

ao atraso médio de transmissão, isto em função da carga na rede, do tamanho dos pacotes, do tamanho dos *buffers* e do tipo de tráfego. Apresentamos, ainda, uma estimativa do número máximo de canais de voz, sem degradação significativa da inteligibilidade do sinal reconstruído, para diferentes tipos de tráfego. O problema de sincronização em sistemas de voz com comutação de pacotes é também aqui abordado. Finalmente, propomos um algoritmo de reconstrução, sem perdas, do sinal para um sistema em que não há detecção de silêncio.

A rede *Ethernet* e o seu mecanismo de acesso não determinístico, o CSMA/CD, revelaram-se adequados para a integração de voz e dados e outras aplicações em tempo real. Concluímos, ainda, que o estabelecimento de uma conversação é possível, mesmo para cargas relativamente elevadas na ordem dos 60 a 70%, com uma perda de pacotes não superior a 2% e um atraso inferior a 200 ms. Testes informais mostraram que perdas desta ordem para pacotes de 32 ms são aceitáveis pelo utilizador.

Abstract

Local computer networks, today widely spread, have emerged from the need of exchanging data between terminals, computers and all sorts of network servers. However, the spoken word still remains the communication mode of choice for mankind - in the form of two-way conversations, one-way broadcasts and others. The use of local networks for the transmission of voice/data traffic has been subject of many studies. Many of these networks exhibit good throughput but poor delay characteristics; variable or excessive delays may render intelligibility problems to users of such a voice communication system.

This work presents an experimental study on two personal computer based implementations of a 64 kbit/s voice communication system over the Ethernet local network - the former working at the logic link layer and the latter over TCP/IP protocol. System performance is evaluated according to its average transmission delay and percentage of packet loss as a function of network load, packet size, buffers size, and traffic characteristics. An estimate of the maximum number of voice channels that can be supported without significant degradation of the reconstructed signal is also presented. A reference is made to synchronisation problems in packet voice systems and a lossless reconstruction algorithm for a system with no silence detection is proposed.

Ethernet and its non deterministic access method, CSMA/CD, have proven to be adequate for voice/data integration and other real-time applications. We finally conclude that it is possible to establish a conversation, even at relatively high loads of up to 60 or 70%, with less than 2% of lost packets and delays smaller than 200 ms. Informal tests showed that packet loss within this range for 32 ms packets is acceptable to the user.

Título: Topologias Analógicas para Filtros Gama

Title: Analog Topologies for Gamma-Filters

Autor/Author: Vitor Grade Tavares

Orientador/Advisor: Pedro Guedes de Oliveira

Data Apresentação/Acceptance Date: 03/94

Palavras Chave: CMOS-VLSI, condensadores comutados, mosfet-c, resistências activas cmos, filtros adaptativos, filtros gama

Key Words: CMOS-VLSI, switch capacitors, mosfet-c, cmos active resistors, adaptive filters, gamma filters

Mestrado/M.S.

Resumo

Os filtros que pertencem à classe de *generalized feedforward filters* (GNFF), podem-se considerar como modelos intermédios entre os filtros FIR e IIR em termos de complexidade, estabilidade e capacidade. Nesta dissertação são versadas, em particular, as questões relacionadas com a implementação analógica de um destes GNFF, denominado de filtro gama. Este é um filtro IIR sujeito a restrições bem determinadas, as quais permitem que o controlo de estabilidade se torne trivial. Prova-se que qualquer sinal pode ser decomposto na base formada pelos filtros gama, com um erro arbitrariamente pequeno em L2. Existem métodos de gradiente para a aprendizagem dos diversos parâmetros do filtro.

Propostas em tempo contínuo e discreto são analisadas. Em tempo discreto é apresentado um novo método de implementação de filtros com resistências activas em CMOS que permite a programação contínua dos parâmetros do filtro. São ainda analisados diversos esquemas de circuitos SC (*Switch Capacitor*).

Em tempo contínuo apresenta-se uma solução com multiplicadores de capacidade para aplicações de muito baixa frequência. Estes esquemas utilizam métodos de filtragem do tipo MOSFET-C (linearização da característica quadrática dos MOSFETs).

Para as topologias em tempo discreto são apresentados ainda modelos simples de CFT (Clock Feedthrough) que permite determinar, em primeira ordem, quais os componentes críticos e até que ponto estes efeitos não lineares influenciam o desempenho do filtro. São também analisados os efeitos das correntes de fugas nas topologias de tempo discreto. Apresentam-se modelos que permitem muito facilmente prever quais os *off-sets* e ruído na saída devido a estas correntes.

Foram finalmente desenhados em CMOS-VLSI algumas das topologias propostas para validar as simulações.

Abstract

Filters that belong to the class of "Generalized Feedforward Filters" (GNFF), can be considered as models in-between FIR and IIR according to its complexity, stability and capacity. This dissertation is concerned in particular with issues related with analogue implementation of one of these GNFF, called the gamma filter. This is an IIR filter with constraints that make the

stability control trivial. It can be proved that any signal can be decomposed over the gamma filter basis with an arbitrary small error in L2. Gradient methods to adapt the different parameters are available.

