

IDENTIFICADOR NEURAL DE COMANDOS VOCAIS PARA O ACIONAMENTO SEM FIOS DE UM ROBÔ LEGO MINDSTORMS

FÁBIO F. RODRIGUES¹, AGNALDO J. R. REIS¹, MÔNICA DO AMARAL²

1 Departamento de Engenharia de Controle e Automação e Técnicas Fundamentais, Escola de Minas, Universidade Federal de Ouro Preto, Campus Morro do Cruzeiro, Ouro Preto-MG, 35.400-000

2 Universidade Federal de São Carlos, Programa de Pós-Graduação em Engenharia de Produção, Rod. Washington Luís, km 235, São Carlos-SP, 13. 565-905

E-mails: *fabiofonsek@gmail.com, agnreis@gmail.com, monica@dep.ufscar.br*

Abstract— The process of automatic speech recognition by a robot is very complex because there may be, e.g., changes in the way that words are pronounced. In this paper, the wireless process of driving a mobile robot, which stems from a Lego kit Mindstorms®, is described. The robot should receive vocal commands such as forward, back, right, left and stop, and execute them. The input commands are captured by a microphone, identified by means of a neural network and transmitted via infrared signals to the robot. The tests presented here reflect real situations of robot's operation.

Keywords— Speech Recognition, Artificial Neural Networks, Database, Lego Mindstorms, Robotics.

Resumo— O processo de reconhecimento automático da fala por um robô é de grande complexidade uma vez que podem existir, e.g., variações no modo da fala do locutor. Neste artigo, descreve-se o processo de acionamento sem fios de um robô móvel, construído utilizando-se o kit Lego Mindstorms®. O robô deverá receber comandos vocais tais como: frente, ré, direita, esquerda e pare, e executá-los. Os comandos de entrada serão capturados por um microfone, identificados através de uma rede neural e transmitidos ao robô via sinais de infravermelho. Os testes realizados refletem situações reais de operação do robô.

Palavras-chave— Reconhecimento da Fala, Redes Neurais Artificiais, Banco de Dados, Lego Mindstorms, Robótica.

1 Introdução

O reconhecimento automático da fala humana tem sido alvo de muitos estudos, por possibilitar automatismos a uma diversidade de processos. A evolução dos recursos de *hardware* e *software* tem feito com que essa tecnologia seja cada vez mais difundida, o que permite uma vasta gama de aplicações (BRAGA, 2006; VALLIATI, 2000).

Esse automatismo pode trazer um grande número de benefícios, tais como celeridade na realização de tarefas, mobilidade do usuário enquanto realiza ações, aplicações médicas para manipulação de equipamentos e o acesso de deficientes físicos e visuais a novos contextos, dentre vários outros (BRAGA, 2006).

O reconhecimento da fala por meio de Redes Neurais Artificiais (RNA) teve início na década de 1980. A capacidade de processamento paralelo e distribuído das RNA fez com que elas obtivessem uma boa eficiência (VALIATTI, 2000; De PAULA, 2000). Também se destacam na área técnicas que utiliza métodos estatísticos, como os Modelos Ocultos de Markov (YNOGUTI, 1999).

No presente trabalho, utilizam-se RNA para a tarefa de reconhecimento de comandos vocais. Para tanto, segmentos de áudio são capturados via microfone, digitalizados e processados, e servem de base para a produção dos padrões vocais de interesse (e.g., frente, ré, direita etc.), padrões esses que serão apresentados ao identificador neural. Uma vez identificado o comando vocal, ele é enviado ao robô via sinais de infravermelho, que processa essa informação e a executa.

Têm-se por objetivos, desenvolver: 1) uma rede neural capaz de identificar alguns comandos vocais; utilizando-se como base o trabalho de Nascimento Jr. & Yoneyama (2000); 2) um banco de dados que possa ser acessado pelo robô remotamente; 3) uma estratégia para combinar RNA com robótica, dando mais visibilidade às primeiras e estimulando o estudo de ambas. A integração desses três pontos é a principal contribuição deste artigo.

A seguir, a seção 2 trata da implementação do sistema de reconhecimento de voz. Já a seção 3 apresenta detalhes construtivos do robô, enquanto a seção 4 se refere à integração entre *software* e *hardware*. Na seção 5 são apresentados os testes e resultados obtidos, ao passo que a seção 6 apresenta as principais conclusões e sugestões de trabalhos futuros. Por fim, apresenta-se uma lista com as referências bibliográficas consultadas.

