

Vinicius André Sorgato

**RECONHECIMENTO DE MENSAGENS
EM UMA REDE PLC BANDA ESTREITA**

Florianópolis
Julho de 2011

Vinicius André Sorgato

**RECONHECIMENTO DE MENSAGENS
EM UMA REDE PLC BANDA ESTREITA**

Trabalho de Conclusão de Curso
apresentado como parte dos requisitos
para obtenção do grau de Bacharel em
Ciências da Computação.

Orientador:

Prof. Dr. Antônio Augusto Medeiros Fröhlich

Co-Orientador:

Roberto de Matos

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CURSO DE BACHARELADO EM CIÊNCIAS DA COMPUTAÇÃO

Florianópolis
Julho de 2011

Título: Reconhecimento de mensagens em uma rede PLC banda estreita

Autor: Vinicius André Sorgato

Banca Examinadora:

Prof. Dr. Antônio Augusto Fröhlich

M.Eng. Roberto de Matos

M.Sc. Arliones Stevert Hoeller Junior

M.Sc. Hugo Marcondes

RESUMO

A utilização da rede de energia elétrica para comunicação de informação não é uma idéia nova. O primeiro sistema de comunicação a utilizá-las começou a operar no ano de 1922. Com os avanços tecnológicos novos interesses surgiram para a utilização das redes de energia elétrica para comunicação de dados, como para acesso a internet e em aplicações de automação residencial. Em aplicações banda estreita, onde pequenas quantidades de informação são trocadas, a utilização de modems PLC para comunicação pode ser um excesso. Neste trabalho é realizado um estudo sobre a utilização de algoritmos de processamento digital de sinais em conjunto com microcontroladores de baixo desempenho para o reconhecimento de mensagens transmitidas via rede de energia elétrica. Os três algoritmos estudados são o algoritmo de Correlação, algoritmo de Goertzel e a Transformada Rápida de Fourier. Simulações são realizadas para verificar o comportamento de cada algoritmo no reconhecimento de sinais injetados a rede de energia elétrica. Um protótipo é desenvolvido utilizando o algoritmo de Correlação. Nos testes realizados o protótipo se mostrou eficaz no reconhecimento das mensagens, com uma taxa de erros baixa e um tempo de execução aceitável para aplicações banda estreita.

SUMÁRIO

1. Introdução.....	8
1.1 Motivação	10
1.2 Objetivos	11
2. Fundamentação Teórica	12
2.1 Power Line Communications	12
2.2 Conversão analógica-digital	13
2.3 Processamento digital de sinais	14
2.3.1 Correlação	14
2.3.2 Transformada Rápida de Fourier	16
2.3.3 Goertzel	17
3. Desenvolvimento.....	19
3.1 Implementação e validação dos algoritmos	19
3.1.1 Correlação	20
3.1.2 Goertzel	21
3.1.3 Transformada Rápida de Fourier	22
3.2 Testes dos algoritmos em um ambiente simulado	23
3.2.1 Teste de variação de potência	24
3.2.1.1 Correlação	24
3.2.1.2 Goertzel	26
3.2.1.3 Transformada Rápida de Fourier	27
3.2.2 Teste de variação do sinal de entrada	29
3.2.2.1 Correlação	29
3.2.2.2 Goertzel	31
3.2.2.3 Transformada Rápida de Fourier	32
3.2.3 Análise comparativa.....	33
3.3 Implementação do protótipo	34
3.3.1 Definição do algoritmo	34
3.3.1.1 Amostras por ciclo	35

3.3.1.2	Quantidade de amostras.....	37
3.3.1.3	Fator de corte.....	39
3.3.2	Mensagens	40
3.3.3	Hardware utilizado	40
3.3.4	Ambiente de teste	41
3.3.5	Testes e resultados.....	42
4.	Considerações finais	43
4.1	Trabalhos futuros.....	44
	Apêndice A – Receptor – Código-fonte C	47
	Apêndice B – Transmissor – Código-fonte C	50
	Anexo A – Algoritmo de Correlação.....	53
	Anexo B – Algoritmo de Goertzel.....	53
	Anexo C – Algoritmo da Transformada Rápida de Fourier.....	54

LISTA DE FIGURAS

<i>Figura 1 – Exemplo de Correlação</i>	15
<i>Figura 2 - Saída da Transformada Rápida de Fourier. Fonte: Smith, 2003.</i>	17
<i>Figura 3 - Saída do Algoritmo de Goertzel</i>	18
<i>Figura 4 - Sinal composto pelas frequências 30 Hz, 1.1 kHz, 2.3 kHz e 3.2 kHz</i>	20
<i>Figura 5 - Valores de correlação entre o sinal de entrada e as frequências de 30 Hz, 450 Hz, 1.1 kHz, 1.8 kHz, 2.3 kHz, 2.7 kHz e 3.2 kHz.</i>	21
<i>Figura 6 – Magnitude de Goertzel para o sinal da Figura 4 e as frequências de 30 Hz, 450 Hz, 1.1 kHz, 1.8 kHz, 2.3 kHz, 2.7 kHz e 3.2 kHz.</i>	22
<i>Figura 7 – Resultado da FFT para o sinal da Figura 4.</i>	23
<i>Figura 8 – Resultado da correlação para 1 kHz, 2 kHz, 3 kHz, 4 kHz variando a potência do sinal injetado a fundamental da rede elétrica.</i>	25
<i>Figura 9 - Resultado do algoritmo de Goertzel para 1 kHz, 2 kHz, 3 kHz, 4 kHz variando a potencia do sinal injetado a fundamental da rede elétrica.</i>	27
<i>Figura 10 – Espectro de frequências da FFT variando a potencia do sinal injetado a fundamental da rede elétrica.</i>	28
<i>Figura 11 – Resultado do algoritmo de Correlação variando-se a quantidade de amostras.</i>	30
<i>Figura 12 – Resultado para o algoritmo de Goertzel variando-se a quantidade de amostras.</i>	31
<i>Figura 13 – Resultado da FFT variando-se a quantidade de amostras do sinal de entrada utilizado.</i>	32
<i>Figura 14 – Influência das amostras no valor da correlação, taxa de amostragem igual a 8 kHz.</i>	36
<i>Figura 15 - Influência das amostras no valor da correlação, taxa de amostragem igual a 16 kHz.</i>	37
<i>Figura 16 – Quantidade de amostras X Conjunto de frequências</i>	38
<i>Figura 17 – Fator de corte para o algoritmo de Correlação.</i>	39
<i>Figura 18 – Diagrama esquemático do ambiente de teste</i>	41

1. INTRODUÇÃO

A utilização da rede de energia elétrica para transmissão de informação (Power Line Communications) não é uma idéia nova. O primeiro sistema de comunicação a utilizá-las começou a operar no ano de 1922, fazendo uso das linhas de alta tensão numa frequência entre 15 a 500 kHz. O propósito desta comunicação era apenas para controle das subestações de distribuição de energia em caso de falhas, necessitando a transmissão de pouca informação para tal operação.

Com os avanços tecnológicos e a expansão da internet, novos interesses surgiram para a utilização das redes de energia elétrica para comunicação de dados. Como na utilização da vasta infra-estrutura existente para acesso à internet, chegando a localidades onde o custo de instalação de um novo meio de comunicação se tornaria inviável, ou para aplicações de controle e monitoramento, como no desenvolvimento de aplicações de automação residencial, onde utilizando a fiação elétrica já existente é possível fazer a comunicação entre diferentes dispositivos conectados à rede de energia.

A comunicação é realizada através de um dispositivo chamado modem PLC. Este é responsável por fazer a modulação e demodulação das mensagens que trafegam na rede elétrica. Para aplicações banda larga, onde existe a necessidade de comunicação de grandes quantidades de informação, os modems PLC podem ser utilizados para transformar uma rede de energia elétrica residencial em uma LAN (Local Area Network), enquanto que para aplicações banda estreita, onde uma pequena quantidade de informação é trocada, os modems PLC podem ser utilizados para controle, possibilitando executar funções do dispositivo integrado.