Discrete and continuous time implementations are analyzed. In discrete time is presented a new method to design filters. It utilizes active resistors in CMOS that allows a continuous programming of the filters parameters. Several SC circuits are also analyzed.

In continuous time is presented a capacitor multiplier solution, for very low frequency applications, with MOSFET-C filtering techniques.

For the discrete time topologies, simple models of CFT (Clock Feedthrough) and stray currents are presented. The models allow, in a first order approximation, to determine which are the critical components, and until which point these effects influence the filter performance. These models predict, in a relative simple form, the noise and offset of the different topologies.

Most of the topologies were implemented in a CMOS-VLSI technology to validate the simulations.

Título: Virtualização de Autômatos Programáveis

Title: Virtualization of Programmable Logic Controllers

Autor/Author: Jorge Augusto Fernandes Ferreira

Orientador/Advisor: João Pedro Estima de Oliveira

Data Apresentação/Acceptance Date: 07/94

Palavras Chave: Autômatos programáveis, linguagem de contactos, diagramas de escada, PLC virtual, virtualização, programação orientada por objectos, programação em ambiente Windows, interfaces gráficas, simulação orientada por objectos, simulação em tempo real

Key Words: Programmable logic controllers, ladder diagrams, virtual PLC, virtualization, object oriented programming, programming in Windows, graphic interfaces, object oriented simulation, real time simulation

Mestrado/M.S.

Resumo

Os autômatos programáveis (vulgarmente denominados PLCs, do inglês *Programmable Logic Controllers*) são bastante usados na indústria. Isto implica que um maior número de pessoas deve obter treino suficiente para desenvolver, manter e melhorar aplicações envolvendo PLCs, desde o simples comando de um motor até sofisticados controladores de processos. Neste trabalho foi desenvolvido um sistema (VPC) para o ensino da programação de PLCs, sem a necessidade de laboratórios de custo elevado, e para testar programas antes de usá-los em condições reais.

A ideia básica é a virtualização de um PLC. Isto significa que o PLC é construído por *software*, por exemplo num computador pessoal, de tal modo que as entradas são introduzidas pelo teclado ou pelo rato, e as saídas são visualizadas no monitor.

O sistema VPC contém três blocos principais: o editor de programação, o PLC virtual e o simulador. O editor suporta a conhecida linguagem de contactos (*Ladder*) e fornece as ferramentas necessárias para desenvolver programas para PLCs, utilizando uma interface *point-and-click* baseada em menus e ícones. O PLC virtual e o simulador implementam as operações comuns de um PLC real, tais como contactos abertos e fechados, saídas, temporizadores, contadores e registos de deslocamento. Ligado ao simulador existe um editor de estímulos para definir as entradas do PLC virtual.

O PLC virtual pode executar programas em três modos diferentes: a simulação rápida, onde os sinais aparecem sob a forma de diagramas temporais no editor de estímulos, o qual fornece também informação temporal; a simulação em tempo real, onde as entradas e as saídas são visualizadas sobre uma imagem do PLC real, cujo comportamento está a ser emulado, através dos usuais díodos emissores de luz; e a simulação interactiva, também uma simulação em tempo real, onde as entradas são definidas durante o processo de simulação e os resultados são visualizáveis numa janela específica.

Presentemente, o bloco PLC virtual, do sistema descrito, suporta um autómato específico, mas o uso de técnicas de programação por objectos permite uma fácil modificação, de modo a poder emular outros PLCs comerciais. O uso destas técnicas permite também a fácil adição de instruções *Ladder* mais avançadas. O sistema VPC corre como uma aplicação para o ambiente *Microsoft Windows*.

Abstract

Programmable Logic Controllers (PLC) are widely used in industry. This implies that more and more people must be trained in order to obtain enough knowledge to develop, maintain and upgrade applications involving PLCs, from simple motor commands to sophisticated process controllers. This project presents a software system (VPC) to teach PLC programming, without the need of costly practical class rooms, and to test programs before using them in real conditions.

The basic idea is the virtualization of the PLC. This means that the PLC is software constructed, for instance inside a personal computer, in such a way that the inputs are obtained with the keyboard or the mouse and the outputs are visualized on the screen.

The VPC system has three main blocks: the Editor, the Virtual PLC, and the Simulator. The Editor, with menus, icons and a point-and-click interface, supports the well known relay ladder programming language and supplies the tools needed to develop a PLC program. The Virtual PLC and the Simulator implement the common operations of a real PLC, such as open and closed contacts, outputs, timers, counters and shift registers. Linked to the Simulator there is a stimuli editor to input the conditions into the Virtual PLC.

The Virtual PLC can simulate programs in three different modes: the fast simulation, where the signals appear as a diagram of logic states inside the stimuli