2 Sistema Neural de Reconhecimento de Voz

O sistema neural de reconhecimento de voz (SNRV) desenvolvido pode ser dividido em 3 partes, a saber: aquisição do sinal sonoro, processamento desse sinal e sua posterior identificação. Na seqüência, cada uma dessas etapas será detalhada. A Figura 1 mostra um diagrama de blocos de um SNRV.

2.1 A Aquisição do sinal sonoro

Para se fazer a aquisição do sinal de som, ou seja, da fala, e a digitalização do sinal, utilizou-se um computador com sistema operacional Windows

XP® e um microfone ligado à placa de som. Os parâmetros tempo de amostragem, frequência de amostragem e quantização receberam os valores de 2 s, 8.000 Hz e 8 bits, respectivamente. Esses valores foram obtidos após uma série de testes, sendo distintos daqueles utilizados por Nascimento Jr. & Yoneyama (2000).

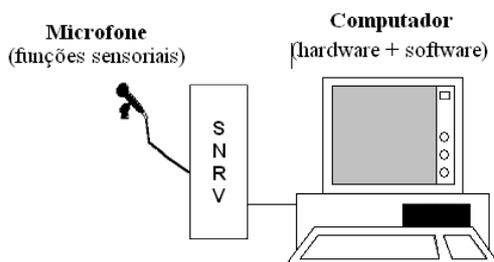


Figura 1: Diagrama de blocos de um SNRV.

2.2 O processamento do sinal de som

Na literatura, encontram-se duas técnicas mais usuais para se realizar a análise espectral. Uma utiliza o *Linear Predictive Coding* (LPC), cuja análise resulta em parâmetros referentes a um banco de filtros. A outra, utilizada neste trabalho, é a Transformada Discreta de Fourier (TDF), em que se transfere um sinal no domínio do tempo para o domínio da frequência (BRESOLIN, 2003).

O primeiro passo que deve ser executado para se realizar o processamento do sinal é aplicar-se a ele a TFD, obtendo-se assim uma representação discreta do sinal no domínio da frequência (Figura 2.a). Em seguida, corta-se a parte simétrica do espectro gerado, i.e., desconsideram-se os pontos no intervalo de 4.001 a 8.000 (Figura 2.b). Daí, o espectro resultante é segmentado em 31 bandas com largura de 100 Hz cada, iniciando-se a primeira banda em 50 Hz (e.g., Figura 2.c). Para se usar esses valores como padrão de entrada da rede neural, faz-se uma normalização dos mesmos no intervalo de -1 a 1, para que eles fiquem dentro dos limites teóricos da função de ativação tangente hiperbólica (Figura 2.d). O padrão de energia em cada uma dessas bandas será apresentado a uma rede neural com 5 saídas, cada uma representando um dos 5 comandos (Figura 3).

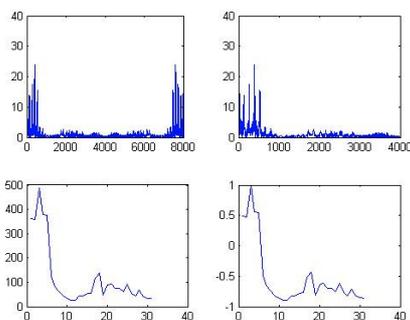


Figura 2: Aplicação da TDF no comando 'frente'. (a) 'Frente' no domínio da frequência. (b) Obtenção de sua parte simétrica. (c) Banda típica de 100 Hz (como obtida). (d) Banda típica de 100 Hz (normalizada entre -1 e 1).

2.3 A identificação do comando vocal

2.3.1 A arquitetura da rede neural

A rede escolhida para este trabalho é um Perceptron de Múltiplas Camadas (PMC) com 31 entradas, que são os padrões gerados a partir do comando vocal (e.g., Figura 2.d), 9 neurônios na camada oculta e 5 neurônios na camada de saída. A resposta dada pela rede se encontra na forma de um vetor coluna com cinco posições, onde 4 posições são preenchidas com o valor 0 e a posição restante é preenchida com o valor 1, indicando o comando a ser executado. A título de exemplo, a saída 'frente' produz o seguinte vetor de saída: $[1\ 0\ 0\ 0\ 0]^T$.

