se realizar estudos da dinâmica de determinados sistemas com o intuito de analisar o seu comportamento e, dependendo dos objetivos envolvidos, realizar controle sobre o mesmo. Usualmente, tais análises são possíveis de realizar através da aquisição, ao longo do tempo, de alguns sinais referentes a uma ou mais variáveis de estado do sistema. Conseqüentemente, o que se tem são dados dispostos na forma de séries temporais (Principe e Lo, 1991; Abarbanel, 1996; Kinsner, 2006).

De forma geral, define-se o vetor série temporal, X , sendo representado pela expressão (4.1),

$$X = \begin{bmatrix} x(t_0) & x(t_1) & x(t_2) & \dots & x(t_N) \end{bmatrix} \quad (4.1)$$

onde $x(t_j)$, para $j = 0, 1, 2, \dots, N$, representam os pontos adquiridos em instantes de tempo t_j e N consiste no número máximo de aquisições consideradas. O intervalo de tempo entre duas aquisições consecutivas é constante e é dado por $\Delta t \triangleq t_{j+1} - t_j$. A partir do vetor X pode-se, então, analisar a resposta temporal do sistema dinâmico em estudo, procedimento análogo ao que foi proposto na Seção 3.2.

A importância das séries temporais neste trabalho diz respeito ao fato de que é uma ferramenta bastante eficiente para a realização de estudos quantitativos dos modelos implementados no FPAA a partir da aquisição, a uma taxa constante, dos sinais provenientes do *kit* de desenvolvimento apresentado na Figura 2.3. Maiores detalhes sobre os procedimentos experimentais adotados são discutidos no Capítulo 6. A seguir, é apresentada uma ferramenta de análise qualitativa bastante importante no contexto dos estudos de séries temporais.

4.3 Reconstrução do Atrator

Suponha que a série fosse construída a partir de um sistema cujo comportamento, dinâmica e modelo fossem características totalmente desconhecidas. Então, de posse do vetor série temporal, seria interessante, em um primeiro momento, identificar e comprovar o real comportamento do sistema. A análise pela reconstrução do atrator possibilita verificar o comportamento do sistema no plano ou no espaço de estados a partir da série temporal de uma única variável de estado, permitindo qualificar rapidamente o comportamento do sistema. Existem algumas metodologias propostas para a realização desse tipo de análise, tais como Decomposição por Valores Singulares (*Singular Value Decomposition* - SVD) (Albano *et al.*, 1988; Darbyshire e Broomhead, 1996; Banbrook, 1996), e Método de Atraso no Tempo (*Time-Delay Method*) (Broomhead e King, 1986; Brown *et al.*, 1991; Abarbanel, 1996; Sprott, 2003) sendo essa a metodologia que foi considerada neste trabalho de dissertação visto que são utilizados alguns de seus conceitos durante o procedimento de cálculo do maior expoente de *Lyapunov* apresentada por Wolf *et al.* (1985) e que será discutido na Seção 4.4.

A reconstrução do atrator consiste de vários procedimentos que devem ser adotados. Basicamente, o objetivo é transformar o vetor X em uma matriz cujas colunas representam, cada uma delas, dimensões do espaço de estado euclidiano $\mathbb{R}^{d_E}$ (Broomhead e King, 1986). Para melhor compreensão, inicialmente, considera-se que $t_{j+1} = t_j + j\tau$ onde τ foi convencionado como sendo o tempo de atraso de *Takens* (Takens, 1981), sendo um múltiplo do período de amostragem considerado. Portanto, qualquer ponto do vetor X pode ser genericamente escrito na forma representada pela expressão (4.2),

$$x(t_{j+n}) = x(t_j + n\tau) \quad (4.2)$$

Dessa forma, podem-se obter “janelas” do vetor X que permitem a visualização de alguns elementos da série. Essas janelas definem os vetores linha $z(j)$, expressão (4.3),

$$z(j) = \begin{bmatrix} x(t_j) & x(t_j + \tau) & x(t_j + 2\tau) & \dots & x(t_j + (d_E - 1)\tau) \end{bmatrix} \quad (4.3)$$

os quais formam a matriz de trajetória do espaço reconstruído, sendo simbolizada por Z e definida pela expressão (4.4),

$$Z = \begin{bmatrix} x(t_0) & x(t_0 + \tau) & x(t_0 + 2\tau) & \dots & x(t_0 + (d_E - 1)\tau) \\ x(t_1) & x(t_1 + \tau) & x(t_1 + 2\tau) & \dots & x(t_1 + (d_E - 1)\tau) \\ & & \vdots & & \\ x(t_j) & x(t_j + \tau) & x(t_j + 2\tau) & \dots & x(t_j + (d_E - 1)\tau) \\ & & \vdots & & \end{bmatrix} \quad (4.4)$$

De posse da matriz de trajetória Z , pode-se reconstruir o atrator estranho no espaço de estados d_E -dimensional, porém alguns cuidados devem ser tomados. Uma das mais importantes precauções que se deve tomar diz respeito à escolha da dimensão d_E na qual o atrator será reconstruído. Genericamente, um resultado formal para a escolha dessa dimensão deve-se às proposições apresentadas por Takens (1981), as quais dizem respeito ao fato de que, se d_A é a dimensão do atrator original do sistema, então uma condição suficiente para a escolha de d_E é dada por (4.5) (Broomhead e King, 1986; Brown *et al.*, 1991; Parlitz, 1992; Abarbanel, 1996; Nichols e Nichols, 2001; Sprott, 2003),

$$d_E > 2d_A \quad (4.5)$$

O motivo do uso da inequação (4.5) está relacionada ao fato de que se d_E não é escolhido grande o suficiente, então, o atrator reconstruído pela metodologia de atraso no tempo pode cruzar a si mesmo em certas regiões do espaço. Por exemplo, objetivando uma melhor compreensão, escolhe-se ao acaso um determinado ponto no espaço de estados reconstruído. Se a relação (4.5) não é respeitada, esse ponto em questão talvez pertença à região de cruzamento do espaço de estados original e, nessa situação, os seus vizinhos no novo espaço de estado podem, na verdade, serem pontos opostos do atrator no espaço original. Essa situação representa o que

é conhecido como vizinhos falsos (*false neighbors*) (Brown *et al.*, 1991). Além disso, a escolha da dimensão d_E pode resultar no aparecimento de um ou mais λ no espectro de *Lyapunov* os quais, na realidade, não existem (Stoop e Parisi, 1991; Parlitz, 1992).

Outro cuidado que deve ser tomado durante o processo de reconstrução do espaço de estados diz respeito à escolha do tempo de atraso ou tempo de *Takens*, τ , intimamente relacionado à quantidade média de informação (*Average Mutual Information*) entre pontos próximos no vetor série temporal. Uma das justificativas do procedimento de reconstrução do atrator baseia-se na característica de que o estado atual de qualquer uma das variáveis de estado (por exemplo, $x(t_j)$) carrega informação do estado anterior ($x(t_{j-1})$) e, por essa razão, dependendo do tempo de atraso escolhido, o atrator reconstruído pode apresentar deformações devido à perdas de informação (Fraser e Swinney, 1986; Fraser, 1989; Brown *et al.*, 1991; Abarbanel, 1996; Nichols e Nichols, 2001; Sprott, 2003). Com o objetivo de ilustrar, na Figura 4.1 são apresentados alguns atratores reconstruídos para o sistema de *Lorenz* (3.1),

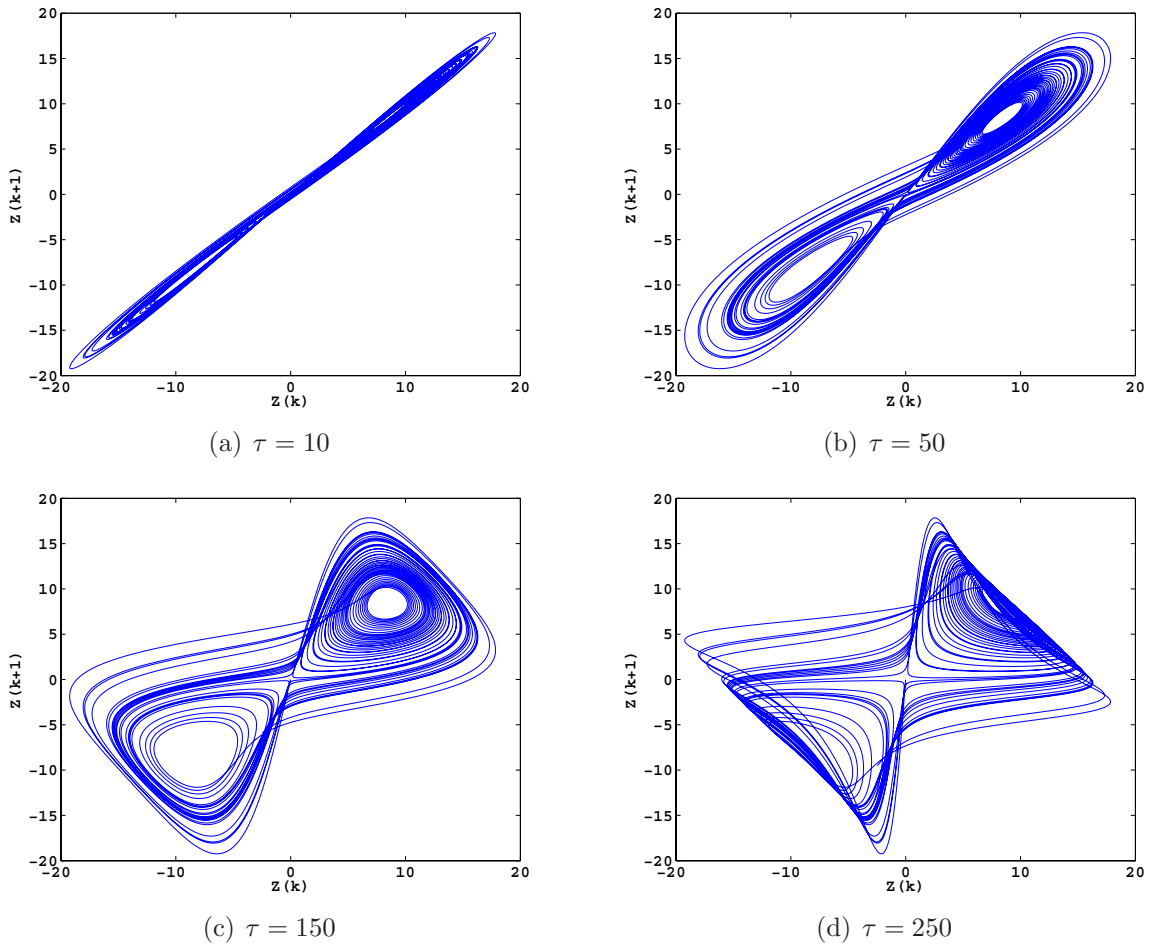


Figura 4.1: Atratores reconstruídos a partir da série temporal para o modelo de *Lorenz* (3.1).

O sistema de *Lorenz* (3.1) foi escolhido por ser bastante difundido, e, conseqüentemente, existem resultados já publicados na literatura (Abarbanel, 1996) que auxiliam nas análises e comparações. A partir dos atratores apresentados na Figura 4.1, pode-se perceber como o valor de τ afeta o procedimento de reconstrução. Para um valor de $\tau = 10$, Figura 4.1(a), o atrator ficou “achatado”. Isso representa que o sistema de *Lorenz* (3.1) não gerou uma quantidade de informação suficientemente significativa. No caso do atrator da Figura 4.1(b), $\tau = 50$, o atrator reconstruído assemelha-se bastante ao resultado obtido para o sistema original, Figura 3.3(a). Para a situação em que $\tau = 150$, Figura 4.1(c), o atrator apresenta indícios de deformação que são evidenciados no caso de $\tau = 250$, Figura 4.1(d). Para esse último caso, a matriz de trajetória Z foi construída utilizando-se a formulação (4.4), porém, informações foram perdidas no processo. Para esse sistema, foi considerado $d_E = 3$ em todas as simulações, além de integração numérica *Runge-Kutta* de 4ª ordem com passo fixo de 0.01. As condições iniciais utilizadas em todos os casos foram $x_0 = 0.0$, $y_0 = -0.01$, $z_0 = 9.0$ e parâmetros dados por $\sigma = 10$, $r = 28$ e $b = \frac{8}{3}$ e o método de integração numérico utilizado foi o *Runge-Kutta* de 4ª ordem, com passo fixo de 0.001.

4.4 Obtenção dos Expoentes de *Lyapunov*

A reconstrução do atrator é interessante do ponto de vista qualitativo da análise do sistema. Contudo, são necessárias análises quantitativas, e, para isso, pode-se analisar as características consideradas invariantes do sistema, como dimensões fractais do atrator e os expoentes e dimensões de *Lyapunov* (Abarbanel, 1996).