As redes PLC banda estreita são utilizadas em aplicações na qual são transmitidas pequenas quantidades de informações. A utilização de um modem PLC neste tipo de aplicação pode ser um excesso. Além disso, os grandes avanços no desenvolvimento de circuitos integrados, trazem novas oportunidades para utilização de sistemas de baixo desempenho no processamento digital de sinais. Assim, a proposta deste trabalho é analisar o

uso de microcontroladores em conjunto com técnicas de processamento digital de sinais, para o reconhecimento de mensagens transmitidas por uma rede PLC banda estreita. O ambiente hostil, as limitações dos microcontroladores e a complexidade dos algoritmos de DSP, são os fatores adversos estudados para o desenvolvimento deste trabalho.

As redes elétricas não foram feitas para realizar transmissão de dados. Este é um ambiente hostil, onde dispositivos conectados a ela, como motores elétricos, podem gerar ruídos que interferem na comunicação. Por isso, um sistema de detecção de mensagens deve ser robusto o suficiente para reconhecer as mensagens transmitidas, sem confundi-las com os ruídos da rede elétrica. O microcontrolador deve ter recursos de hardware suficientes para atender os requisitos do sistema de detecção, que incluem a captação do sinal e a execução do algoritmo de DSP.

Um ambiente de testes foi montado no software Matlab utilizando um sinal capturado diretamente de uma rede elétrica. Simulações foram feitas injetando-se frequências a este sinal para verificar o comportamento dos algoritmos na identificação das mesmas. As frequências eram injetadas com diferentes potências, para verificar a amplitude mínima necessária que um sinal deveria ter para que os algoritmos continuassem a identificá-lo em meio aos ruídos da rede elétrica. Analisou-se também como a variação do tamanho do sinal de entrada influencia a detecção das frequências que compõe o sinal para cada algoritmo.

Realizou-se uma análise comparativa dos três algoritmos considerando os resultados obtidos nos testes e a complexidade existente em implementá-los. Escolheu-se um dos algoritmos para implementação de um protótipo. Um transmissor e um receptor foram desenvolvidos para testar o sistema de comunicação. O transmissor é responsável por gerar as mensagens e transmiti-las ao receptor, que deve capturar o sinal enviado e realizar a identificação das mensagens. Um ambiente de testes foi montado e medições foram realizadas para verificar a eficácia do sistema de detecção.

1.1 Motivação

A construção de sistemas embarcados consiste no desenvolvimento do *hardware* e do *software* que juntos desempenham uma função específica correspondente ao propósito de um sistema. Diversos fatores devem ser levados em consideração no desenvolvimento de um sistema embarcado, como o custo de produção de cada unidade, o custo de desenvolvimento, performance, consumo de energia, flexibilidade, tamanho físico, entre outros.

Os recursos de hardware do sistema embarcado devem ser restritos a somente o que é necessário para desempenhar a sua função. Nenhum componente de hardware que não seja utilizado deve estar presente no sistema embarcado. Também, para reduzir o consumo de energia, a frequência do clock deve ser suficiente somente para que o sistema atenda seus requisitos de desempenho.

No projeto desenvolvido na disciplina de Integração Software/Hardware de 2009-2 foi proposta uma aplicação de automação residencial. Entre as diversas funcionalidades implementadas no projeto, uma delas era as “tomadas inteligentes”. Estas tinham a capacidade de ser controlada de forma remota, via um controlador central (SmartHome).

A comunicação entre as tomadas e o controlador central foi feita através da rede de energia elétrica com a utilização de um modem PLC. Inicialmente as tomadas poderiam receber mensagens de controle, ligar ou desligar. Entretanto o projeto pode ser estendido para reconhecer valores de intensidade, para a tomada atuar como um dimmer. Assim, o modem PLC das tomadas era de certa forma subutilizado, pois sua utilização era somente para o reconhecimento de uma pequena variedade de mensagens.

Além disso, o custo de produção de cada tomada estava diretamente ligado ao custo do modem PLC utilizado, sendo que este componente eleva o custo de produção da tomada. Em conseqüência, o custo do sistema aumenta na medida em que acrescenta-se mais tomadas. Assim, percebe-se que substituindo um componente do sistema que é utilizado diversas vezes (modem PLC, por exemplo), por outro que consiga desempenhar a mesma funcionalidade, porem, com um custo menor, é possível diminuir o custo de produção do sistema, sem afetar as suas características.

Com base nestas informações surgiu a idéia de utilizar-se um microcontrolador de baixo poder de processamento em conjunto com técnicas de Processamento Digital de Sinais (DSP – Digital Signal Processing) para o reconhecimento de mensagens transmitidas via rede de energia elétrica.

1.2 Objetivos

Este trabalho tem como principal objetivo identificar um algoritmo de processamento digital de sinais para reconhecer mensagens transmitidas em uma rede de energia elétrica, o qual seja eficaz e possa ser implementado em sistemas com pouco poder de processamento.

Com base no objetivo principal, os seguintes objetivos específicos são definidos:

- Identificar dentre os algoritmos implementados qual apresenta a melhor relação entre complexidade e eficácia na identificação das mensagens.
- Implementar o algoritmo selecionado em um ambiente embarcado.
- Testar e analisar o comportamento do sistema implementado em um ambiente real.

2. FUNDAMENTAÇÃO TEÓRICA

Nesta seção, apresenta-se a revisão bibliográfica sobre os estudos realizados sobre o tema. Serão apresentados conceitos referentes à tecnologia de comunicação de dados através da rede de energia elétrica (*Power Line Communications*), correlação de sinais, transformada rápida de Fourier e o algoritmo de reconhecimento de tons de Goertzel.

2.1 Power Line Communications

A tecnologia *Power Line Communications* (PLC) utiliza a rede elétrica como meio de transmissão de informações de conteúdo multimídia (*dados, voz, vídeo, áudio*) ou para a transmissão de dados de gerenciamento, automação e controle de dispositivos conectados à rede elétrica.

A comunicação de dados via rede de energia elétrica não é uma idéia nova, muitas empresas fornecedoras de energia elétrica já utilizavam esta tecnologia há algumas décadas para aplicações de telemetria e controle. Com a explosão da internet, os avanços em “*Very-Large-Scale Integrated Circuits*” e nas tecnologias de processamento digital de sinais, o interesse em se utilizar redes elétricas para comunicação de dados foi renovado (Yousuf e El-Shafei, 2007).

A tecnologia PLC está dividida em dois segmentos, “*to the home*” ou “*last mile access*” e “*in the home*” ou “*last inch access*”. O “*last mile access*” diz respeito a como a internet é trazida até as residências. Para isso o PLC é uma das diversas alternativas que incluem também o *cable modem* e diferentes tipos de xDSL, e difusão sem fio, como o WiMAX. Já o “*last inch access*” diz respeito a interconexão de equipamentos dentro de uma residência. Esta tecnologia pode ser utilizada na automação residencial para controle de luzes, sistemas de alarme ou qualquer outro tipo de aplicação que utilize a rede de energia para comunicação, sem a necessidade de instalação de um novo meio de comunicação (MAJUMDER, 2004).

A rede elétrica é transformada em uma rede de comunicação através da superposição de um sinal de baixa potência modulado na senóide da rede

elétrica. Para se certificar de que existe a coexistência e a separação dos dois sistemas, a faixa de frequência usada para comunicação deve ser bem diferente da usada pela rede elétrica. Enquanto a rede elétrica tem uma frequência de 50 a 60 Hz, os dados devem ter uma frequência de 1,705 MHz a 30 MHz para aplicações de PLC banda larga (Resolução 527 Anatel).

2.2 Conversão analógica-digital

Fenômenos do mundo físico são analógicos por natureza. Um fenômeno analógico é um tipo de sinal que varia continuamente em função do tempo. Entretanto, sistemas digitais são discretos. Para existir uma interação entre estes dois mundos, é preciso uma maneira de se converter um sinal analógico em um sinal discreto e vice-versa.

O processo de converter-se um sinal analógico em um sinal discreto é chamado de conversão analógica-digital (ADC – Analog-to-Digital Converter). Já o processo de conversão de um sinal digital para um sinal analógico, é chamado de conversão digital-analógica (DAC – Digital-to-Analog Converter).