A Figura 3 mostra o PMC utilizado. Sua configuração foi obtida em Amaral (2005).

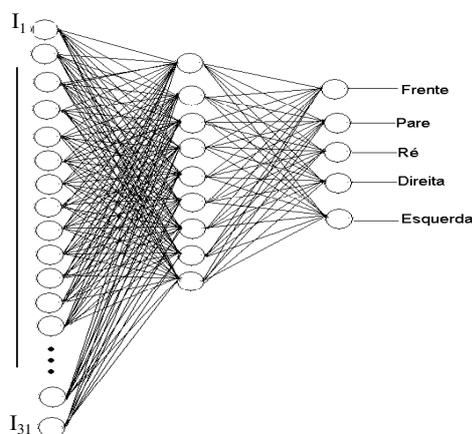


Figura 3: Arquitetura do identificador neural.

2.3.2 O Treinamento da rede neural

Uma vez definidos o tipo de rede e sua estrutura, partiu-se para a etapa de treinamento da mesma. Para tanto, estabeleceram-se 15 padrões de entrada para cada comando vocal, totalizando 75 padrões. Desse total, 50% dos padrões foram utilizados no conjunto de treinamento e os outros 50% compuseram o conjunto de testes. Como critério de parada, definiu-se que o número máximo de épocas seria 10.000 ou que o Erro Quadrático Médio (EQM) fosse inferior a 0,001, definidos com base em Amaral (2005).

A Tabela 1 mostra uma síntese dos principais parâmetros discutidos nas seções 2.3.1 e 2.3.2. Já a Figura 4 mostra os resultados do treinamento do identificador neural para reconhecer o comando 'frente'. Nesse caso, constata-se que foram necessárias 235 épocas para que o EQM fosse inferior a 0,001.

3 O Projeto do Robô

Como citado anteriormente, o robô foi desenvolvido a partir do *kit* Lego Mindstorms® (LEGO EDUCATION, 2009). O *kit* contém um conjunto de peças encaixáveis que permitem o

desenvolvimento de várias aplicações. O cérebro dele é um microcomputador chamado RCX, que foi inicialmente desenvolvido em colaboração com *Massachusetts Institute of Technology Media Lab*. O kit ainda contém vigas, pranchas, engrenagens, polias, rodas, cabos, conectores, motores, sensores e a torre de transmissão de sinais de infravermelho, entre outros. Foram utilizados dois motores conectados às portas A e C do RCX, além de outras peças. A Figura 5 mostra uma foto do protótipo, que deverá ser capaz de 'ouvir' os comandos e executá-los, e da torre de transmissão de dados.

Tabela 1: Parâmetros da RNA e seus valores.

Parâmetro da RNA	Valor
Número de neurônios na camada de entrada	31
Número de neurônios na camada de saída	5
Número de neurônios na camada oculta	9
Taxa de aprendizado	0,1
Número máximo de épocas	10.000
Erro quadrático médio máximo permitido	0,001
Função de ativação	Tang. hiperbólica
Número de padrões de entrada semelhantes	8
Largura das bandas do sinal de som	100 Hz

4 A Integração entre o Software e o Hardware

4.1 A Conexão do Matlab com o banco de dados

Toda vez que uma nova saída da rede é produzida, ela deve ser armazenada em um banco de dados. Isso se deve ao fato de o Matlab® não se comunicar diretamente com o RCX.

Neste trabalho, utiliza-se o MySQL da Sun Microsystems® (MYSQL, 2009), por ser um banco robusto, gratuito e de fácil integração com o Matlab. Graças a essa integração, é possível se inserir comandos responsáveis por executar *queries* que armazenam, modificam e/ou excluem entradas em um banco de dados a partir do Matlab.

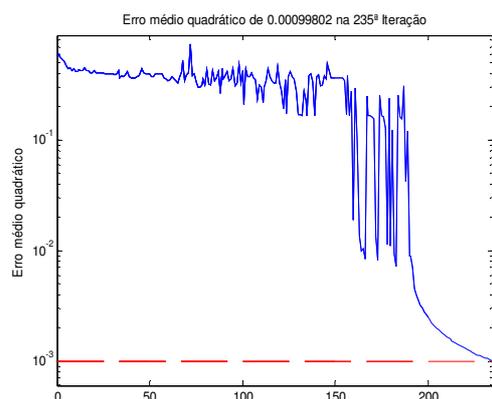


Figura 4: Treinamento do identificador neural.