O cálculo dos expoentes de *Lyapunov*, quando as equações dinâmicas do sistema são conhecidas, pode ser realizado por diversos métodos, sendo que um deles é conhecido por utilizar matriz Jacobiana e ortonormalização de *Gram-Schmidt* (Wolf *et al.*, 1985; Parker e Chua, 1989; Ramasubramanian e Sriram, 2000), mas, apesar de bastante eficaz, é muito custosa, sob o contexto computacional (Nogueira, 2001). No trabalho apresentado por Rangarajan *et al.* (1998) tem-se uma metodologia onde se aplicam matrizes ortogonais e decomposições do mapa tangente para contornar a necessidade da ortonormalização durante o procedimento do cálculo dos expoentes de *Lyapunov*.

Neste trabalho de dissertação, objetivou-se calcular os expoentes a partir da série temporal obtida, possibilitando, assim, quantificar o comportamento do sistema dinâmico implementado no FPAA, que será discutido no Capítulo 6. Para isso e durante os estudos realizados, a metodologia apresentada por Wolf *et al.* (1985) foi utilizada objetivando-se calcular o maior

expoente de *Lyapunov* a partir de uma série temporal de dados. Dessa maneira, caso o maior λ calculado fosse positivo, então ficaria comprovado o comportamento caótico do sistema que originou a série temporal.

Segundo Wolf *et al.* (1985), para que fosse possível estimar λ_1 , definido como o maior expoente de *Lyapunov*, a partir da série temporal, seria necessário monitorar a evolução de um par de órbitas do atrator que fossem próximas. Pela utilização da reconstrução do espaço de estados por atraso no tempo, pode-se simular duas trajetórias, a fiducial e a perturbada, a partir de uma única série de dados. Para isso, escolhe-se pontos na série cuja separação no tempo é de, no mínimo, um período na órbita do atrator pois, caso esses pontos escolhidos estejam temporalmente muito próximos, isso seria caracterizado como um expoente de *Lyapunov* nulo. Os dois pontos da série escolhidos distantes temporalmente na série original de D posições, Figura 4.2,

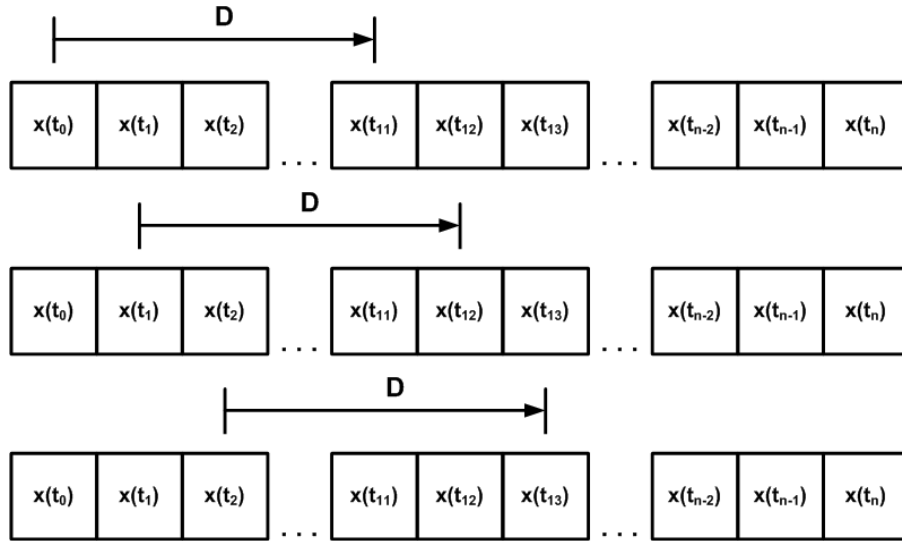


Figura 4.2: Ilustração do procedimento de simulação das trajetórias fiducial e secundária.

são considerados para definir a distância inicial entre a trajetória fiducial e a trajetória secundária, simbolizada por $L(t_i)$. No algoritmo, apresentado no Apêndice A, essa distância mínima é representada pela variável SCALMN. A separação D é mantida conforme a série temporal é percorrida, ou seja, ao longo das evoluções do algoritmo (simbolizada pela variável EVOLV), até que a máxima distância entre as trajetórias, definida previamente por $L'(t_i)$, e simbolizada por SCALMX, seja alcançada, Figura 4.3.

Quando a separação das trajetórias alcança o limite pré-estabelecido, $L'(t_i)$, passa a ser interessante realizar a ortogonalização de *Gram-Schmidt* no vetor que define $L'(t_i)$,

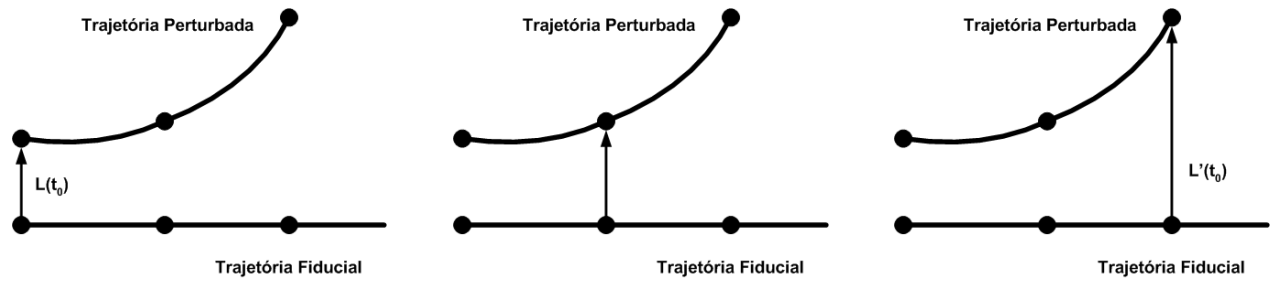


Figura 4.3: Ilustração do procedimento de simulação das trajetórias fiducial e secundária (continuação).

simplesmente normalizando-o. Nesse momento do procedimento, tem-se o par de pontos que representa $L'(t_i)$, estando simbolizados na Figura 4.4 nas cores verde (origem) e amarelo (destino). O ponto que pertence à trajetória secundária é descartado e a série temporal original é novamente varrida em busca do ponto (em azul) mais próximo ao ponto da trajetória fiducial (destacado em verde) mas que respeite a distância mínima SCALMN. Esse processo é repetido até que o final da série temporal seja atingido.

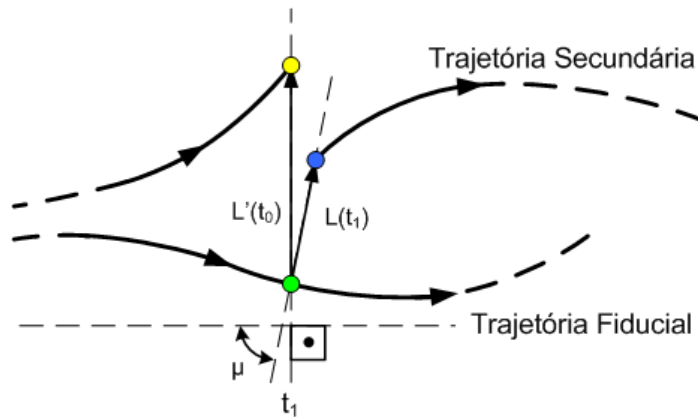


Figura 4.4: Ilustração do procedimento de escolha dos pontos durante o cálculo do maior expoente de *Lyapunov*.

Outros métodos para calcular os expoentes de *Lyapunov* a partir de séries temporais podem ser encontrados em trabalhos tais como o apresentado por Sano e Sawada (1985), por Darbyshire e Broomhead (1996) e por Banbrook (1996). A grande vantagem das metodologias propostas consiste na possibilidade de calcular-se não somente o maior expoente de *Lyapunov* mas todo o espectro de *Lyapunov* ficando permitido o estudo de outra característica do sistema denominada dimensão de *Lyapunov* que representa a dimensão fractal do atrator (Abarbanel, 1996).

Capítulo 5

Ambiente de Desenvolvimento

5.1 Considerações Iniciais

Neste capítulo, são apresentadas as principais características da ferramenta disponível no MATLAB conhecida como *System-Function*, utilizada para estudo dos modelos dinâmicos considerados durante os experimentos realizados. Além disso, é realizada uma discussão sobre a metodologia de aquisição de dados do sistema dinâmico implementado em *hardware* a partir do *toolbox* do MATLAB conhecido como XPC TARGET.

5.2 Sobre o Simulink

O SIMULINK é um *software*, integrado ao MATLAB, que serve para modelar, simular e analisar sistemas dinâmicos lineares e não-lineares. Além disso, o modelamento pode ser realizado em tempo contínuo, em tempo discreto ou de forma híbrida, isto é, parte do sistema sendo executado em tempo discreto e parte em tempo contínuo. Os sistemas implementados também podem ser desenvolvidos utilizando-se diversas frequências de execução em partes distintas do modelo (Nogueira, 2001).

Para o modelamento, o SIMULINK dispõe de uma interface gráfica (*Graphical User Interface* - GUI) que permite a construção de diagramas de blocos a partir da biblioteca padrão de componentes do próprio *software*. Quando desejável e/ou necessário, é possível a construção de blocos específicos para a execução de uma determinada função. O desenvolvimento dos blocos personalizados é realizado a partir de uma ferramenta disponível no MATLAB e no

SIMULINK denominada *System-Function*.

5.3 As *System-Functions*

Uma *System-Function*, ou, como será tratada daqui em diante neste texto, uma *s-function*, consiste em uma linguagem de descrição do próprio SIMULINK, podendo ser desenvolvida a partir de linguagens de programação tais como C/C++, Fortran e a própria linguagem do MATLAB. A sua chamada é realizada a partir de blocos homônimos disponíveis na biblioteca padrão do SIMULINK (Nogueira, 2001).

Quando uma *s-function* é desenvolvida utilizando-se linguagem C/C++ ou Fortran, a *s-function* passa a ser denominada de *MEX-file*. Se a linguagem de programação escolhida foi a do MATLAB, então a *s-function* passa a ser denominada *M-file function*. A principal diferença entre essas duas modalidades de *s-function* consiste no fato de que as *MEX-functions* são compiladas, enquanto as *M-functions*, que são interpretadas. A vantagem é que programas compilados rodam muito mais rápido do que os programas interpretados por serem previamente otimizados (Schildt, 1997; Nogueira, 2001).

Outra grande vantagem está no fato de que as *MEX-functions* possuem funções específicas que podem ser utilizadas, por exemplo, para *backup* de dados. Essa possibilidade permite que eventuais interrupções das simulações (falta de energia, travamento do computador) possam ser contornadas fazendo com que o sistema possa ser reiniciado a partir das condições imediatamente anteriores à interrupção. Nogueira (2001) apresenta um apêndice com um modelo de *s-function* que foi utilizado como base para as *s-functions* desenvolvidas ao longo desse trabalho de dissertação.

Uma possibilidade quanto à utilização do MATLAB e do SIMULINK está relacionada à edição de máscaras que permitem maior automação na execução das simulações e rapidez quanto à alteração de determinados parâmetros do modelo implementado. Na Figura 5.1 tem-se um exemplo dessa possibilidade.

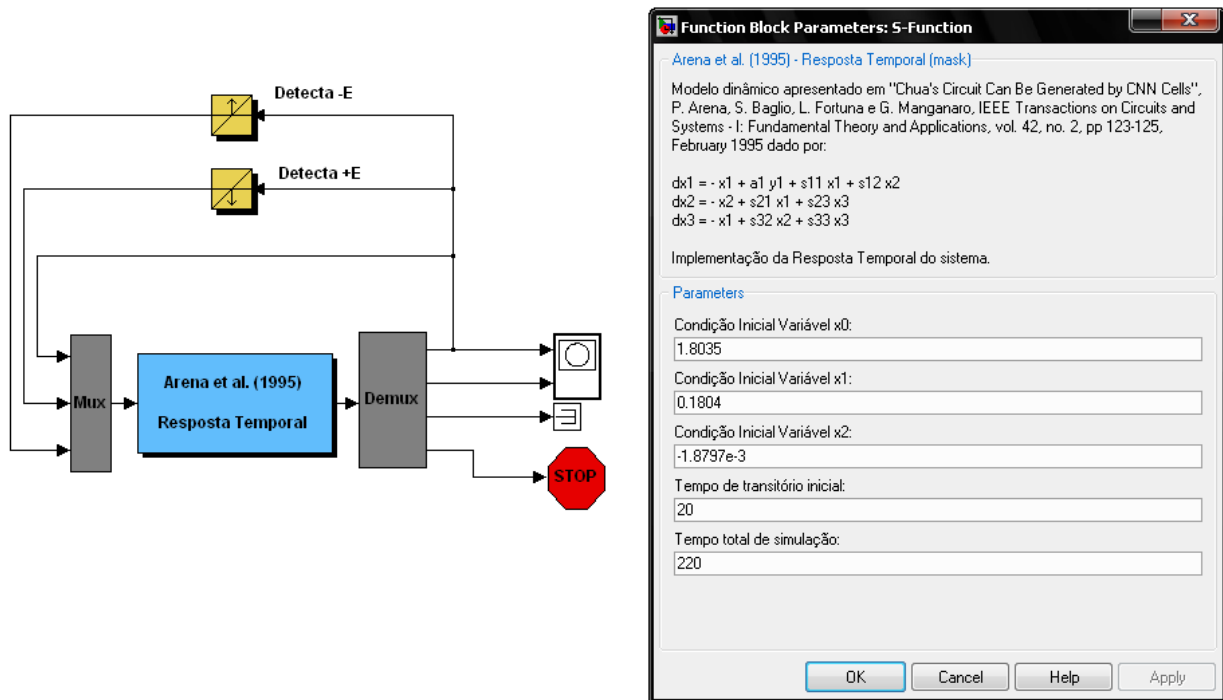


Figura 5.1: Modelo (6.5) implementado utilizando-se *s-function* e a máscara desenvolvida para a alteração dos parâmetros do modelo.