O processo de discretização de um sinal analógico é chamado de amostragem. A amostragem é feita fazendo-se medições instantâneas de um sinal analógico contínuo em intervalos regulares, com a utilização de sensores e conversores. Sensores são dispositivos que respondem a um estímulo físico (tensão, temperatura, etc.), gerando um impulso correspondente. O conversor é utilizado para transformar este impulso em um valor digital. O intervalo entre as amostras é chamado de taxa de amostragem (WANNER, 2008).

O teorema de Nyquist-Shannon diz que para uma perfeita reconstrução de um sinal contínuo analógico em um sinal discreto digital, é necessária uma taxa de amostragem duas vezes maior que a frequência máxima do sinal analógico. Assim, um sinal composto por conjunto de frequências é completamente determinado usando uma taxa de amostragem de pelo menos duas vezes a frequência máxima do sinal analógico (WANNER, 2008).

Tanto sensores como conversores estão sujeitos a várias fontes de erros, incluindo erros de precisão, quantização e não-linearidade. Portanto,

deve-se ter cuidado na utilização destas interfaces uma vez que elas determinam a confiabilidade do processamento digital das informações.

2.3 Processamento digital de sinais

O processamento digital de sinais (DSP – Digital Signal Processing) consiste nos métodos matemáticos, nos algoritmos e nas técnicas usadas para manipular sinais que foram convertidos da sua forma analógica para um formato digital. Técnicas de DSP são utilizadas com diversos objetivos como na edição digital de imagens, reconhecimento de voz, comunicação, etc.

Um exemplo de utilização de técnicas DSP é na comunicação telefônica. Em ligações de curta distância, onde os ecos gerados pela fala sofrem um atraso de apenas alguns milissegundos, este problema não é perceptível, pois o ouvido humano é acostumado a ouvir ecos com tempos pequenos de atraso. Porém, conforme a distancia aumenta, estes ecos se tornam cada vez mais perceptíveis, tornando a comunicação desagradável. Utilizando algoritmos de DSP, é possível medir o atraso sofrido pelo eco e gerar um novo sinal eliminando o efeito. (Smith, 2003).

Nas próximas seções são apresentados três algoritmos de processamento digital de sinais, algoritmo de Correlação, algoritmo de Goertzel e a Transformada Rápida de Fourier.

2.3.1 Correlação

A correlação é uma operação matemática utilizada em diversas áreas de conhecimento, ela mostra o grau de similaridade entre duas variáveis. No processamento digital de sinais, para o cálculo da correlação, temos como entrada dois sinais discretos no tempo, e como saída, um novo sinal é produzido. Este sinal é chamado de correlação cruzada, e nos mostra o grau de similaridade entre dois sinais de entrada distintos, levando em conta possíveis deslocamentos no tempo. Se um sinal é correlacionado com ele mesmo, o sinal resultante é chamado de auto-correlação.

A correlação é utilizada para detectar um sinal conhecido dentro de outro sinal desconhecido. Como no funcionamento de um radar, por exemplo, é transmitido um formato de onda de radio conhecido em uma determinada direção. Se esta onda atinge algum objeto, uma pequena fração é refletida de volta na direção do transmissor e é captada por um receptor. O sinal recebido é o sinal que foi transmitido deslocado no tempo, mais os ruídos que ocorrerem no ambiente de propagação. Com a correlação conseguimos identificar se o formato de onda transmitido esta presente no sinal recebido. Através da velocidade de propagação do sinal radio e o deslocamento no tempo do sinal alvo no sinal recebido, calcula-se a distância entre o transmissor e o objeto refletor.

A Figura 1 apresenta um exemplo do funcionamento da correlação. Temos como entrada dois sinais, o sinal desconhecido e o sinal alvo (sinal conhecido), e como saída temos um terceiro sinal, que é a correlação cruzada entre os dois sinais de entrada.

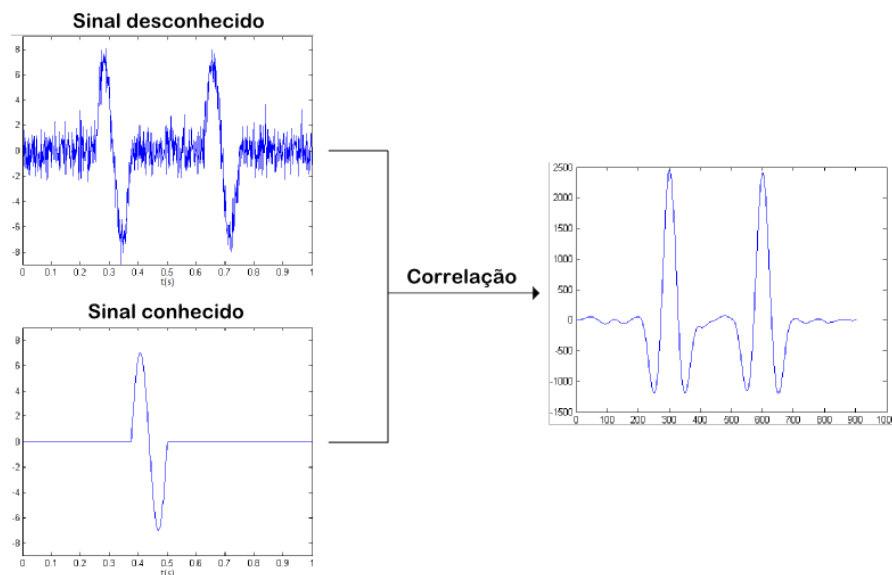


Figura 1 – Exemplo de Correlação

A amplitude de cada amostra do sinal de correlação cruzada é uma medida de quanto o sinal alvo esta relacionado com o sinal desconhecido naquele instante. Isto significa que ira ocorrer um pico no sinal da correlação cruzada toda vez que o sinal alvo estiver presente no sinal desconhecido. Como ilustrado na Figura 1, os dois picos no sinal de saída representam as duas ocorrências do sinal alvo no sinal desconhecido. O valor da correlação

cruzada é maximizado quando o sinal alvo está alinhado na parte do sinal desconhecido que lembra as suas características.

2.3.2 Transformada Rápida de Fourier

A Transformada Rápida de Fourier (FFT – Fast Fourier Transform) é um algoritmo eficiente usado para converter um sinal no domínio do tempo em um sinal equivalente no domínio da frequência, baseado na Transformada Discreta de Fourier (DFT - Discret Fourier Transform) (CHASSAING, 2005). A FFT é de grande importância para diversas aplicações, desde processamento digital de sinais e resolução de equações diferenciais até algoritmos para multiplicação de números inteiros grandes.

A FFT se tornou bastante popular após a publicação de James Cooley e John W. Tukey, do artigo “An algorithm for the machine calculation of complex Fourier Series”, em 1965. Entretanto, outros haviam descoberto a técnica muitos anos antes, como o matemático Carl Friedrich Gauss, que em 1805 utilizou um algoritmo similar ao proposto por Cooley e Tukey nos seus estudos de interpolação das orbitas de corpos celestes.

Atualmente a FFT é uma das ferramentas mais importantes no processamento digital de sinais. Ela pode ser usada para calcular o espectro de frequências de um sinal. Com o espectro de frequências é possível determinar a frequência, fase e amplitude das ondas que compõem o sinal.

A Figura 2 apresenta um exemplo do espectro de frequências fornecido pela FFT. Os dados utilizados como entrada da FFT foram captados com a utilização de um microfone submarino. Entre 10 e 70 hertz, o sinal consiste de uma região relativamente plana, que é chamada de ruído branco. Acima de 70 hertz, o rápido decréscimo do ruído branco é causado por um filtro para as frequências que estão acima de 70 hertz. O pico em 60 hertz é resultado de interferências eletromagnéticas da energia elétrica comercial. O pico existente em 13 hertz é chamado de frequência fundamental, e pode ter sido gerado pela rotação da hélice de um barco. O pico em 26 é a sua segunda harmônica e o pico em 39 hertz é a terceira harmônica (Smith, 2003).