Em (MATHWORKS, 2009) é possível se obter as rotinas 'mysql_input.m' e 'mysql_output.m', que permitem a integração do Matlab com o MySQL através de uma página do tipo *HyperText Preprocessor* (PHP). Para que isso ocorra, o conteúdo de saída do arquivo mysql_output.m é

enviado para o arquivo PHP¹ que irá realizar a sua inserção no banco de dados do MySQL.

Neste trabalho, o único parâmetro que será realmente necessário para o robô é o comando vocal identificado (i.e., a saída da rede). Outros parâmetros, tais como 'hostname', 'username' e 'password' são deixados no próprio arquivo PHP, que manipula o banco de dados e deve ficar no diretório raiz do servidor. Além disso, para que as operações realizadas entre o banco de dados e o Matlab funcionem corretamente, o SQL, o PHP, e o servidor APACHE devem estar instalados e em execução no computador.

Maiores informações sobre o trabalho podem ser encontradas em Rodrigues (2009). Em relação à instalação completa dos pacotes supracitados, ver Wampserver, (2009) e para a integração entre o Matlab e o MySQL, ver Almgren (2009) ou Lawrence (2006).



Figura 5: Protótipo do robô e sua torre infravermelho.

4.2 O monitoramento do banco de dados e o envio de informações

A parte responsável por fazer o monitoramento do banco de dados e a comunicação com o RCX é uma página do tipo *Java Server Pages* (JSP). Essa página tem quase a mesma função de uma página PHP, só que pode ser programada em Java e oferece suporte de comunicação com o RCX utilizando da torre de transmissão de infravermelho (FERRARI, 2002; NETO, 2004; PERENHA, 2009).

O componente que realiza a conexão e captura os dados do banco é um 'bean'² que deverá ser importado para essa página JSP. A partir de uma requisição feita por essa página, o bean apenas retorna um valor, ou um conjunto de valores, que no caso deste trabalho é o resultado da consulta ao banco de dados recuperando-se o último valor inserido nele (i.e., a saída da rede mais recente).

¹ Para se rodar o PHP é necessário a presença de um servidor de web, e.g., o Apache (APACHE, 2009).

² Um 'bean' se assemelha a um 'script file'.

De posse da saída da rede, transforma-se a 'string' correspondente ao comando vocal em um número inteiro, facilitando o seu envio ao RCX.

A comunicação entre a página JSP e o RCX se dá através da torre de transmissão e é feita a partir das classes presentes em RCX.comm (SCHOLZ *et al.*, 2002). Essas classes são resultantes da compilação de arquivos Java presentes no conjunto de rotinas do Lego Java Operating System (LeJOS). A propósito, essa linguagem foi escolhida após a realização de um estudo comparativo entre as linguagens LeJOS, Not Quite C (NQC) (KNUDSEN, 2009) e Robolab³ (LEGO EDUCATION, 2009). Optou-se por utilizar o LeJOS por permitir uma maior flexibilidade de programação ao desenvolvedor, ampliando sobremaneira a gama de aplicações possíveis.

Para visualizar páginas JSP é preciso instalar e configurar um servidor *web* dedicado a este tipo de página. Neste trabalho utilizou-se o servidor Tomcat 6.0 (D'AVILA, 2003). Para deixar o robô apto a receber comandos, deve-se inicializar esse servidor e localizar através do navegador a página construída para esse fim. Finalizando, a estrutura de dados com os arquivos necessários para o processo de comunicação entre o JSP e o RCX é mostrada na Figura 6.

4.3 A programação do robô

Utilizou-se o *software* RCXTools para realizar a comunicação com o RCX (SCHOLZ *et al.*, 2002). Por meio dele, pode-se transferir o 'firmware' necessário para executar programas em Java para o robô, permitindo a compilação e/ou transferência de programas entre o banco de dados e o robô. Um programa típico recebe o valor enviado pela página JSP e aciona os motores de acordo com o comando vocal reconhecido.