5.4 xPC Target

O xPC TARGET¹ consiste de um *toolbox* do MATLAB que, dentre as muitas possibilidades, permite a simulação de diagramas de blocos construídos no SIMULINK a partir de um *kernel* de tempo real desenvolvido pela própria MATHWORKS. Além disso, possibilita a incorporação aos diagramas implementados, blocos relativos às placas de interfaceamento para, por exemplo, a aquisição de dados.

Em um esquema baseado no *toolbox* xPC TARGET, os computadores utilizados são computadores pessoais comuns interconectados em configuração do tipo mestre/escravo. No computador mestre (*host*, como será tratado daqui por diante), implementa-se um diagrama em blocos no SIMULINK pelo uso de blocos da sua biblioteca padrão de componentes ou blocos personalizados desenvolvidos em *s-function*, por exemplo, representando um sistema dinâmico. Finalizada a construção do diagrama, compila-se o mesmo através de *softwares* específicos para essa finalidade, tais como o Open Watcom² versão 1.3 e o Microsoft Visual Studio 6.0 ou 2003.

¹Maiores informações em <http://www.mathworks.com/access/helpdesk/help/helpdesk.html>

²Compilador gratuito. Pode ser adquirido em http://www.openwatcom.org/index.php/Main_Page.

Após compilado, o modelo é carregado no computador escravo (ou *target*) via protocolo de comunicação RS232 ou TCP/IP, onde um *kernel* de tempo real já está em execução. Na Figura 5.2 tem-se uma ilustração para o caso de o esquema xPC TARGET utilizar o protocolo de comunicação TCP/IP.

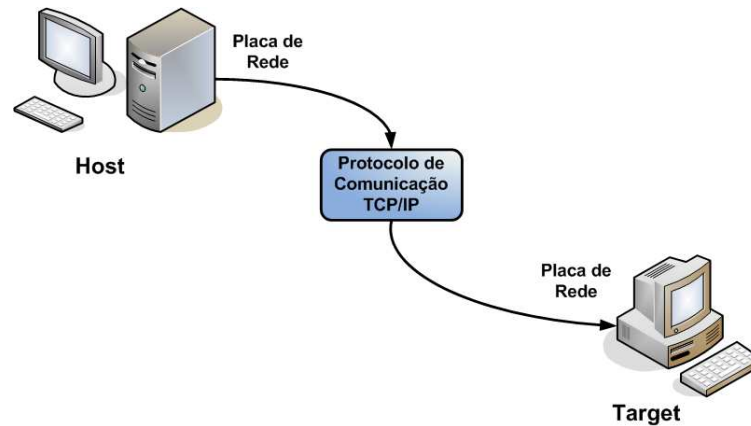


Figura 5.2: Diagrama do funcionamento do xPC TARGET utilizando o protocolo de comunicação TCP/IP.

Uma das vantagens em utilizar o esquema xPC TARGET consiste em possibilitar a execução de modelos em um ambiente de tempo real, com *hardware* totalmente dedicado à execução da tarefa. Isso é importante no contexto de validação de sistemas em desenvolvimento que venham a ser utilizados em *hardware* específico, tais como microcontroladores e DSPs.

Outra vantagem do xPC TARGET consiste em possibilitar a utilização de placas de aquisição de sinais no modelo implementado a partir de blocos da biblioteca padrão de componentes. Esses blocos correspondem às *s-functions* desenvolvidas pela própria MATHWORKS, consistindo de *drivers* das placas. Como grande parte das funções do MATLAB podem ser alteradas, dependendo da aplicação, esses *drivers* podem ser modificados, compilados e incorporados ao sistema (Nicolato, 2007). Os dados adquiridos podem, então, ser salvos em discos no próprio *target* e transferidos ao *host*, onde serão estudados utilizando-se ferramentas disponíveis no próprio MATLAB.

O esquema do xPC TARGET foi utilizado no desenvolvimento desse trabalho pois havia a necessidade de:

1. Utilização de uma placa de aquisição de dados que possuisse conversores A/D;
2. Gravação, em disco, dos dados adquiridos.

Dessa maneira, foi utilizada a placa de aquisição CIO-DAS16/330, da MEASUREMENT COMPUTING, com conversor A/D de 12 *bits* trabalhando a uma taxa de conversão máxima de 330 kHz, Figura 5.3. Os dados adquiridos foram salvos em disco no computador *target* e, após finalizada a execução do sistema, os arquivos de dados foram transmitidos ao computador *host* através do protocolo de comunicação TCP/IP.

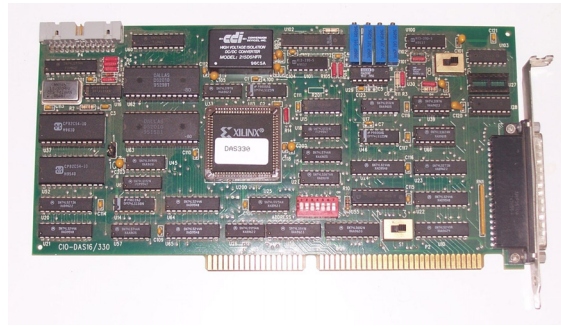


Figura 5.3: Placa de aquisição de dados utilizada.

No Capítulo 6 são apresentados maiores detalhes dos procedimentos experimentais realizados, bem como a descrição da montagem da configuração xPC TARGET adotada para a aquisição dos sinais do modelo implementado no FPAA.

Capítulo 6

Resultados Experimentais

6.1 Considerações Iniciais

Este capítulo tem como principal objetivo apresentar os procedimentos e os resultados experimentais obtidos pela implementação, em *hardware*, do sistema do Circuito de *Chua*. Além disso, são realizados estudos qualitativos, pelo uso da metodologia baseada no *toolbox* XPC TARGET, e quantitativos, pela utilização das ferramentas de análises baseadas em séries temporais.

6.2 Análises Teóricas e Implementações Computacionais

Para a realização dos experimentos, decidiu-se por implementar o modelo matemático representativo do Circuito de *Chua* (Chua, 1993; Liu *et al.*, 2007), sendo que o trabalho apresentado por Caponetto *et al.* (2005) serviu como base teórica e prática para os experimentos realizados durante esse trabalho de mestrado. Para a implementação do modelo em questão, utilizou-se o *kit* de desenvolvimento AN221K04 - ANADIGMVORTEXTM, introduzido no Capítulo 2.

O circuito tradicional de *Chua* é um circuito autônomo não-linear e de terceira ordem, sendo considerado um paradigma para o Caos, em virtude de possibilitar, pelo simples ajuste de parâmetros, a observação de uma vasta gama de fenômenos, com uma família de atratores estranhos bem como bifurcações e rotas para o caos (Chua, 1993; Shil'nikov, 1993; Liu *et al.*, 2007). Uma característica importante desse circuito é que o comportamento tensão $\times$ corrente

do Diodo de *Chua*¹ pode ser caracterizado por qualquer função não-linear como, por exemplo, uma função linear por partes, Figura 6.1(b), ou uma função polinomial cúbica do tipo $f(x) = c_0x + c_1x^3$ (Shil'nikov, 1993). Na Figura 6.1(a), tem-se um esquemático ilustrativo deste circuito.

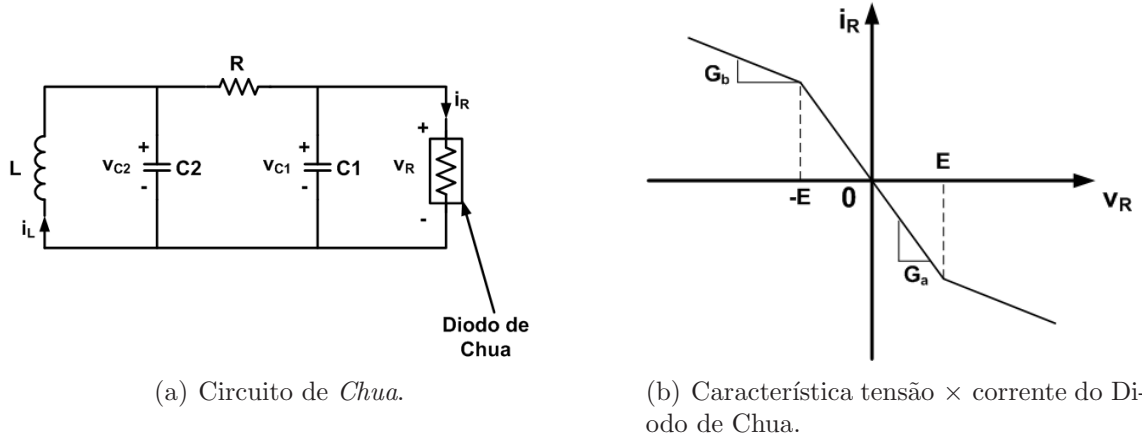


Figura 6.1: Representação do Circuito de *Chua* em sua forma original e a característica tensão × corrente mais usual do Diodo de *Chua*, linear por partes.

O Circuito de *Chua* tornou-se um paradigma muito difundido para pesquisas em caos por ser de simples confecção e possibilitar experimentos em laboratório, utilizando componentes analógicos discretos básicos tais como resistores, capacitores, indutores, amplificadores operacionais (Matsumoto *et al.*, 1985) e transistores bipolares (Matsumoto *et al.*, 1986). Além disso, o circuito pode ser simulado computacionalmente por possuir um modelo matemático também simples baseado em equações diferenciais ordinárias. As equações diferenciais apresentadas em (6.1) formam o modelo do circuito de *Chua* em função dos seus parâmetros C_1 , C_2 , L e R ,

$$\begin{aligned}
 C_1 \frac{dv_{c1}}{dt} &= \frac{1}{R} v_{c2} - v_{c1} - g(v_{c1}) \\
 C_2 \frac{dv_{c2}}{dt} &= \frac{1}{R} v_{c1} - v_{c2} - i_L \\
 L \frac{di_L}{dt} &= -v_{c2}
 \end{aligned} \tag{6.1}$$

onde v_{c1} , v_{c2} e i_L são a tensão sobre o capacitor C_1 , a tensão sobre o capacitor C_2 e a corrente do indutor L , respectivamente, e $g(v_{c1})$ corresponde à característica tensão × corrente do resistor não-linear (Chua, 1993; Liu *et al.*, 2007), dada por (6.2):

¹Componente não-linear do Circuito de *Chua*.

$$g(v_{c1}) = G_1 v_1 + \frac{1}{2}(G_a - G_b)(|v_1 + E| - |v_1 - E|) \quad (6.2)$$

Para simplificar, o sistema (6.1) pode ser reescrito de maneira a obter-se um sistema de equações diferenciais adimensional. Esse novo formato do modelamento matemático do Circuito de *Chua* é representado pelo sistema (6.3) (Shil'nikov, 1993),

$$\begin{aligned} \dot{x} &= \alpha[y - h(x)] \\ \dot{y} &= x - y + z \\ \dot{z} &= -\beta y - \gamma z \end{aligned} \quad (6.3)$$

onde a característica não-linear do sistema, $h(x)$, é representada pela equação (6.4),

$$h(x) = m_1 x + \frac{1}{2}(m_0 - m_1)(|x + 1| - |x - 1|) \quad (6.4)$$

sendo x , y , z as variáveis de estado e α , β , γ , m_0 e m_1 parâmetros do sistema.