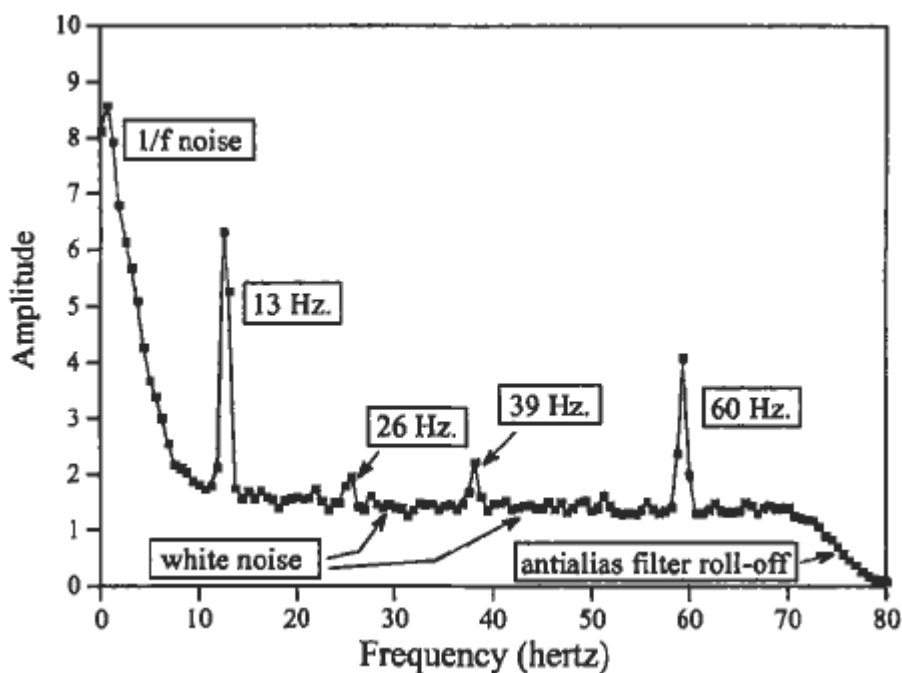


Figura 2 - Saída da Transformada Rápida de Fourier. Fonte: Smith, 2003.

Além da análise do espectro de frequências, a FFT pode ser utilizada para encontrar a resposta de um sistema a um impulso. Ou ainda, como um passo intermediário em uma análise de sinais mais elaborada.

2.3.3 Goertzel

O algoritmo de Goertzel é uma técnica de processamento digital de sinais usado para identificar as frequências que compõem um sinal. Ele foi publicado por Gerald Goertzel em 1958 no artigo "An Algorithm for the Evaluation of Finite Trigonometric Series" (Goertzel, 1958).

O algoritmo de Goertzel é bastante utilizado para detecção de tons telefônicos DTMF (Dual-Tone Multi-Frequency), onde o sistema de detecção deve monitorar constantemente a linha para determinar as duas frequências presentes no sinal, indicando qual botão foi pressionado. Outro exemplo de utilização do algoritmo de Goertzel é na demodulação FSK (Frequency-Shift Keying) onde, tipicamente, duas frequências são utilizadas para transmitir dados binários.

A Figura 3 ilustra a saída do algoritmo de Goertzel, para um sinal ruidoso composto por três frequências, 10 kHz, 1.24kHz e 1.26kHz, onde se desejava detectar a presença das frequências de 1.24kHz e 1.26kHz. Os picos existentes em 1.24 kHz e 1.26 kHz demonstram a detecção das duas frequências no sinal de entrada.

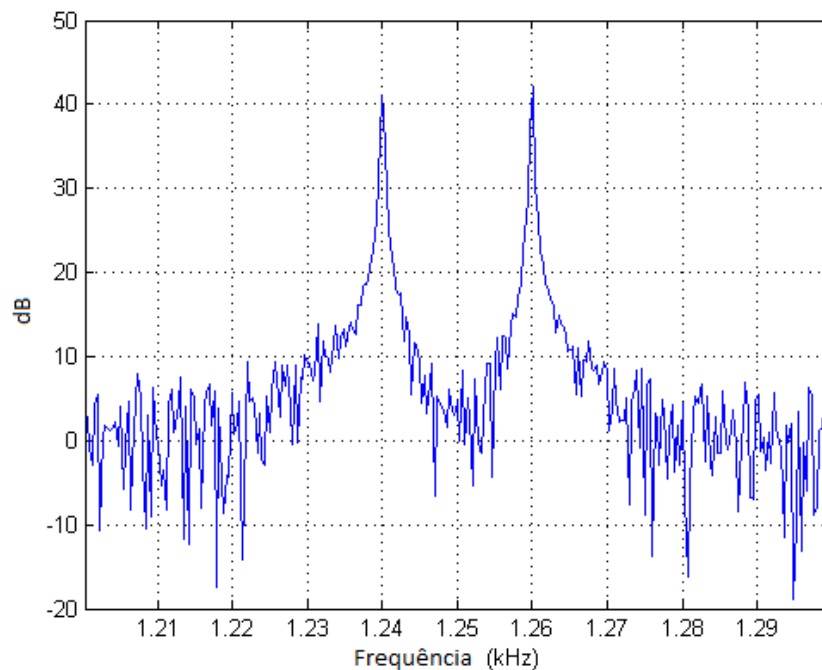


Figura 3 - Saída do Algoritmo de Goertzel

Este algoritmo é útil em situações onde uma pequena quantidade do espectro de frequências é necessária, ao invés de todo ele. Como os dados são processados conforme eles são amostrados, ele permite que o seu processamento já inicie quando a primeira amostra é coletada, diferente da transformada rápida de Fourier, que precisa de um bloco completo para começar suas operações.

3. DESENVOLVIMENTO

Nesta seção será apresentada a metodologia do trabalho, com os procedimentos de testes realizados para validação dos algoritmos e implementação do protótipo.

O primeiro passo foi a implementação e verificação da corretude dos algoritmos no software Matlab. Os três algoritmos foram estimulados com os mesmos sinais e seus resultados foram analisados. Os sinais utilizados nestes testes eram gerados sinteticamente no Matlab, o que garantiu uma previsibilidade do comportamento de cada um.

No segundo passo, testes foram realizados para verificar o comportamento dos algoritmos em um ambiente que simulava as características da rede elétrica. Utilizou-se para isto um sinal capturado diretamente da rede elétrica com a ajuda de um osciloscópio digital. Injetava-se na fundamental da rede uma frequência e estimulavam-se os algoritmos com este sinal. O sinal era injetado com diferentes amplitudes, para verificar a potência necessária para que os algoritmos continuassem a detectar corretamente as frequências que compunham o sinal, evitando interferências dos ruídos da rede elétrica. Uma análise comparativa é feita ajustando-se a quantidade de amostras para cada algoritmo individualmente.

A terceira etapa foi o desenvolvimento do protótipo. Um dos algoritmos foi definido para ser implementado em um sistema embarcado com base nos testes realizados. Algumas peculiaridades deste algoritmo são analisadas e ajustadas para melhorar o sistema de detecção. Apresenta-se o hardware utilizado e o ambiente de testes.

3.1 Implementação e validação dos algoritmos

Para garantir o funcionamento dos algoritmos implementados, utilizou-se o software Matlab para o desenvolvimento de rotinas de teste. Estimulando os algoritmos com um sinal composto por diversas frequências, analisaram-se os seus resultados. Os anexos A, B e C, apresentam os algoritmos de Correlação, Goertzel e Transformada Rápida de Fourier em linguagem Matlab.

O MATLAB é um ambiente de desenvolvimento interativo de alto nível, no qual os problemas e soluções são expressos de maneira similar, como são descritos matematicamente. O MATLAB integra análise numérica, cálculo matricial e processamento de sinais em um ambiente intuitivo e de fácil uso.

A Figura 4 apresenta um exemplo de um sinal de entrada utilizado nos procedimentos de teste. O sinal gerado sinteticamente no Matlab é composto pelas frequências de 30 Hz, 1.1 kHz, 2.3 kHz e 3.2 kHz, com uma taxa de amostragem de 8 kHz. Este sinal foi utilizado como entrada para os algoritmos.