5 Testes e Resultados

O procedimento de treinamento e testes do identificador neural envolveu 15 amostras de cada palavra. Dessas, 8 amostras foram utilizadas no treinamento da rede e as demais na avaliação de sua capacidade de generalização. O treinamento foi realizado com uma versão modificada do algoritmo *Back-Propagation*⁴. Os testes realizados foram do tipo 'cego', uma vez que os padrões usados nos testes não participaram do estágio de treinamento.

³ Ferramenta de programação oficial do kit Lego Mindstorms.

⁴ Utilizou-se a subrotina MATLAB chamada de TRAINGDY.

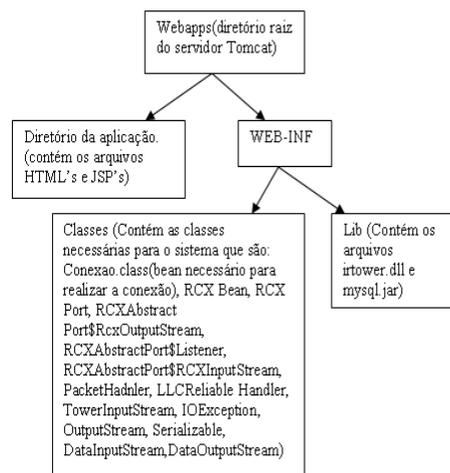


Figura 6: Estrutura de dados e classes fundamentais.

A Tabela 2 mostra os resultados obtidos. A partir dela, nota-se que a taxa de acerto média é de aproximadamente 90% e que o identificador neural apresentou um pouco mais de dificuldade para classificar corretamente o comando 'direita', obtendo-se uma taxa de acerto de 80%.

Tabela 2. Taxas de acerto para o conjunto de testes.

Comando	Taxa de acerto (%)
Frente	90
Ré	90
Pare	90
Direita	80
Esquerda	90

Com relação à comunicação entre o Matlab e o robô, não foram encontrados quaisquer problemas durante os testes realizados. Porém, deve-se observar que: 1) a torre de transmissão é alimentada com uma bateria de 9 v e a grande demanda de energia durante o processo de transmissão pode ocasionar seu desgaste prematuro. Portanto, sugere-se utilizar baterias alcalinas ou adaptar a alimentação para uma fonte CC externa; 2) o alcance da torre é de 2,5 m, definindo uma área de atuação para o robô.

Finalmente, utilizando ainda a mesma rede, averiguou-se a influência do ruído do ambiente (de baixa intensidade) na sua capacidade de generalização. Isto é, a rede foi treinada com padrões sem interferência, mas foi testada com padrões contaminados com algum tipo de ruído.

Em linhas gerais, se o usuário falar bem próximo ao microfone e se o ruído for de baixa magnitude, a rede é capaz de ignorar o ruído e se concentrar no comando. Desse modo, as taxas de acerto médias continuam elevadas, mas são um pouco deterioradas e ficam na faixa de 85 %. Os testes ainda sugerem que a entonação da voz é uma característica mais relevante que o ruído ambiente de baixa intensidade.

6 Conclusão

Neste artigo apresentou-se uma metodologia para o acionamento sem fios de um robô móvel através de comandos vocais reconhecidos por uma rede neural. Além do apelo educacional e do caráter interdisciplinar envolvidos, destaca-se a integração entre o *kit* Lego Mindstorms® e o Matlab como motivação para a realização desse trabalho. Dessa forma, contribui-se para um melhor aproveitamento dos recursos disponíveis nas universidades e motiva-se a realização de projetos robóticos por alunos de graduação.

A solução desenvolvida foi dividida em três etapas. Na primeira, discutiu-se a implementação de um sistema neural para reconhecimento de voz. Vários testes cegos foram realizados, obtendo-se taxas de acerto entre 80 e 90%. Na etapa seguinte, desenvolveu-se um procedimento que permitisse a comunicação entre o identificador neural e o robô móvel. Para tanto, utilizou-se um banco de dados MySQL, páginas PHP e JSP, e um servidor de *web* Tomcat 6.0. Durante todos os testes realizados, não foram observadas falhas de comunicação. Os cuidados a serem tomados dizem respeito à carga da bateria da torre de transmissão e ao seu alcance máximo de 2,5 m. Finalmente, a etapa 3 diz respeito à construção do robô móvel. O protótipo foi desenvolvido a partir do kit Lego Mindstorms.