Para comprovar o comportamento caótico intrínseco ao Circuito de *Chua*, a metodologia proposta em (Arena, Baglio, Fortuna e Mangarano, 1995) apresenta um desenvolvimento teórico objetivando reestruturar o circuito. Para isso, os autores implementam o circuito a partir de componentes analógicos discretos tais como resistores, capacitores e amplificadores operacionais. O modelo obtido a partir dessa reestruturação está apresentado em (6.5),

$$\begin{aligned} \dot{x}_1 &= -x_1 + a_1 y_1 + s_{11} x_1 + s_{12} x_2 \\ \dot{x}_2 &= -x_2 + s_{21} x_1 + s_{23} x_3 \\ \dot{x}_3 &= -x_3 + s_{32} x_2 + s_{33} x_3 \end{aligned} \quad (6.5)$$

onde x_1 , x_2 e x_3 correspondem às variáveis de estado e y_1 é a variável de saída do circuito. A equação que modela y_1 é dada pela expressão (6.6),

$$y_1 = 0.5(|x_1 + 1| - |x_1 - 1|) \quad (6.6)$$

Comparando-se o sistema (6.3) com o sistema (6.5), obtém-se os valores das constantes de (6.5): $a_1 = \alpha(m_1 - m_0)$, $s_{33} = 1 - \gamma$, $s_{21} = s_{23} = 1$, $s_{11} = 1 - \alpha m_1$, $s_{12} = \alpha$ e $s_{23} = -\beta$. Caponetto *et al.* (2005) utiliza os seguintes valores numéricos para as constantes: $\alpha = 9$, $\beta = 14.286$, $\gamma = 0$, $m_0 = -\frac{1}{7}$ e $m_1 = \frac{2}{7}$.

A título de ilustração, o sistema de equações (6.5) foi simulado através do desenvolvimento de uma *s-function* implementando esse modelo. Os parâmetros considerados nesta simulação foram os mesmos apresentados por Arena, Baglio, Fortuna e Mangarano (1995) e o resultado obtido está apresentado na Figura 6.2.

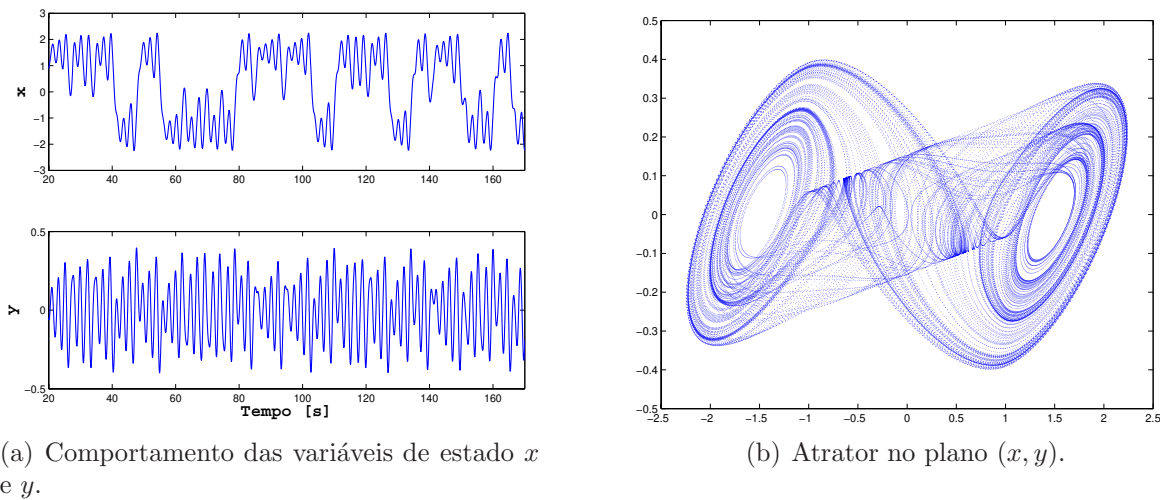


Figura 6.2: Atrator e resposta temporal obtidos pela implementação do modelo (6.5) pela utilização da ferramenta *s-function*.

Para que fosse possível a obtenção do atrator apresentado na Figura 6.2(b), utilizou-se um microcomputador com processador AMD SEMPRON 2800+ (256 kbytes de memória *cache* L2, soquete de 754 pinos), disco rígido Samsung SATA II, 1 GB de memória RAM (*Random Access Memory*) e placa de vídeo *off-board* com processador gráfico dedicado da ATI com 256 MB de memória. O passo de integração utilizado foi de 0.01 e o tempo total de simulação foi, de aproximadamente, 7 minutos. Contudo, alterando-se o passo de integração para 0.001, o tempo de simulação necessário para a resolução do sistema (6.5) foi de, aproximadamente, 7 horas.

Fazendo-se uma análise mais aprofundada, é fácil perceber que, realmente, para bem simular este circuito seria necessário usar passos de integração ainda menores do que 10^{-4} , pois, dependendo dos valores dos parâmetros utilizados, variações nesta ordem de grandeza podem produzir bifurcações tanto de mudanças de periodicidade nas transições entre oscilações periódicas para caos como vice-versa. Dessa maneira, tornam-se notáveis os problemas oriundos da computação apenas digital.

6.3 Procedimentos Experimentais

6.3.1 Considerações sobre a Implementação em *Hardware*

Como pôde ser constatado, o tempo necessário para a obtenção dos resultados pela simulação computacional de modelos representativos de sistemas dinâmicos operando em caos é algo bastante significativo. As dificuldades relacionadas ao tempo de execução computacional comprometem o estudo de sistemas de dinâmica caótica. Segundo Nogueira (2001), utilizando-se um microcomputador com processador *Pentium* III com *clock* de 450 MHz, o tempo médio para a realização das experimentos propostos foi de 11 dias de simulação contínua. Para possibilitar a análise dos resultados intermediários durante o processo de cada uma das simulações, foi adotado o procedimento de impressão dos pontos em gráficos mantidos na tela do computador para a realização de análises preliminares. Conseqüentemente, essa solução acarreta em um decréscimo no desempenho global do computador devido a grande quantidade de informação decorrente de cada um dos experimentos realizados.

A diminuição do tempo de execução das simulações de sistemas caóticos pode ser conseguida empregando-se componentes analógicos. Nesse contexto, foram utilizados os dispositivos FPAAs para a implementação do modelo do Circuito de *Chua* com o objetivo de reduzir o tempo necessário de execução para a obtenção de resultados, tais como o atrator estranho, objetivando estudar a dinâmica desse circuito. Na Figura 6.3, está ilustrado como foi feita a implementação desse circuito a partir do ambiente de desenvolvimento ANADIGMDESIGNER® EDA.

A Tabela 6.1 apresenta os valores dos principais parâmetros das CAMS numeradas na Figura 6.3 e que foram utilizadas na implementação do modelo do Circuito de *Chua* no FPAA. Para todas estas CAMS, a frequência de operação utilizada foi de 250 kHz.

Basicamente, o desenvolvimento do modelo implementado no FPAA baseia-se em

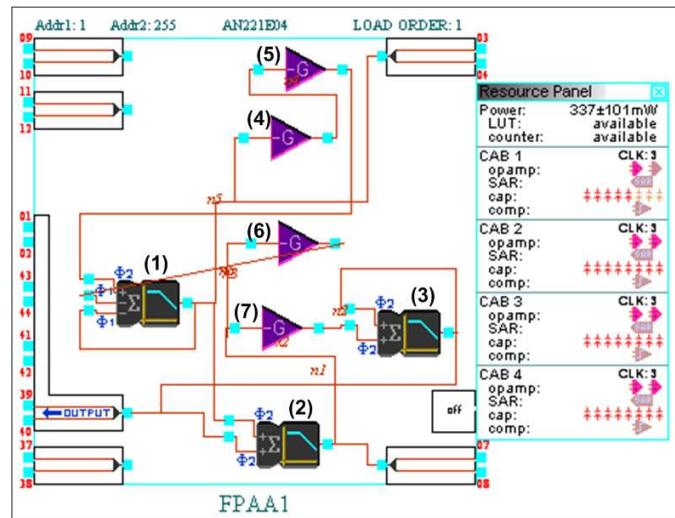


Figura 6.3: Circuito de *Chua* (6.5) implementado no FPA1.

Tipo da CAM	Número da CAM	Parâmetro Ajustado	Valor do Parâmetro
SUMFILTER	1	Frequência de Corte	0.4
		Ganho 1	2
		Ganho 2	1
		Ganho 3	1.5
	2	Frequência de Corte	0.4
		Ganho 1	2
		Ganho 2	2
	3	Frequência de Corte	0.4
		Ganho 2	1
GAININV	4	Ganho	4
	5	Ganho	0.5
	6	Ganho	g
	7	Ganho	6

Tabela 6.1: Parâmetros para as CAMs utilizadas na implementação do modelo de *Chua*.

duas CAMs, a de ganho inversor (GAININV) e a de somador (SUMFILTER). O ponto principal dessa implementação está no fato de que o bloco SUMFILTER representa um integrador e, além disso, permite o ajuste de ganhos internos, possibilitando a eliminação de outros blocos do tipo GAININV. Dessa maneira, a quantidade de CAMs necessárias para a construção do modelo do Circuito de *Chua* no FPA1 fica bastante reduzido (Caponetto *et al.*, 2005). Além disso, vale observar que o parâmetro g referente ao ganho da CAM de número 6 é o responsável pelo controle do comportamento do sistema, ou seja, conforme esse parâmetro for

variado o sistema pode apresentar comportamentos distintos (estável, periódico, quasi-periódico e caótico) (Parker e Chua, 1989; Fiedler-Ferrara e do Prado, 1994; Nogueira, 2001).

6.3.2 Descrição da Bancada e Resultados Experimentais

Para a realização tanto dos estudos qualitativos quanto dos estudos quantitativos, que serão detalhados na Seção 6.4, os sinais provenientes do *kit* de desenvolvimento referentes ao modelo implementado foram adquiridos utilizando-se o esquema xPC TARGET em conjunto com a placa de aquisição CIO-DAS16/330. Ao final do procedimento de aquisição, os dados, na forma de séries temporais, foram salvos em disco no computador *target* e, em seguida, transferidos ao computador *host* via protocolo TCP/IP. Dessa forma, com a possibilidade de utilização de todas as ferramentas de análise disponíveis no MATLAB, os dados adquiridos puderam ser estudados. A Figura 6.4 mostra um esquemático contendo as interconexões e o fluxo de funcionamento do sistema em bancada experimental.

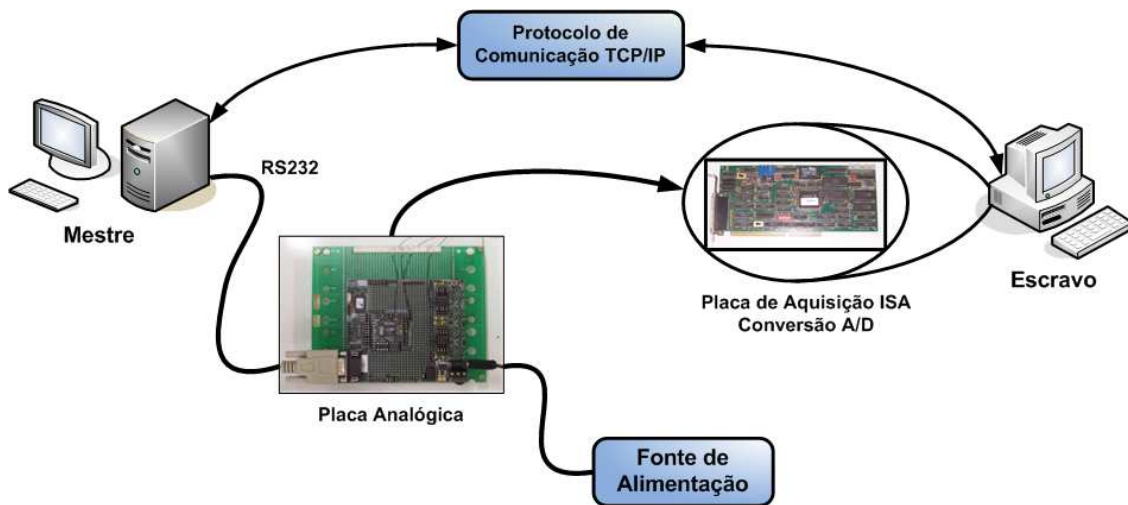


Figura 6.4: Ilustração da configuração da bancada experimental.

Durante a realização de todos os experimentos apresentados neste trabalho utilizando-se esse esquema de montagem, foram utilizados 3 canais da placa de aquisição, sendo, então, adquiridos os sinais das 3 variáveis de estado do modelo, x_1 , x_2 e x_3 . O tempo de amostragem foi definido como sendo $0.0013s$. Esse valor foi o máximo que o computador *target* aceitou, ou seja, frequências de amostragem maiores resultavam em seu travamento. Portanto, ainda poder-se-ia conseguir mais rapidez se fossem utilizados computadores com maior capacidade de processamento.