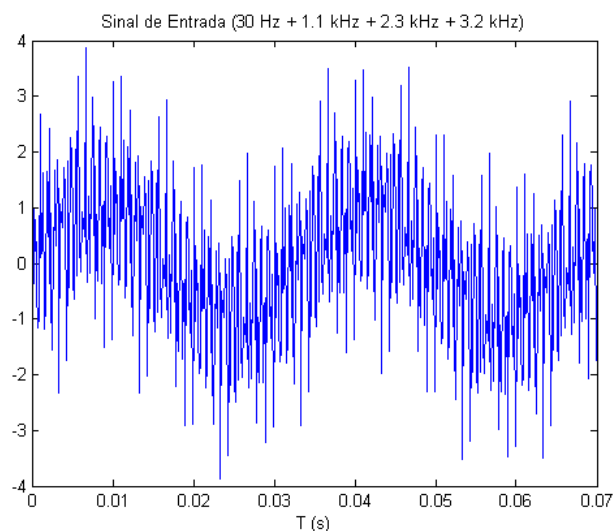


Figura 4 - Sinal composto pelas frequências 30 Hz, 1.1 kHz, 2.3 kHz e 3.2 kHz

3.1.1 Correlação

O algoritmo de Correlação procura por formatos de ondas conhecidos em um sinal desconhecido, por isso, foram definidas outras frequências para serem correlacionadas, além das que estavam presentes no sinal de entrada. Desta maneira é possível demonstrar a diferença entre as correlações das frequências presentes no sinal com as frequências que não estão presentes.

O sinal de entrada mostrado na Figura 4 foi correlacionado com as frequências de 30 Hz, 1.1 kHz, 2.3 kHz, 3.2 kHz, que estão presentes no sinal de entrada, e com as frequências de 450 Hz, 1.8 kHz e 2.7 kHz, que não estão

presentes no sinal de entrada. A Figura 5 apresenta o valor da correlação de cada uma das frequências com o sinal de entrada. As barras mostram a magnitude do valor de correlação obtido para aquela frequência com o sinal de entrada.

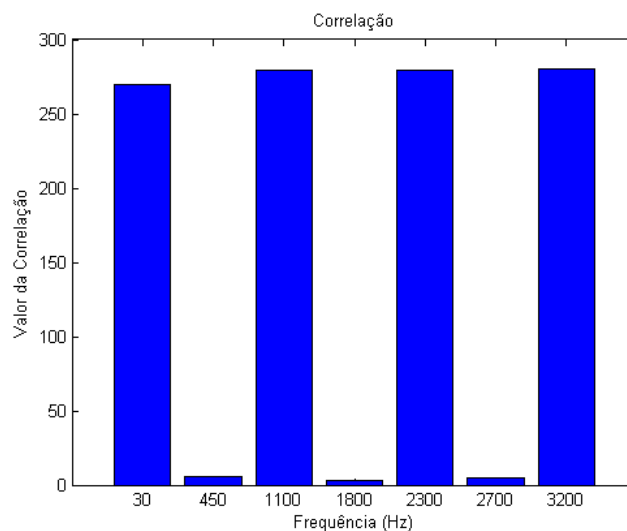


Figura 5 - Valores de correlação entre o sinal de entrada e as frequências de 30 Hz, 450 Hz, 1.1 kHz, 1.8 kHz, 2.3 kHz, 2.7 kHz e 3.2 kHz.

É possível notar que o valor da correlação é alto, quando o sinal de entrada é correlacionado com umas das frequências que o compõe, já, quando correlacionado com frequências que não estão presentes no sinal de entrada, o valor da correlação é baixo.

3.1.2 Goertzel

O mesmo sinal de entrada (Figura 4) foi utilizado no algoritmo de Goertzel, que procurou identificar as frequências de 30 Hz, 450 Hz, 1.1 kHz, 1.8 kHz, 2.3 kHz, 2.7 kHz e 3.2 kHz. Os resultados foram analisados através da magnitude obtida para cada uma das frequências, que são apresentados na Figura 6.

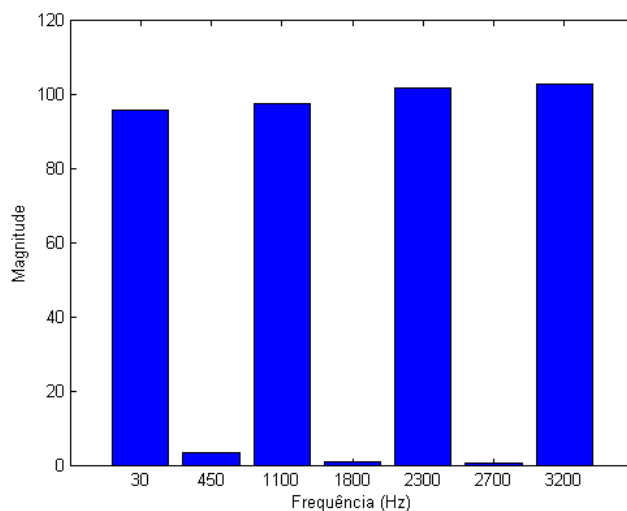


Figura 6 – Magnitude de Goertzel para o sinal da Figura 4 e as frequências de 30 Hz, 450 Hz, 1.1 kHz, 1.8 kHz, 2.3 kHz, 2.7 kHz e 3.2 kHz.

Nota-se que os valores de magnitude são altos para as frequências que estão presentes no sinal de entrada, entretanto, para as frequências que não compõem o sinal de entrada, os valores de magnitude são baixos.

3.1.3 Transformada Rápida de Fourier

Também foi realizado o teste da magnitude do sinal para o algoritmo da Transformada Rápida de Fourier, utilizando o sinal da Figura 4. O resultado é apresentado na Figura 7, que mostra as frequências que compõe o sinal na forma de picos de magnitude. O primeiro pico refere-se a frequência de 30 Hz, o segundo a frequência de 1.1 kHz, o terceiro a frequência de 2.3 kHz e o quarto a frequência de 3.2 kHz.

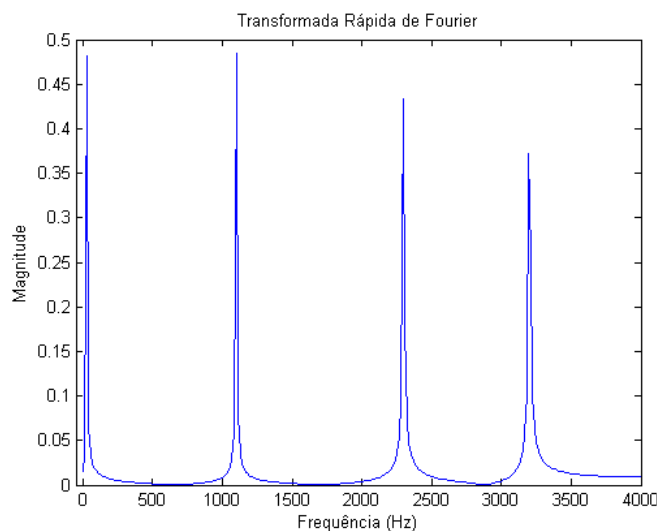


Figura 7 – Resultado da FFT para o sinal da Figura 4.

Diferente do algoritmo de Correlação e do algoritmo de Goertzel, a FFT fornece como resultado todo o espectro de frequências possíveis para um determinado sinal de entrada. Os vales da Figura 7 representam a ausência daquelas frequências no sinal de entrada, enquanto os picos representam a correta detecção das frequências que compõe o sinal da Figura 4.

3.2 Testes dos algoritmos em um ambiente simulado

Foram realizados testes para verificar o funcionamento de cada algoritmo na identificação de diferentes frequências, em um ambiente que simula as características de uma rede elétrica. O sinal utilizado nos testes foi capturado diretamente da rede elétrica, com o auxílio de um osciloscópio digital.