Uma extensão natural deste trabalho seria o acionamento sem fios de um robô via internet. Nesse caso, o robô pode ser operado remotamente desde que no local onde ele se encontra exista uma câmera e um computador com acesso a *web*. Outra melhoria desejável, senão mandatária, seria a investigação de procedimentos que aumentassem a taxa de acertos do identificador neural. Por fim, a criação de uma biblioteca de funções poderia facilitar a realização de melhorias e novos projetos.

Referências Bibliográficas

- ALMGREN, Robert. MySQL and Matlab. <http://mmf.utoronto.ca/resrchres/mysql>. Acessado em 15/05/2009.
- AMARAL, Mônica do. Identificador Neural de Comandos de Voz Embutido em um Robô Lego Mindstorms. Trabalho de Conclusão de Curso em Engenharia de Controle e Automação. Ouro Preto, 2005.
- APACHE FOUNDATION. The Apache Software Foundation. <http://www.apache.org>. Acessado em 05/07/2009.
- BRAGA, Petrônio de L. Reconhecimento de voz dependente do locutor utilizando Redes Neurais Artificiais. Trabalho de Conclusão de Curso em Eng. da Computação. Recife, 2006.
- BRESOLIN, Adriano. Estudo do Reconhecimento de Voz para o Acionamento de Equipamentos Elétricos via Comandos em Português. Dissertação de Mestrado. Joinville, 2003.
- D'AVILA, Marcos. Tutorial Tomcat - Instalação e configuração básica, 2003. <http://www.mhavila.com.br/topicos/java/tomcat.html>. Acessado em 10/05/2009.
- DE PAULA, Maurício. Reconhecimento de palavras faladas utilizando Redes Neurais Artificiais. Trabalho de Conclusão de Curso. Pelotas, 2000.
- FERRARI, Giulio. Programming Lego mindstorms with java. Rockland: Syngress Publishing, 2002.
- KNUDSEN, Jonathan. The Unofficial Guide to LEGO MINDSTORMS Robot. <http://oreilly.com/catalog/lmstorms/chapter/ch04.html>. Acessado em 05/07/2009.
- LAWRENCE, David. How to get matlab to talk to mysql, 2006. <http://desk.stinkpot.org:8080/tricks/index.php/2006/02/how-to-get-matlab-to-talk-to-mysql>. Acessado em 15/05/2009.
- LEGO EDUCATION. The ROBOLAB Software. <http://www.lego.com/eng/education/mindstorms/home.asp?pagename=robolab>. Acessado em 05/07/2009.
- MATHWORKS, 2009. <http://www.mathworks.com/matlabcentral/fileexchange/15002>. Acessado em 05/07/2009.
- MYSQL. <http://www.mysql.com>. Acessado em 05/07/2009.
- NASCIMENTO JR., Cairo & YONEYAMA, Takashi. Inteligência Artificial em Controle e Automação. São Paulo: Edgar Blücher, 2000. Págs. 185 a 187.
- NETO, Oziel. Entendendo e Dominando o Java. São Paulo: Digerati, 2004.
- PERENHA, Rodrigo. Aplicação em JSP e Javabeans utilizando banco de dados, 2009. <http://www.javafree.org/artigo/10956/Aplicacao-em-JSP-e-Javabeans-utilizando-Banco-de-Dados.html>. Acessado em 20/05/2009.
- RODRIGUES, Fábio F. Acionamento de um Robô Lego Mindstorms por Comandos Vocais Utilizando Redes Neurais Artificiais. Trabalho de Conclusão de Curso em Engenharia de Controle e Automação. Ouro Preto, 2009.
- SCHOLZ, M.P.; BAGNALL, Brian; GRIFFITHS, Lawrie. The Lejos Tutorial, 2002. <http://lejos.sourceforge.net/tutorial/index.html>. Acesso em 26/05/2009.
- VALLIATI, João F. Reconhecimento de Voz para Comandos de Direcionamento por meio de Redes Neurais. Dissertação de Mestrado. Univ. Federal do Rio Grande do Sul, 2000.
- WAMPSEVER. <http://www.wampserver.com/en>. Acessado em 05/07/2009.
- YNOGUTI, Carlos A. Reconhecimento de Fala Continua Usando Modelos Ocultos de Markov. Tese de Doutorado. Campinas, 1999.