Com essa metodologia, foi possível a obtenção de alguns resultados qualitativos, via osciloscópio, bastante importantes pela implementação do modelo (6.5) utilizando-se o FPAA. Durante os experimentos realizados, e conforme apresentado por Caponetto *et al.* (2005), o valor do parâmetro g foi variado entre os valores de 2.60 a 3.05 em passos de 0.01. Na Figura 6.5, estão apresentados os resultados obtidos e, através dela, pode-se perceber alguns comportamentos característicos dos sistemas dinâmicos (Nogueira, 2001), tais como ponto de equilíbrio, Figura 6.5(a), ciclo-limite, Figura 6.5(b) e Figura 6.5(c), e comportamento quasi-periódico, Figura 6.5(d). Na Figura 6.5(e) e na Figura 6.5(f), estão apresentados atratores simples e característicos de sistemas operando em caos.

Na Figura 6.6 estão apresentados os resultados obtidos para um caso em que o comportamento caótico fica evidenciado. Nessa situação, quando o parâmetro é $g = 2.91$, tem-se o início da construção do atrator de *Lorenz* e, para a situação em que $g = 2.96$, o atrator ficou melhor definido.

Finalmente, na Figura 6.7(a) está apresentado o comportamento temporal das variáveis x_1 (curva superior) e x_2 obtidas para o caso em que o parâmetro $g = 3.05$. Já na Figura 6.7(b) está mostrado o atrator de *Lorenz* o qual é resultado da implementação do sistema no FPAA e cujo plano de fase foi definido por (x_1, x_2) , comprovando-se, qualitativamente, o comportamento caótico. Pode-se perceber grande semelhança com os resultados obtidos pela simulação computacional, Figura 6.2. Todas as análises realizadas basearam-se nesse mesmo valor do parâmetro g .

Esse experimento foi de grande importância no contexto do desenvolvimento desse trabalho de dissertação. Em função desse resultado, constatou-se que é possível aprofundar os estudos sobre a metodologia proposta por Caponetto *et al.* (2005) e de modo muito mais rápido. Segundo, possibilitou a melhor compreensão das características do Circuito de *Chua*, importante tópico no contexto de sistemas caóticos e de circuitos elétricos não-lineares. Terceiro, permitiu melhor entendimento das ferramentas de desenvolvimento associadas ao *kit* AN221K04 e das possibilidades relacionadas à utilização do dispositivo FPAA quando aplicado aos estudos de sistemas dinâmicos não-lineares. Finalmente, foi comprovado que o estudo de sistemas caóticos pode ser bastante beneficiado por essa tecnologia, permitindo a realização de novos estudos em tempos de desenvolvimento bem inferiores aos encontrados em simulações computacionais que foram e são motivo de intensas pesquisas na atualidade. A título de observação, a obtenção da Figura 6.7(b) deu-se de maneira instantânea após o FPAA ter sido programado, ou seja, não existe nem termo de comparação sobre a rapidez de obtenção de resultados desta técnica em relação àquelas que usam apenas computação digital.

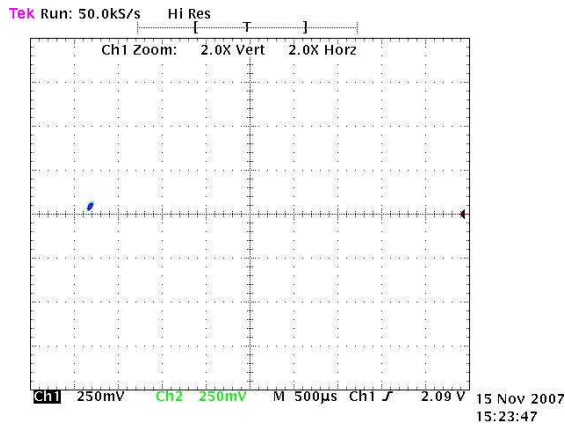
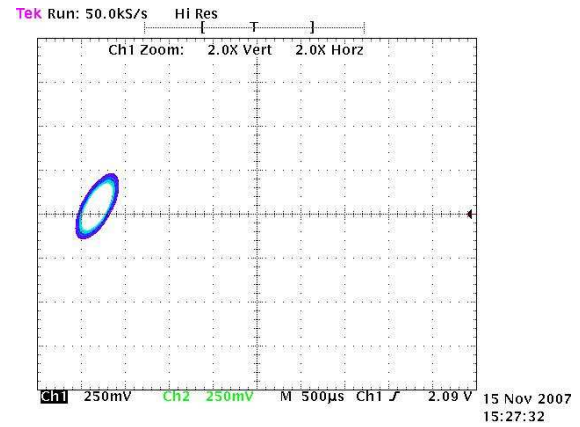
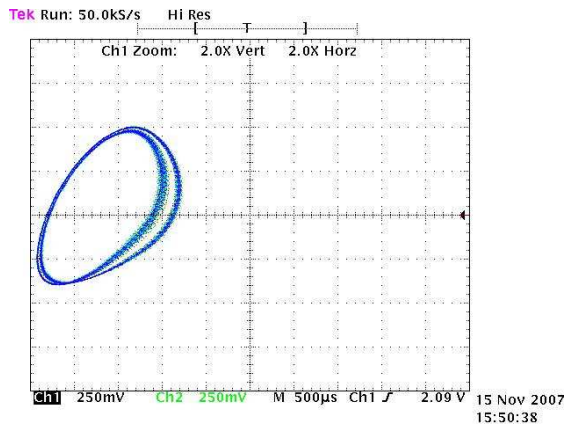
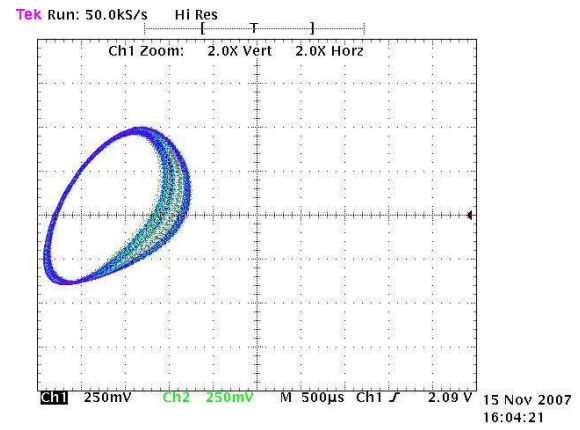
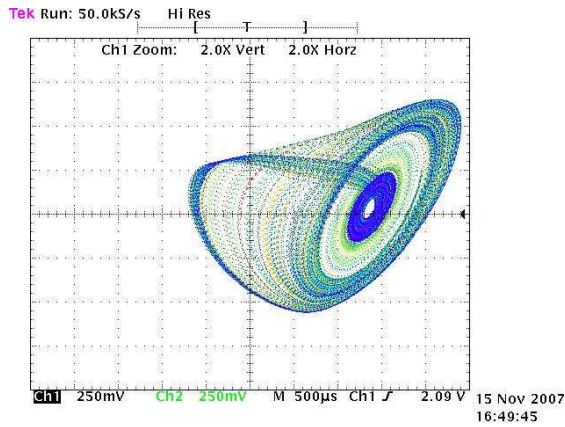
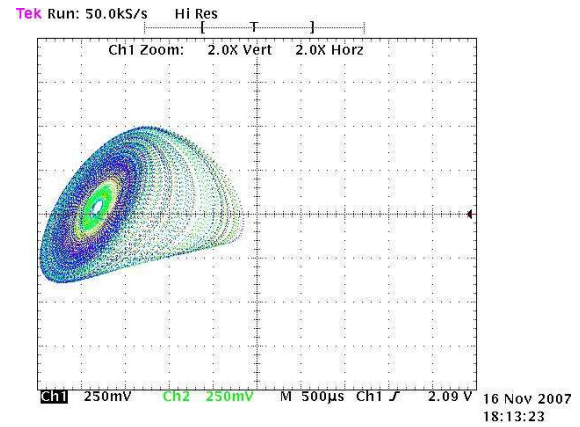
(a) Ponto de equilíbrio: $g = 2.60$.(b) Ciclo limite: $g = 2.62$.(c) Ciclo limite: $g = 2.66$.(d) Quasi-periódico: $g = 2.68$.(e) Atrator simples: $g = 2.73$.(f) Atrator simples: $g = 2.88$.

Figura 6.5: Resultados qualitativos, obtidos via osciloscópio digital com retenção de 6.95s, para o modelo do Sistema de *Chua* implementado no FPAA.

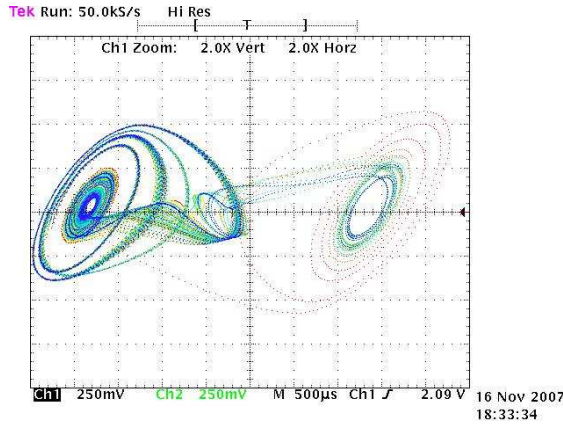
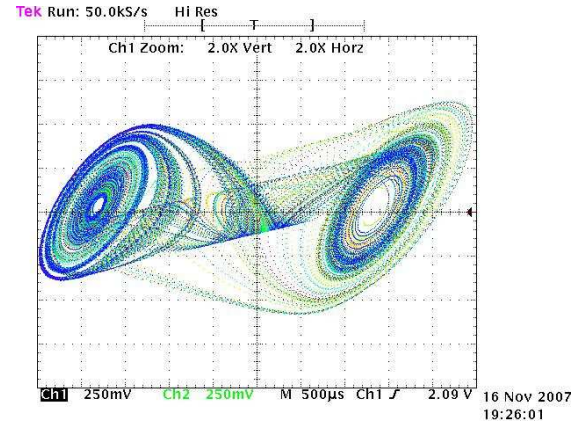
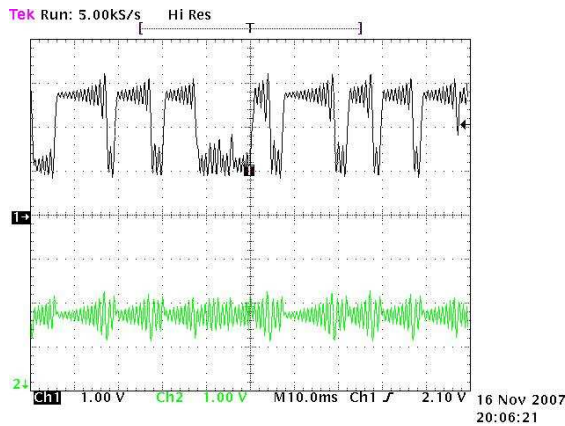
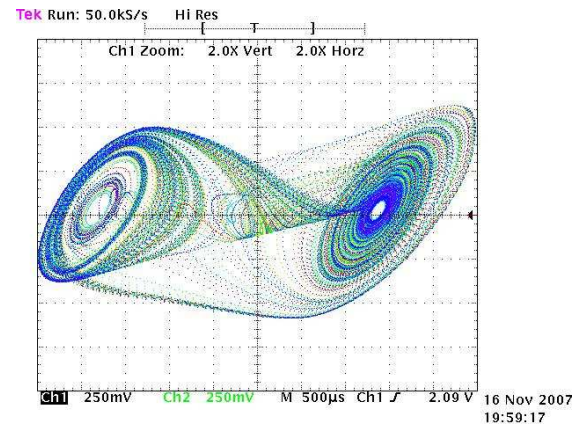
(a) Atrator de Lorenz: $g = 2.91$.(b) Atrator de Lorenz: $g = 2.96$.

Figura 6.6: Resultados qualitativos, obtidos via osciloscópio digital com retenção de 6.95s, para o modelo do Sistema de Chua implementado no FPAA (continuação).



(a) Comportamento das variáveis de estado para o sistema implementado em FPAA.



(b) Atrator obtido utilizando o FPAA conectado ao osciloscópio.

Figura 6.7: Resposta temporal das variáveis de estado x e y e atrator para o Circuito de Chua representado pelo sistema (6.5).

6.3.3 Reconstrução do Atrator

Antes de iniciar os estudos quantitativos das séries temporais obtidas, Seção 6.4, será apresentada a reconstrução do atrator a partir dos dados coletados na situação em que o sistema implementado em *hardware* está operando em caos. Nesse caso, os dados considerados foram adquiridos quando o parâmetro estrutural foi definido como sendo $g = 3.05$. Além disso, o tempo de amostragem foi mantido, ou seja, 0.0013segundos. Finalmente, foram considerados 32768 pontos na série temporal, permitindo que os resultados obtidos apresentassem boa

resolução.