A taxa de amostragem utilizada para captura do sinal foi de 25 kHz. O sinal capturado era composto somente pela fundamental da rede elétrica, entretanto, outras frequências foram adicionadas a fundamental para realização dos testes. Um filtro foi utilizado antes da entrada de cada algoritmo para minimizar a influência da fundamental da rede elétrica. Frequências menores do que 500 Hz eram atenuadas por este filtro.

A faixa de frequências utilizados nos testes foi limitada de 1 kHz a 4 kHz. Esta limitação foi imposta para simplificar os procedimentos de testes e a análise dos resultados.

3.2.1 Teste de variação de potência

Variando a potência dos sinais adicionados sinteticamente à fundamental, verifica-se a eficácia dos algoritmos na detecção das frequências que compunham o sinal.

O tamanho do vetor de entrada adotado é igual a um ciclo da fundamental. Com uma taxa de amostragem de 25 kHz, tem-se 417 amostras em um ciclo de 60 Hz (17 milissegundos). Para o algoritmo da FFT, é necessária uma quantidade de amostras que seja potência de dois, por isso, o sinal de entrada foi preenchido com zeros até se obter o tamanho de 512 amostras.

Para cada rotina de teste, adicionava-se uma frequência a fundamental e realizava-se o procedimento. O conjunto de frequências utilizadas nos testes é: 1 kHz, 2 kHz, 3 kHz e 4 kHz.

3.2.1.1 Correlação

A Figura 8 apresenta o resultado do procedimento de teste realizado para o algoritmo de Correlação. Os gráficos apresentam os valores de correlação para um sinal composto pela fundamental da rede elétrica mais uma frequência do conjunto de frequências, injetada com diferentes potências. Os gráficos 'a', 'b' e 'c' apresentam os valores de correlação para a frequência de 1 kHz injetada à fundamental da rede elétrica com 1V, 2.5V e 5V de potência, respectivamente. Os gráficos 'd', 'e' e 'f' apresentam os valores de correlação para a frequência de 2 kHz. Os gráficos 'g', 'h' e 'i' apresentam os valores de correlação para a frequência de 3 kHz. E os gráficos 'j', 'k' e 'l' apresentam os valores de correlação para a frequência de 4 kHz.

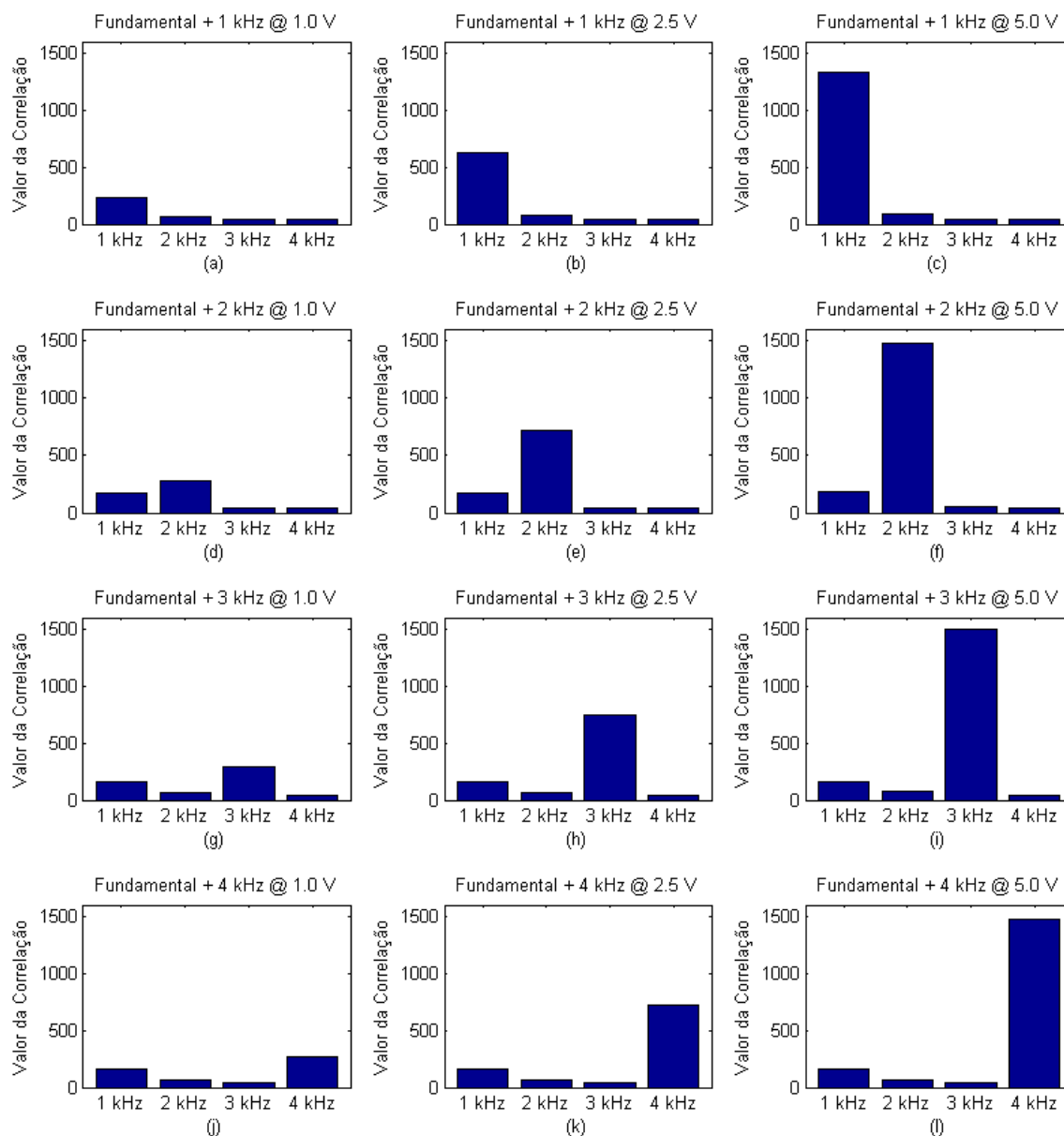


Figura 8 – Resultado da correlação para 1 kHz, 2 kHz, 3 kHz, 4 kHz variando a potência do sinal injetado a fundamental da rede elétrica.

Para o sinal injetado com potência de 1V (Gráficos ‘a’, ‘d’, ‘g’ e ‘j’), a diferença média entre o valor de correlação da frequência que está presente no sinal de entrada e o maior valor de correlação da frequência que não está presente no sinal, é de 47%. Para o sinal injetado com potência de 2.5V (Gráficos ‘b’, ‘e’, ‘h’ e ‘k’), a diferença média é de 79%. E para o sinal injetado com potência de 5V (Gráficos ‘c’, ‘f’, ‘i’ e ‘l’), a diferença média é de 90%. Observa-se que quanto maior for a potência do sinal injetado, maior será a

diferença do valor de correlação entre a frequência que esta presente no sinal de entrada, e as outras frequências do conjunto.

3.2.1.2 Goertzel

A Figura 9 apresenta o resultado do teste para o algoritmo de Goertzel. Os gráficos apresentam os valores obtidos para um sinal composto pela fundamental da rede elétrica mais uma frequência do conjunto de frequências. Os gráficos 'a', 'b' e 'c' apresentam os valores de Goertzel para a frequência de 1 kHz injetada à fundamental da rede elétrica com 1V, 2.5V e 5V de potência. Os gráficos 'd', 'e' e 'f' apresentam os valores de Goertzel para a frequência de 2 kHz. Os gráficos 'g', 'h' e 'i' apresentam os valores de Goertzel para a frequência de 3 kHz. E os gráficos 'j', 'k' e 'l' apresentam os valores de Goertzel para a frequência de 4 kHz.

A diferença média entre o valor de Goertzel da frequência que esta presente no sinal de entrada com o maior dos valores das frequências que não estão presentes no sinal, para as frequências injetadas com 1V de potência (Gráficos 'a', 'd', 'g' e 'j'), é de 49%. Para frequências injetadas com 2.5V de potência (Gráficos 'b', 'e', 'h' e 'k'), a diferença média é de 81%. E para as frequências injetadas com 5V de potência (Gráficos 'c', 'f', 'i' e 'l'), a diferença média é de 91%.