Através da utilização da técnica apresentada no Capítulo 4, foi possível a reconstrução do atrator no plano de fase definido como $(x_1, x_2) = (X(k), X(k+1))$, Figura 6.8, a partir da série temporal gerada pela aquisição do sinal referente à variável de estado $x(t)$ do modelo implementado no FPAA (Caponetto *et al.*, 2005). Para a obtenção desse resultado, o tempo de atraso de *Takens* foi definido como sendo $\tau = 1$. Ao utilizar τ maiores, por exemplo, $\tau = 4$, Figura 6.8(b), foi possível perceber a ocorrência de falsas deformações no espaço de estados. Isto pode ser então usado como informação adicional e critério para ajustes e/ou depuração da programação, no sentido de que valores de τ muito grandes resultam em perda de informação entre os pontos da série (Abarbanel, 1996).

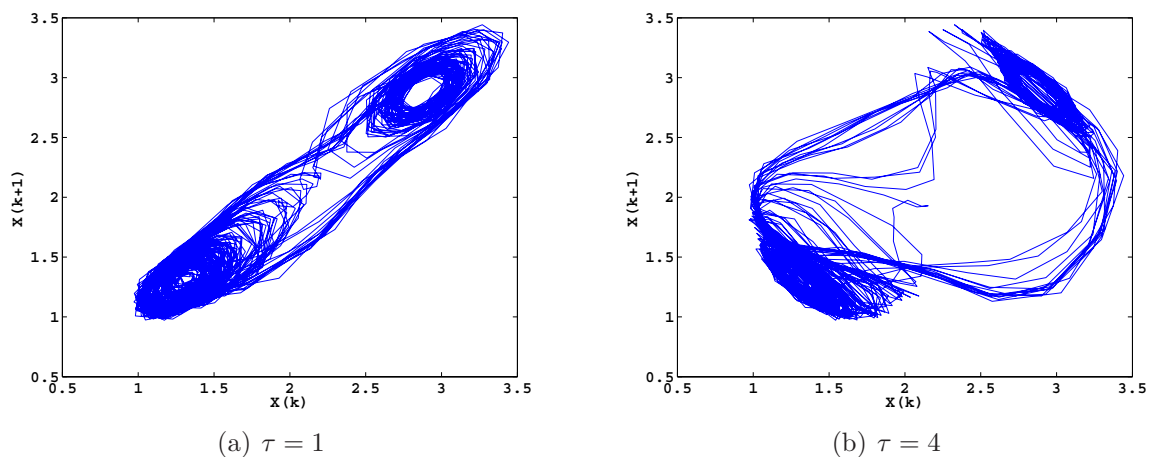


Figura 6.8: Atratores reconstruídos a partir da série temporal.

6.4 Análises Quantitativas dos Resultados

A análise quantitativa dos experimentos realizados baseou-se no cálculo do maior expoente de *Lyapunov* a partir da metodologia apresentada por Wolf *et al.* (1985), mais especificamente no algoritmo apresentado em código FORTRAN, reproduzido no Apêndice A dessa dissertação. Durante as pesquisas realizadas ao longo desse trabalho aqui apresentado, pode-se comprovar que o trabalho de Wolf *et al.* (1985) pode ser considerado uma referência básica no contexto de análise de séries temporais, já que, na maior parte dos artigos pesquisados, por muitas vezes, essa metodologia é referenciada. Por essa razão, as variáveis do algoritmo implementado, tais como SCALMN (distância mínima entre a trajetória fiducial e a trajetória secundária), SCALMX (distância máxima entre a trajetória fiducial e a trajetória secundária)

e EVOLV (número de passos percorridos na série ou, equivalentemente, número de iterações do algoritmo), receberam a mesma nomenclatura apresentada no algoritmo originalmente proposto por tais autores.

Objetivando validar o correto funcionamento do algoritmo implementado, algumas séries temporais foram geradas utilizando-se *s-functions* e modelos dinâmicos bastante conhecidos (Ramasubramanian e Sriram, 2000; Sprott, 2003). Em todos os experimentos, foram considerados 32768 pontos na série temporal, passo fixo de integração de 0.01 e integrador *Runge-Kutta* de 4ª ordem. Uma pequena modificação foi realizada no algoritmo original referente ao uso do logaritmo natural na versão implementada neste trabalho de dissertação em substituição do logaritmo na base 2. Os sistemas implementados estão apresentados na seguinte forma:

Função senoidal: a função seno foi implementada com o objetivo de obter o expoente de *Lya-punov* nulo, pois o sistema é puramente periódico. Para esse sistema, as condições iniciais foram consideradas nulas e os parâmetros do algoritmo foram $\tau = 5$, SCALMN = 0.00, SCALMX = 0.61 e EVOLV = 2.

Mapa de Henón: esse sistema é representado pelas equações (6.7),

$$\begin{aligned} X_{n+1} &= 1 - aX_n^2 + bY_n \\ Y_{n+1} &= X_n \end{aligned} \tag{6.7}$$

cujos parâmetros são definidos como sendo $a = 1.4$, $b = 0.3$, e condições iniciais $X_0 = 0$ e $Y_0 = 0.9$. Para esse sistema, os parâmetros do algoritmo foram $\tau = 1$, SCALMN = 0.18, SCALMX = 0.90 e EVOLV = 3.

Oscilador de Van der Pol: esse sistema é representado pelas equações (6.8),

$$\begin{aligned} \dot{x} &= y \\ \dot{y} &= -x + b(1 - x^2)y + A\sin(\omega t) \end{aligned} \tag{6.8}$$

cujos parâmetros são definidos como sendo $b = 3$, $A = 5$, $\omega = 1.788$, e condições iniciais $x_0 = -1.9$, $y_0 = 0$ e $t_0 = 0$. Para esse sistema, os parâmetros do algoritmo foram $\tau = 20$, SCALMN = 0.10, SCALMX = 1.55 e EVOLV = 7.

Modelo de *Lorenz*: a representação desse modelo é dada pelo sistema de equações (3.1). Os parâmetros para esse sistema foram definidos como $\sigma = 10$, $r = 28$, $b = \frac{8}{3}$, e condições iniciais $x_0 = 0$, $y_0 = -0.01$ e $z_0 = 9$. Para esse sistema, os parâmetros do algoritmo foram $\tau = 4$, SCALMN = 0.25, SCALMX = 1.45 e EVOLV = 2.

Sistema Hipercaótico de *Rössler*: o sistema é representado pelas equações (3.3), onde os parâmetros do sistema foram definidos como $a = 0.25$, $b = 3.0$, $c = 0.05$, $d = 0.5$, e condições iniciais $x_0 = -20$, $y_0 = 0.0$, $z_0 = 0.0$ e $w_0 = 15.0$. Para esses parâmetros, $\tau = 20$, SCALMN = 0.00, SCALMX = 2.90, EVOLV = 4.

Na Tabela 6.2, tem-se os valores λ_1 calculados pelo método clássico, utilizando-se a equação variacional e ortonormalização de *Gram-Schmidt* (Ramasubramanian e Sriram, 2000; Nogueira, 2001; Sprott, 2003), e os valores ao final das iterações consideradas para os expoentes obtidos pelo método do “máximo expoente” implementado a partir do algoritmo proposto por Wolf *et al.* (1985).

Máximo Expoente de <i>Lyapunov</i> (λ_1)		
Sistemas Implementados	Método Padrão	Método do Máximo Expoente
Função Senoidal	0.000000	0.008399
Mapa de <i>Henón</i>	0.419220	0.460223
Oscilador de <i>van der Pol</i>	0.193300	0.198735
Atrator de <i>Lorenz</i>	0.905600	0.897969
Sistema de <i>Rössler</i>	0.112500	0.110942
Sistema (6.5) no FPAA	–	17.82143

Tabela 6.2: Expoentes de *Lyapunov* obtidos a partir das séries experimentais.

Com isso, na Figura 6.9, tem-se a evolução do maior expoente da série adquirida no sistema implementado no FPAA em função do número de iterações consideradas. Nessa situação, o valor do expoente apresentado na Tabela 6.2 foi obtido com os parâmetros SCALMN = 0.60, SCALMX = 3.90, EVOLV = 2 e, em todos os casos apresentados na Figura 6.9, o número de pontos considerado na série temporal foi de 32768, permitindo, assim, uma adequada resolução dos valores obtidos.

Na Figura 6.9 está apresentado o resultado do experimento realizado variando-se o parâmetro EVOLV do algoritmo, objetivando estudar o comportamento da evolução do expoente ao longo das iterações consideradas. Pode-se perceber que conforme esse parâmetro é variado, o algoritmo fornece valores distorcidos, pois ou o sistema não evoluiu o suficiente para gerar informações consideráveis (EVOLV = 1), ou o sistema divergiu de forma a perder informação

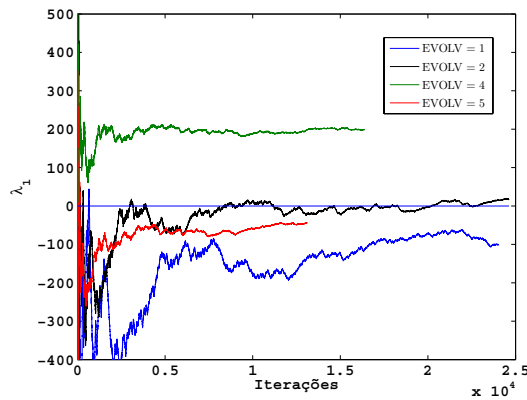


Figura 6.9: Alteração do parâmetro EVOLV.

($EVOLV = 5$) resultando, neste caso, em expoentes de *Lyapunov* negativos, contradizendo o comportamento representado na Figura 6.7.

Em outro experimento, Figura 6.10, variou-se o valor do parâmetro SCALMX entre os valores 1.50, 1.60 (para essa situação, melhor caso) e 1.70 objetivando analisar os efeitos desse parâmetro durante a evolução do cálculo do expoente. Nessa situação, considerando-se, também, $SCALMN = 0.17$, $EVOLV = 2$ e 32768 pontos, $\lambda_1 = 6.848310$. Para valores muito pequenos, interpreta-se que a trajetória secundária não divergiu o suficiente da trajetória fiducial (Wolf *et al.*, 1985), não sendo possível, então, a geração de informação que fosse suficiente e, portanto, proporcionando distorções durante o cálculo. No caso de valores maiores de SCALMX em relação ao que forneceu os resultados corretos, houve divergência das duas trajetórias, e grande o suficiente para perder-se a informação sobre o real comportamento do sistema.

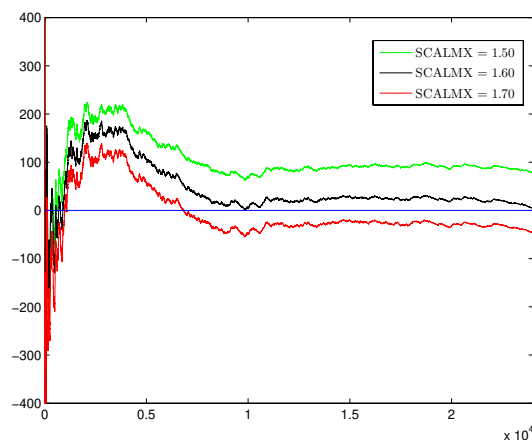


Figura 6.10: Alteração do parâmetro SCALMX.

Capítulo 7

Conclusões e Perspectivas Futuras

7.1 Conclusões

Neste trabalho de dissertação de mestrado objetivou-se realizar estudos em relação à tecnologia baseada em FPAA, em virtude de ser uma tecnologia que vem ganhando interesse, principalmente pelo fato de possibilitar que alterações do modelo implementado possam ser realizadas em tempo de execução, ou seja, *on-the-fly*. Nesse sentido, pode-se concluir que os FPAAs, em especial, os modelos da família AN221E04 da ANADIGM, são bastante promissores e podem ser utilizados em muitas aplicações como o desenvolvimento de filtros. Contudo, por se tratar de uma tecnologia nova, quando comparada à tecnologia baseada em FPGAs, ainda possui algumas limitações mais fortemente relacionadas à quantidade de elementos e ao tamanho do circuito que podem ser implementados no FPAA.

Outro objetivo deste trabalho consistiu em realizar estudos de caso para aplicações com o dispositivo FPAA. Pode-se concluir que esse objetivo foi realizado com êxito no sentido de que foram desenvolvidos estudos específicos sobre a Teoria de Caos e a Teoria de Controle de Sistemas Dinâmicos Não-Lineares. Essas áreas, pelos resultados experimentais obtidos, tendem a beneficiar-se intensamente pela utilização da tecnologia FPAA no sentido de diminuição do tempo necessário para evolução dos modelos dinâmicos em estudo e no sentido de correção de erros relacionados às precisões numéricas inerentes aos computadores pessoais, geralmente utilizados na realização de experimentações em sistemas caóticos.