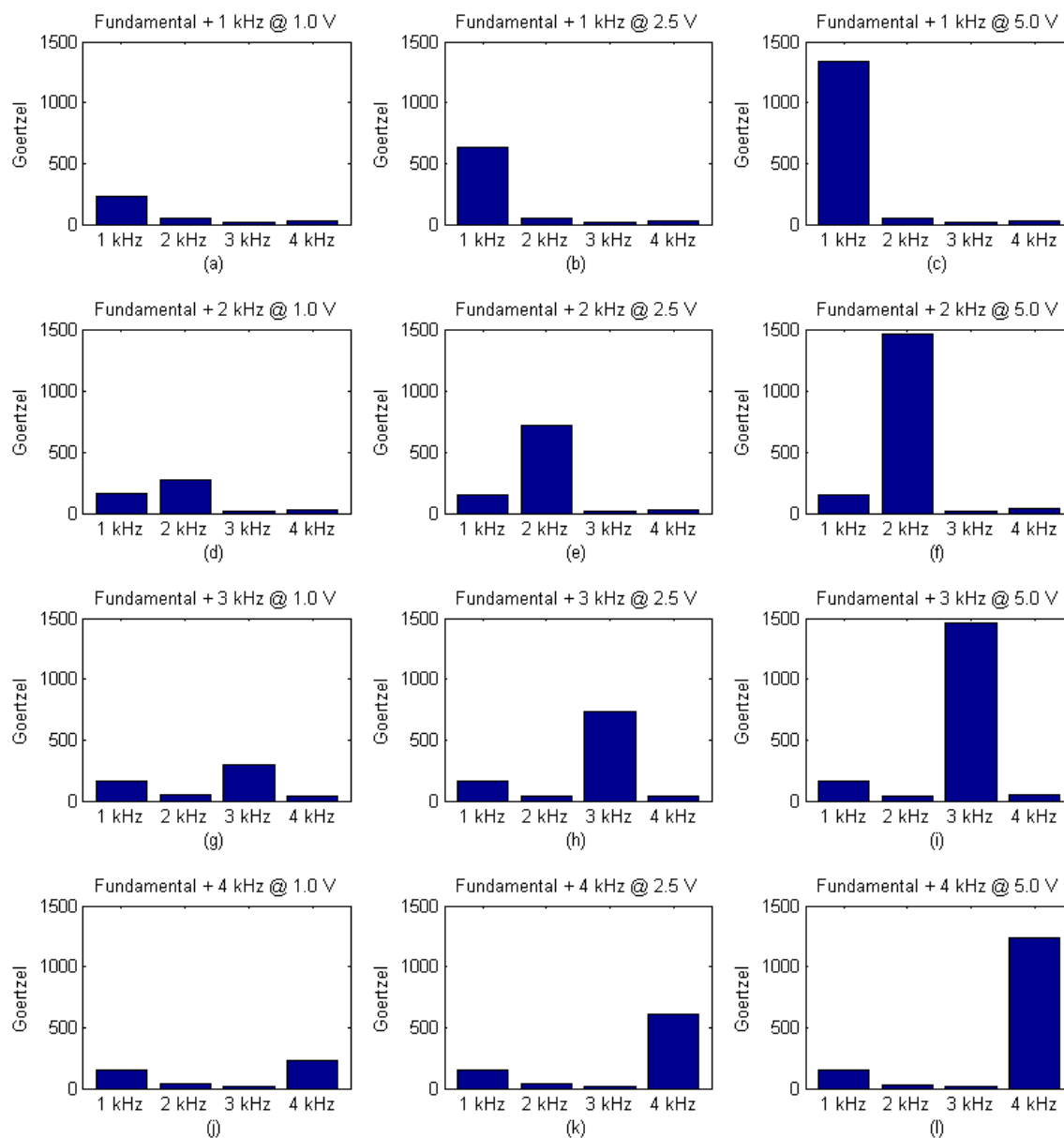


Figura 9 - Resultado do algoritmo de Goertzel para 1 kHz, 2 kHz, 3 kHz, 4 kHz variando a potencia do sinal injetado a fundamental da rede elétrica.

3.2.1.3 Transformada Rápida de Fourier

A Figura 10 apresenta o resultado do teste de variação de potência do sinal injetado na rede elétrica para o algoritmo da FFT. Os gráficos 'a', 'b' e 'c' apresentam o espectro de frequências resultante da FFT para um sinal composto pela fundamental da rede elétrica mais uma frequência de 1 kHz, injetada com 1V, 2.5V e 5V de potência, respectivamente. Os gráficos 'd', 'e' e

'f' para a frequência de 2 kHz. Os gráficos 'g', 'h' e 'i' para a frequência de 3 kHz. E os gráficos 'j', 'k' e 'l' para a frequência de 4 kHz.

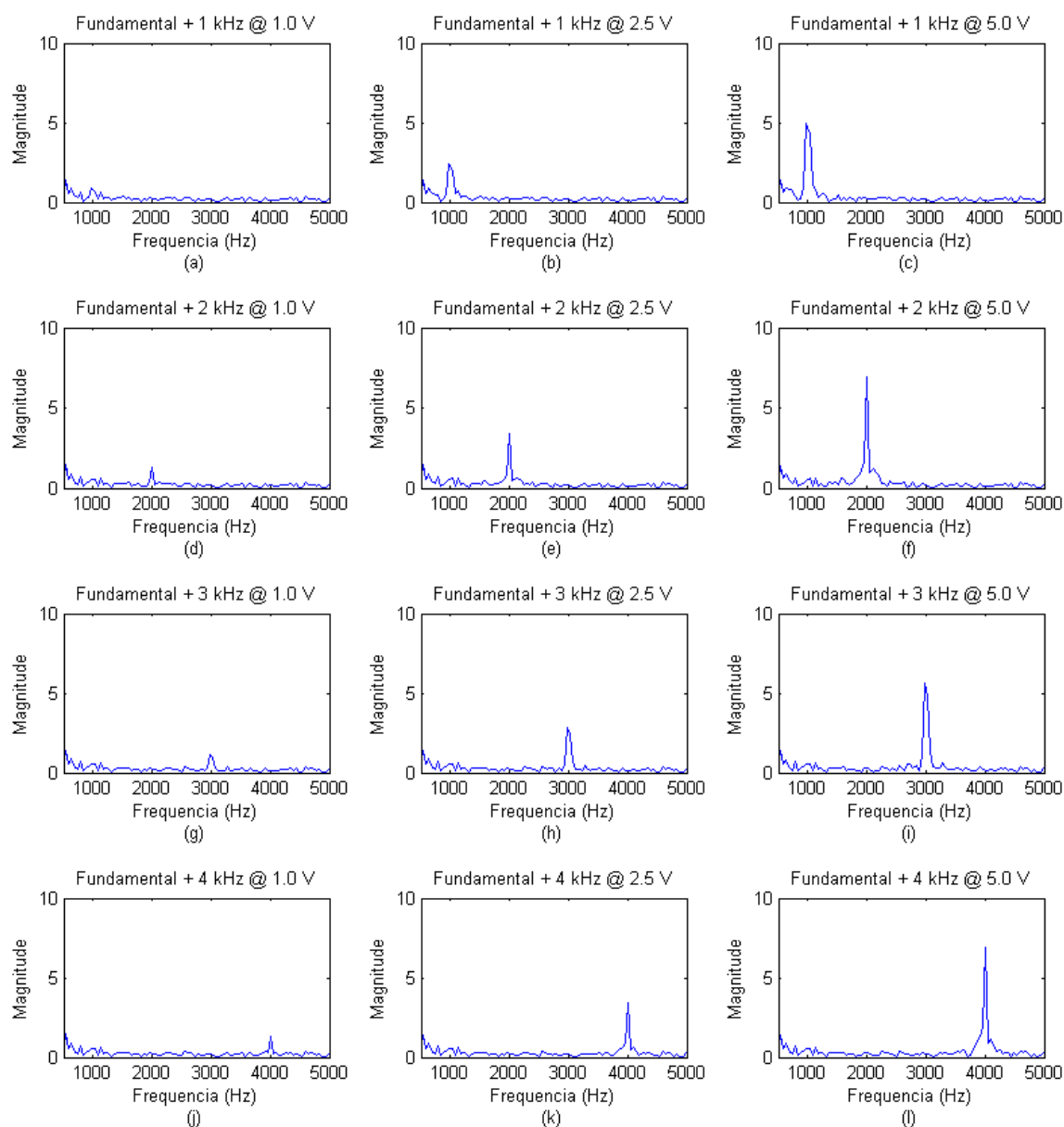


Figura 10 – Espectro de frequências da FFT variando a potência do sinal injetado a fundamental da rede elétrica.