O leitor deve atentar-se ao fato de que este trabalho não visou o estudo aprofundado de algum modelo dinâmico específico, no sentido de analisar seus comportamentos com

extremo nível de detalhes e quantificações, e sim que foram usados alguns modelos tradicionalmente apresentados na literatura especializada como paradigmas para demonstrar que a técnica proposta pode auxiliar em muito tais tipos de estudos, oferecendo vantagens enormes sobre as técnicas até então consideradas tradicionais. Como por exemplo, a tremenda rapidez relativa na resolução de sistemas de equações diferenciais, a precisão nos cálculos, a fidelidade de representação entre modelo real e modelo simulado e a simplicidade e custo de projeto de simuladores para sistemas complexos.

Através desses excelentes resultados obtidos é possível concluir que a técnica não só seja eficiente para atividades de simulação e análise de sistemas, como também tenha grandes possibilidades de ser aplicada no auxílio aos sistemas de controle aplicados, e no sentido semelhante ao que tange sobre as velocidades e precisões nos cálculos e também à inerente facilidade de embarcação de algoritmos sofisticados em circuitos eletrônicos de baixo custo e fácil reprogramação.

7.2 Perspectivas Futuras

Mediante ao que foi observado durante todo período de trabalho dessa dissertação, alguns experimentos e desenvolvimentos poderão ser realizados a partir das análises apresentadas não somente nesse texto mas também a partir de muitas referências. Em especial, outros trabalhos podem ser desenvolvidos como, por exemplo:

- Implementação, em FPAA, de outros modelos dinâmicos que podem apresentar comportamento caótico e, caso necessário, programação de novas CAMs;
- Desenvolvimento de um *toolbox* que englobe vários sistemas dinâmicos e que possam ser analisados a partir das ferramentas apresentadas no Capítulo 3;
- Migração do ambiente de desenvolvimento MATLAB para outro ambiente, preferencialmente baseado em plataformas de *software* livre tais como o Ubuntu, OpenSuse, Kurumin, etc., permitindo a implementação de sistemas compilados, melhorando o desempenho geral do sistema e das análises;
- Realização do processo de reconfiguração dinâmica utilizando-se microcontroladores e/ou FPGAs;
- Desenvolvimento de uma plataforma híbrida que fosse composta de dispositivos FPAAs e microcontroladores e/ou PLDs trabalhando em conjunto e de maneira automatizada.

Apêndice A

Algoritmo do Máximo Expoente

A.1 Considerações Iniciais

Neste apêndice é apresentado o algoritmo, em sua forma original, apresentado por Wolf *et al.* (1985) e que serviu como base para o algoritmo implementado em linguagem de programação do *software* MATLAB. Esse código foi, então, utilizado para as considerações realizadas sobre as séries temporais obtidas durante os experimentos com o FPAA.

A.2 Metodologia do Máximo Expoente

A seguir, tem-se o código, em FORTRAN, extraído do trabalho desenvolvido por Wolf *et al.* (1985).

```
PROGRAM FET1
INTEGER DIM,TAU,EVOLV
DIMENSION X(16384),PT1(12),PT2(12)
C
C      **DEFINE DELAY COORDINATES WITH A STATEMENT FUNCTION**
C      **Z(I,J)=JTH COMPONENT OF ITH RECONSTRUCTED ATTRACTOR POINT**
C
      Z(I,J)=X(I+(J-1)*TAU)
C
      OPEN (UNIT=1,FILE='INPUT.',TYPE='OLD')
C
      TYPE*, 'NPT,DIM,TAU,DT,SCALMX,SCALMN,EVOLV?'
      ACCEPT*,NPT,DIM,TAU,DT,SCALMX,SCALMN,EVOLV
```

```

C
C      **IND:  POINTS TO FIDUCIAL TRAJECTORY**
C      **IND2: POINTS TO SECONDY TRAJECTORY**
C      **SUM:  HOLDS RUNNING EXPOENT ESTIMATE SANS 1/TIME**
C      **ITS:  IS TOTAL NUMBER OS PROPAGATION STEPS**
C
      IND = 1
      SUM = 0.0
      ITS = 0
C
C      **READ IN TIME SERIES**
C
      DO 10 I = 1,NPT
        READ (1,*) X(I)
10    CONTINUE
C
C      **CALCULATE USEFUL SIZE OF DATAFILE
C
      NPT = NPT - DIM*TAU - EVOLV
C
C      **FIND NEAREST NEIGHBOR TO FIRST DAA POINT**
C
      DI = 1.E38
C
C      **DONT TAKE POINT TOO CLOSE TO FIDUCIAL POINT**
C
      DO 30 I = 11,NPT
C
C      **COMPUTE SEPARATION BETWEEN FIDUCIAL POINT AND CANDIDATE**
C
        D = 0.0
        DO 20 J = 1,DIM
          D = D+(Z(IND,J)-Z(I,J))**2
20    CONTINUE
        D = SQRT(D)
C
C      **STORE THE BEST POINT SO FAR BUT NO CLOSER THAN NOISE SCALE**
C
        IF (D.GT.DI.OR.D.LT.SCALMN) GO TO 30
        DI = D
        IND2 = I
30    CONTINUE
C
C      **GET COORDINATES OF EVOLVED POINTS**
C
40    DO 50 J = 1,DIM
      PT1(J) = Z(IND+EVOLV,J)

```

```

        PT2(J) = Z(IND2+EVOLV,J)
50  CONTINUE
C
C      **COMPUTE FINAL SEPARATION BETWEEN PAIR, UPDATE EXPOENT**
C
        DF = 0.0
        DO 60 J = 1,DIM
            DF = DF+(PT1(J)-PT2(J))**2
60  CONTINUE
        DF = SQRT(DF)
        ITS = ITS+1
        SUM = SUM+ALOG(DF/DI)/(FLOAT(EVOLV)*DT*ALOG(2.))
        ZLYAP = SUM/FLOAT(ITS)
        TYPE*,ZLYAP,EVOLV*ITS,DI,DF
C
C      **LOOK FOR REPLACEMENT POINT**
C      **ZMULT IS MULTIPLIER OF SCALMX WHEN GO TO LONGER DISTANCES**
C
        INDOLD = IND2
        ZMULT = 1.0
        ANGLMX = 0.3
70  THMIN = 3.14
C
C      **SEARCH OVER ALL POINTS**
C
        DO 100 I = 1,NPT
C
C      **DONT TAKE POINTS TOO CLOSE IN TIME TO FIDUCIAL POINTS**
C
            III = IABS(I-(IND+EVOLV))
            IF (III.LT.10) GO TO 100
C
C      **COMPUTE DISTANCE BETWEEN FIDUCIAL POINT AND CANDIDATE**
C
            DNEW = 0.0
            DO 80 J = I,DIM
                DNEW = DNEW+(PT1(J)-Z(I,J))**2
80  CONTINUE
            DNEW = SQRT(DNEW)
C
C      **LOOK FURTHER AWAY THAN NOISE SCALE, CLOSER THAN ZMULT*SCALMX**
C
            IF (DNEW.GT.ZMULT*SCALMX.OR.DNEW.LT.SCALMN) GO TO 100
C
C      **FIND ANGULAR CHANGE OLD TO NEW VECTOR**
C
            DOT = 0.0

```

```

        DO 90 J = 1,DIM
            DOT = DOT+(PT1(J)-Z(I,J))*(PT1(J)-PT2(J))
90      CONTINUE
        CTH = ABS(DOT/(DNEW*DF))
        IF (CTH.GT.1.0) CTH = 1.0
        TH = ACOS(CTH)
C
C      **SAVE POINT WITH SMALLEST ANGULAR CHANCE SO FAR**
C
        IF (TH.GT.THMIN) GO TO 100
        THMIN = TH
        DII = DNEW
        IND2 = I
100    CONTINUE
        IF (TH.GT.THMIN) GO TO 110
C
C      **CANT FIND A REPLACEMENT - LOOK AT LONGER DISTANCES**
C
        ZMULT = ZMULT+1
        IF (ZMULT.LE.5) GO TO 70
C
C      **NO REPLACEMENT AT 5*SCALE, DOUBLE SEARCH ANGLE, RESET DISTANCE**
C
        ZMULT = 1.0
        ANGLMX = 2*ANGLMX
        IF (ANGLMX.LT.3.14) GO TO 70
        IND2 = INDOLD+EVOLV
        DII = DF
110    CONTINUE
        IND = IND+EVOLV
C
C      **LEAVE PROGRAM WHEN FIDUCIAL TRAJECTORY HITS END OF FILE**
C
        IF (IND.GE.NPT) GO TO 120
        DI = DII
        GO TO 40
120    CALL EXIT
        END

```

Referências Bibliográficas

- Abarbanel, H. D. I. (1996), *Analysis of Observed Chaotic Data*, Institute for Nonlinear Science, 1a. edição, Springer-Verlag New York, Inc.
- Albano, A. M., Muench, J., Schwartz, C., Mees, A. I. e Rapp, P. E. (1988), ‘Singular-value decomposition and the Grassberger-Procaccia algorithm’, *Physical Review A* **38**(6), 3017–3026.
- Anadigm (2007), ‘Anadigm - the dpASP Company’, Página principal de documentação: <http://www.anadigm.com/an221e04.asp>.
- Andrade Jr., A., Vieira, G., Balen, T. R., Lubaszewski, M., Azaïs, F. e Renovell, M. (2005), ‘Built-in self-test of global interconnects of field programmable analog arrays’, *Microelectronics Journal* **36**(12), 1112–1123.
- Arena, P., Baglio, S., Fortuna, L. e Manganaro, G. (1995), ‘Hyperchaos from cellular neural networks’, *Electronics Letters* **31**(4), 250–251.
- Arena, P., Baglio, S., Fortuna, L. e Mangarano, G. (1995), ‘Chua’s Circuit Can Be Generated by CNN Cells’, *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications* **42**(2), 123–125.
- Arena, P., Fortuna, L., Frasca, M. e Patané, L. (2005), ‘A CNN-Based Chip for Robot Locomotion Control’, *IEEE Transactions on Circuits and Systems I: Regular Papers* **52**(9), 1862–1871.
- Baccigalupi, A. e Liccardo, A. (2007), ‘Field Programmable Analog Arrays for Conditioning Ultrasonic Sensors’, *IEEE Sensors Journal* **7**(8), 1176–1182.
- Baker, G. L. e Gollub, J. P. (1996), *Chaotic dynamics: an introduction*, 2a. edição, Cambridge University Press.

- Banbrook, M. (1996), Nonlinear Analysis of Speech from a Synthesis Perspective, Tese de doutorado, The University of Edinburgh.
- Broomhead, D. S. e King, G. P. (1986), ‘Extractiing Qualitative Dynamics from Experimental Data’, *Physica D: Nonlinear Phenomena* **20**(2-3), 217–236.
- Brown, R., Bryant, P. e Abarbanel, H. D. I. (1991), ‘Computing the Lyapunov spectrum of a dynamical system from observed time-series’, *Physical Review A* **43**(6), 2787–2806.
- Brown, S. e Rose, J. (1996), ‘FPGA and CPLD Architectures: A Tutorial’, *IEEE Design and Test of Computers* **13**(2), 42–57.
- Cai, D., Lai, Y.-C. e Winslow, R. L. (1993), ‘Complex Dynamics in Coupled Cardiac Pacemaker Cells’, *Physical Review Letters* **71**(15), 2501 – 2504.
- Cai, D., Winslow, R. L. e Noble, D. (1994), ‘Effects of Gap Junction Conductance on Dynamics of Sinoatrial Node Cells: Two-Cells and Large-Scale Network Models’, *IEEE Transactions on Biomedical Engineering* **41**(3), 217–231.
- Callegari, S., Merendino, G., Golfarelli, A., Zagnoni, M. e Tartagni, M. (2006), ‘Applicability of field programmable analog arrays to capacitive sensing in the Sub-pF range’, *Analog Integrated Circuits and Signal Processing* **47**(1), 39–51.
- Caponetto, R., Mauro, A. D., Fortuna, L. e Frasca, M. (2005), ‘Field Programmable Analog Array to Implement a Programmable Chua’s Circuit’, *International Journal of Bifurcation and Chaos* **15**(5), 1829–1836.
- Carloni, L. P., Bernardinis, F. D., Sangiovanni-Vicentelli, A. L. e Sgroi, M. (2002), The Art and Science of Integrated Systems Design, *em* ‘Proceeding of the 32nd Solid-State Device Research Conference 2002’, pp. 19–30.
- Carroll, T. L. e Pecora, L. M. (1991), ‘Synchronizing Chaotic Circuits’, *IEEE Transactions on Circuits and Systems* **38**(4), 453–456.
- Chen, G. e Ueta, T. (1999), ‘Yet Another Chaotic Attractor’, *International Journal of Bifurcation and Chaos* **9**(7), 1465–1466.
- Chua, L. O. (1993), ‘Global Unfolding of Chua’s Circuit’, *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences* **E76-A**(5), 704–734.

- Chua, L. O., Komuro, M. e Matsumoto, T. (1986), ‘The Double Scroll Family’, *IEEE Transactions on Circuits and Systems* **CAS-33**(11), 1073–1118.
- Corti, S., Molteni, F. e Palmer, T. N. (1999), ‘Signature of recent climate change in frequencies of natural atmospheric circulation regimes’, *Nature* **398**(6730), 799–802.
- Darbyshire, A. G. e Broomhead, D. S. (1996), ‘Robust estimation of tangent maps and Liapunov spectra’, *Physica D: Nonlinear Phenomena* **89**(3-4), 287–305.
- de Souza, M. A. T. (2000), Otimização de controladores nebulosos de Takagi-Sugeno utilizando algoritmos genéticos, Dissertação de Mestrado, FEEC - UNICAMP.
- Delai, A. L. (2008), Aplicação de Controlador Evolutivo a Pêndulo Servo Acionado, Dissertação de Mestrado, FEEC - UNICAMP.
- Farmer, J. D., Ott, E. e Yorke, J. A. (1983), ‘The Dimension of Chaotic Attractors’, *Physica D: Nonlinear Phenomena* **7**(1-3), 153–180.
- Femat, R., Campos-Delgado, D. U. e Martínez-López, F. J. (2005), ‘A family of driving forces to suppress chaos in jerk equations: Laplace domain design’, *Chaos* **15**(4), 043102(1)–043102(9).
- Fiedler-Ferrara, N. e do Prado, C. P. C. (1994), *Caos: Uma Introdução*, Editora Edgard Blücher Ltda.
- Fraser, A. M. (1989), ‘Information and Entropy in Strange Attractors’, *IEEE Transactions on Information Theory* **35**(2), 245–262.
- Fraser, A. M. e Swinney, H. L. (1986), ‘Independent coordinates for strange attractors from mutual information’, *Physical Review A* **33**(2), 1134–1140.
- Guardia Filho, L. E. (2005), Sistema para Controle de Máquinas Robotizadas utilizando Dispositivos Lógicos Programáveis, Dissertação de Mestrado, FEEC - UNICAMP.
- Guardia Filho, L. E. (2007), Sistema Híbrido com Dispositivos Reconfiguráveis Digitais e Analógicos para Estudo e Simulação de Caos, Tese de doutorado, FEEC - UNICAMP. Em preparação.
- Guess, D. e Sailor, W. (1993), ‘Chaos Theory and the Study of Human Behavior: Implications for Special Education and Development Disabilities’, *The Journal of Special Education* **27**(1), 16–34.

- Haykin, S. S. (2000), *Redes Neurais - Princípios e Prática*, 2a. edição.
- Holt, T. A. (2002), ‘A chaotic model for tight diabetes control’, *Diabetic Medicine* **19**(4), 274–278.
- Hoshi, N., Utsumi, Y. e Oguchi, K. (2004), A Web-accessible FPGA-based direct torque controller for permanent magnet synchronous motor, *em* ‘The 4th International Power Electronics and Motion Control Conference, 2004. IPEMC 2004.’, Vol. 3, IEEE Conference Proceeding, pp. 1325–1330.
- Huang, C.-K., Tsay, S.-C. e Wu, Y.-R. (2005), ‘Implementation of chaotic secure communication systems based on OPA circuits’, *Chaos, Solitons and Fractals* **23**(2), 589–600.
- Islama, M. M. e Muraseb, K. (2005), ‘Chaotic dynamics of a behavior-based miniature mobile robot: effects of environment and control structure’, *Neural Networks* **18**(2), 123–144.
- Jungbeck, M. (2001), Implementação de Controladores Neurais de Kim-Lewis-Dawson com Parâmetros Otimizados por Algoritmos Genéticos, Dissertação de Mestrado, FEEC - UNICAMP.
- Kaygisiz, B. H., Erkmen, I. e Erkmen, A. M. (2006), ‘Intelligent analysis of chaos roughness in regularity of walk for a two legged robot’, *Chaos, Solitons and Fractals* **29**(1), 148–161.
- Kiers, K., Klein, T., Kolb, J., Price, S. e Sprott, J. C. (2004), ‘Chaos in a Nonlinear Analog Computer’, *International Journal of Bifurcation and Chaos* **14**(8), 2867–2873.
- Kinsner, W. (2006), ‘Characterizing Chaos Through Lyapunov Metrics’, *IEEE Transactions on Systems, Man and Cybernetics - Part C: Applications and Reviews* **36**(2), 141–151.
- Kolodko, J. e Vlacic, L. (2003), Experimental system for real-time motion estimation, *em* ‘2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, 2003. AIM 2003. Proceedings.’, Vol. 2, IEEE Conference Proceeding., pp. 981–986.
- Kroll, M. H. (1999), ‘Biological variation of glucose and insulin includes a deterministic chaotic component’, *Biosystems* **50**(3), 189–201.
- Laknaur, A., Durbha, R. e Wang, H. (2006), ‘Built-In-Self-Testing Techniques for Programmable Capacitor Arrays’, *Journal of Electronic Testing: Theory and Applications* **22**(4-6), 449–462.

- Liu, X., Wang, J. e Huang, L. (2007), ‘Attractors os Fourth-Order Chau’s Circuit and Chaos Control’, *International Journal of Bifurcation and Chaos* **17**(8), 2705–2722.
- Lorenz, E. N. (1963), ‘Deterministic Nonperiodic Flow’, *Journal of the Atmospheric Sciences* **20**(2), 130–141.
- Lorenz, E. N. (2005), ‘A look at some details of the growth of initial uncertainties’, *Tellus Series A* **57**(1), 1–11.
- Lü, J., Chen, G., Cheng, D. e Celikovsky, S. (2002), ‘Bridge the Gap between the Lorenz System and the Chen System’, *International Journal of Bifurcation and Chaos* **12**(12), 2917–2926.
- Madrid, M. K. (1988), Robô-Manipulador Mecânico TRRR para Posicionamento Espacial com Controle Digital Hierárquico à Micro-processadores., Dissertação de Mestrado, FEEC - UNICAMP.
- Madrid, M. K. (1994), Controle de Trajetórias Contínuas por Seccionamento em Sub-Trejetórias Usando Inteligência Artificial num Robô Multi-Tarefas, Tese de doutorado, FEEC - UNICAMP.
- Matsumoto, T., Chua, L. O. e Komuro, M. (1985), ‘The Double Scroll’, *IEEE Transactions on Circuits and Systems* **CAS-32**(8), 798–818.
- Matsumoto, T., Chua, L. O. e Tokumasu, K. (1986), ‘Double Scroll Via a Two-Transistor Circuit’, *IEEE Transactions on Circuits and Systems* **CAS-33**(8), 828–835.
- Merendino, G., Callegari, S., Golfarelli, A., Zagnoni, M. e Tartagni, M. (2005), ‘Signal conditioning for capacitive sensors with field programmable analog arrays’, *International Symposium on Signals, Circuits and Systems, 2005. ISSCS 2005.* **1**, 139–141.
- Nichols, J. M. e Nichols, J. D. (2001), ‘Attractor reconstruction for non-linear systems: a methodological note’, *Mathematical Biosciences* **171**(1), 21–32.
- Nicolato, F. (2002), Arquitetura de Hardware para a Extração em Tempo Real de Características de Múltiplos Objetos em Imagens de Vídeo: Classificação de Cores e Localização de Centróides, Dissertação de Mestrado, FEEC - UNICAMP.
- Nicolato, F. (2007), Estudo e Implementação de um Método de Cinemática Inversa Baseado em Busca Heurística para Robôs Manipuladores: Aplicação em Robôs Redundantes e Controle Servo Visual, Tese de doutorado, FEEC - UNICAMP.

- Nogueira, R. G. (2001), Técnicas alternativas de reconhecimento de caos em sistemas com dinâmica complexa, Tese de doutorado, FEEC - UNICAMP.
- Orrel, D. (2002), ‘Role of the metric in forecast error growth: how chaotic is the weather?’, *Tellus Series A* **54**(4), 350–362.
- Parker, T. S. e Chua, L. O. (1989), *Practical Numerical Algorithms for Chaotic Systems*, Springer-Verlag.
- Parlitz, U. (1992), ‘Identification of True and Spurious Lyapunov Exponents from Time Series’, *International Journal of Bifurcation and Chaos* **2**(1), 155–165.
- Principe, J. C. e Lo, P.-C. (1991), Estimation Of Lyapunov Exponents From The EEG, *em* ‘Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 1991. Vol.13’, pp. 2230–2231.
- Quiroga, R. Q. (1998), Quantitativa analysis of EEG Signals: Time-frequency methods and Chaos theory, Tese de doutorado, Institute of Physiology and Institute of Signal Processing - Medical University Lübeck.
- Ramasubramanian, K. e Sriram, M. S. (2000), ‘A comparative study of computation of Lyapunov spectra with different algorithms’, *Physica D: Nonlinear Phenomena* **139**(1-2), 72–86.
- Rangarajan, G., Habibi, S. e Ryne, R. D. (1998), ‘Lyapunov Exponents without Rescaling and Reorthogonalization’, *Physical Review Letters* **80**(17), 3747–3750.
- Rohden, H. (2007), *Einstein: O Enigma do Universo*, Martin Claret. Texto Integral.
- Sano, M. e Sawada, Y. (1985), ‘Measurement of the Lyapunov Spectrum from a Chaotic Time Series’, *Physical Review Letters* **55**(10), 1082–1085.
- Schildt, H. (1997), *C Completo e Total*, 3a. edição, Makron Books.
- Shil’nikov, L. P. (1993), ‘Chua’s Circuit: Rigorous Results and Future Problems’, *IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications* [veja também *em IEEE Transactions on Circuits and Systems I: Regular Papers*] **40**(10), 784–786.
- Spong, M. W. e Vidyasagar, M. (1989), *Robot Dynamics and Control*, 1a. edição, IE-WILEY.
- Sprott, J. C. (2003), *Chaos and Time-Series Analysis*, 1a. edição, Oxford University Press.

- Stoop, R. e Parisi, J. (1991), ‘Calculation of Lyapunov exponents avoiding spurious elements’, *Physica D: Nonlinear Phenomena* **50**(1), 89–94.
- Takens, F. (1981), *Detecting strange attractors in turbulence*, Vol. 898 de *Lecture Notes in Mathematics*, Springer Berlin / Heidelberg, pp. 366–381.
- Tang, W. e Yip, L. (2004), Hardware implementation of genetic algorithms using FPGA, *em* ‘The 47th Midwest Symposium on Circuits and Systems, 2004. MWSCAS ’04.’, Vol. 1, IEEE Conference Proceeding, pp. 549–552.
- Terry, M. A., Marcus, J., Farrell, M., Aggarwal, V. e O’Reilly, U.-M. (2006), GRACE: Generative Robust Analog Circuit Exploration., *em* F. Rothlauf, J. Branke, S. Cagnoni, E. Costa, C. Cotta, R. Drechsler, E. Lutton, P. Machado, J. H. Moore, J. Romero, G. D. Smith, G. Squillero e H. Takagi, editores, ‘EvoWorkshops’, Vol. 3907 de *Lecture Notes in Computer Science*, Springer, pp. 332–343.
- Vianello, R. L. e Alves, A. R. (1991), *Meteorologia Básica e Aplicações*, 1a. edição, Livraria Editora UFV.
- Winslow, R. L., Cai, D., Varghese, A. e Lai, Y.-C. (1995), ‘Generation and Propagation of Normal and Abnormal Pacemaker Activity in Network Models of Cardiac Sinus Node and Atrium’, *Chaos, Solitons and Fractals* **5**(3 - 4), 491–512.
- Wolf, A., Swift, J. B., Swinney, H. L. e Vastano, J. A. (1985), ‘Determining Lyapunov Exponents from a Time Series’, *Physica D: Nonlinear Phenomena* **16**(3), 285–317.
- Znamirowski, L., Palusinski, O. A. e Reiser, C. (2002), ‘Optimization Technique for Dynamic Reconfiguration of Programmable Analog/Digital Arrays’, *Analog Integrated Circuits and Signal Processing* **31**(1), 19–30.
- Znamirowski, L., Palusinski, O. A. e Vrudhula, S. B. K. (2004), ‘Programmable Analog/Digital Arrays in control and simulation’, *Analog Integrated Circuits and Signal Processing* **39**(1), 55–73.