Os picos apresentados nos gráficos representam as magnitudes das frequências no sinal de entrada. Com uma potência de 1V a magnitude média encontrada pelo algoritmo da FFT é de 1.17dB, com 2.5V a magnitude média é de 3dB e com 5V a magnitude média é de 6.1dB de potência.

3.2.2 Teste de variação do sinal de entrada

Nos testes realizados anteriormente, a quantidade de amostras utilizada para o sinal de entrada foi a mesma para os três algoritmos, um ciclo da fundamental da rede elétrica (com taxa de amostragem de 25 kHz, tem-se 417 amostras, 512 para FFT). Entretanto, esta quantidade de amostras pode ser demasiadamente grande para se obter bons resultados para alguns dos algoritmos.

Para testar o funcionamento de cada algoritmo, usando uma quantidade de amostras mais adequada a cada um, foi realizado um teste variando o tamanho do sinal de entrada, para identificar as frequências que compunham o sinal. As frequências são injetadas a fundamental da rede elétrica com uma potência de 5V.

3.2.2.1 Correlação

Para o algoritmo de Correlação funcionar de maneira correta, é necessária uma quantidade de amostras mínima para se formar um ciclo da menor frequência do conjunto de frequências. Utilizando o conjunto de frequências definido anteriormente (1 kHz, 2 kHz, 3kHz e 4 kHz), e uma taxa de amostragem de 25 kHz, obtêm-se uma quantidade de amostras igual a 25.

A Figura 11 apresenta o resultado do teste para o algoritmo de Correlação. Iniciou-se o teste com uma quantidade de amostras igual a quatro vezes o mínimo necessário, diminuindo-se este tamanho por dois até o mínimo. Os gráficos 'a', 'b', 'c' e 'd' mostram o resultado da correlação para um sinal composto pela fundamental da rede elétrica mais uma das frequências de 1 kHz, 2 kHz, 3 kHz e 4 kHz, respectivamente, com uma quantidade de amostras igual a 100. Os gráficos 'e', 'f', 'g' e 'h' para uma quantidade de amostras igual a 50, e os gráficos 'i', 'j', 'k' e 'l' para uma quantidade de amostras igual a 25.

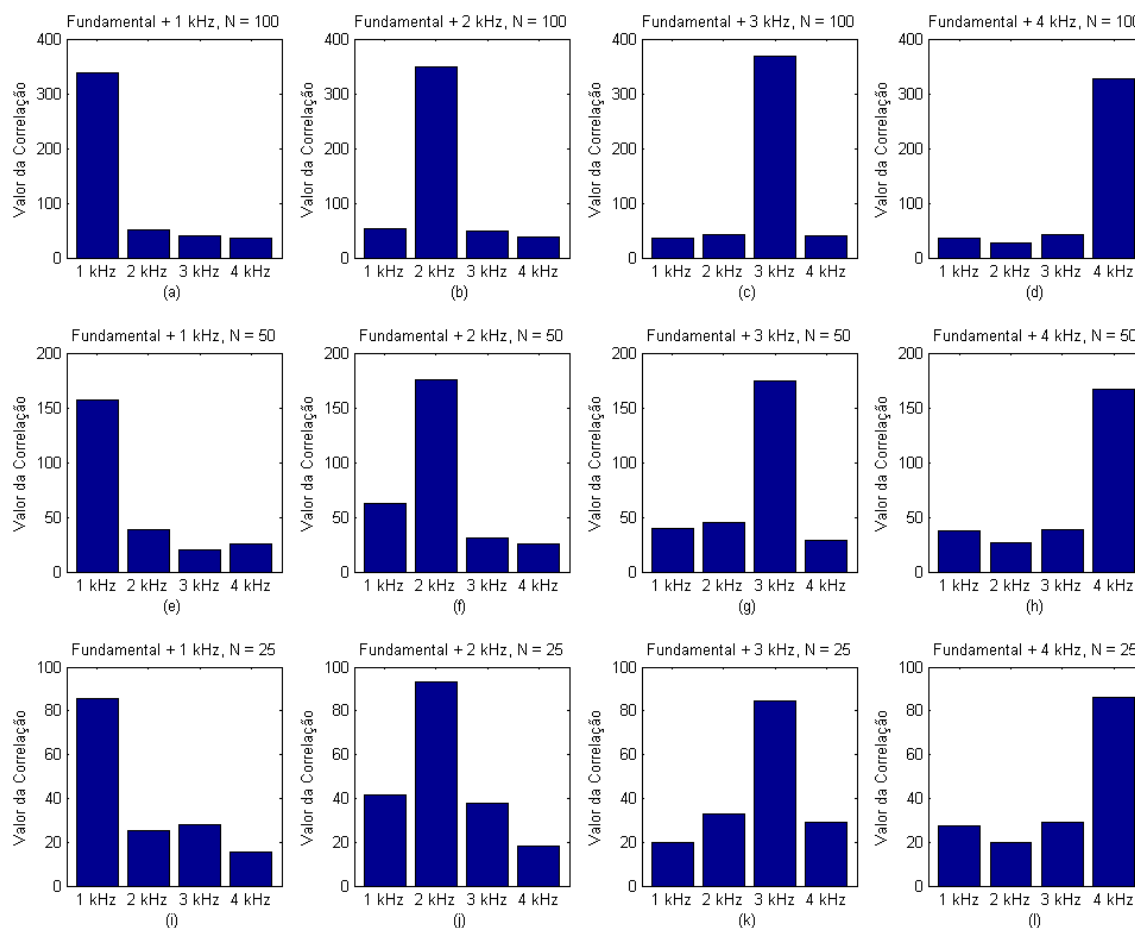


Figura 11 – Resultado do algoritmo de Correlação variando-se a quantidade de amostras.

A magnitude dos valores de correlação diminui na mesma proporção em que se diminui a quantidade de amostras. A diferença média entre os valores de correlação das frequências que estão presentes no sinal de entrada com as frequências que não estão presentes, para uma quantidade de amostras igual a 100 foi de 86%, para uma quantidade de amostras igual a 50 foi de 73% e para uma quantidade de amostras igual a 25 foi de 62%. Estas diferenças mostram onde um valor de correlação mínimo pode ser definido para se determinar quando ou não da detecção de uma frequência.

Observa-se ainda que com uma quantidade de amostras mínimas, os valores de correlação obtidos continuam corretos. Diferenciando entre as frequências presentes e ausentes no sinal de entrada.

3.2.2.2 Goertzel

A quantidade mínima de amostras para o sinal de entrada para o algoritmo de Goertzel é definida pela resolução do espectro de frequências que se deseje alcançar. Ela determina o quão próximo duas frequências podem estar sem que elas se fundam em um mesmo pico de detecção. A resolução do espectro de frequências é definida pela relação entre a taxa de amostragem e a quantidade de amostras do vetor de entrada. Dividindo a taxa de amostragem pela quantidade de amostras, obtêm-se a resolução.

A Figura 12 apresenta o resultado do teste para o algoritmo de Goertzel. Iniciou-se o teste com uma quantidade de amostras igual a 128 (Gráficos 'a', 'b', 'c' e 'd'), obtendo-se uma resolução de 195 Hz. A cada iteração diminuía-se pela metade a quantidade de amostras, passando por 64 amostras (Gráficos 'e', 'f', 'g' e 'h'), e alcançado um mínimo de 32 amostras (Gráficos 'i', 'j', 'k' e 'l'), com uma resolução igual 780 Hz.

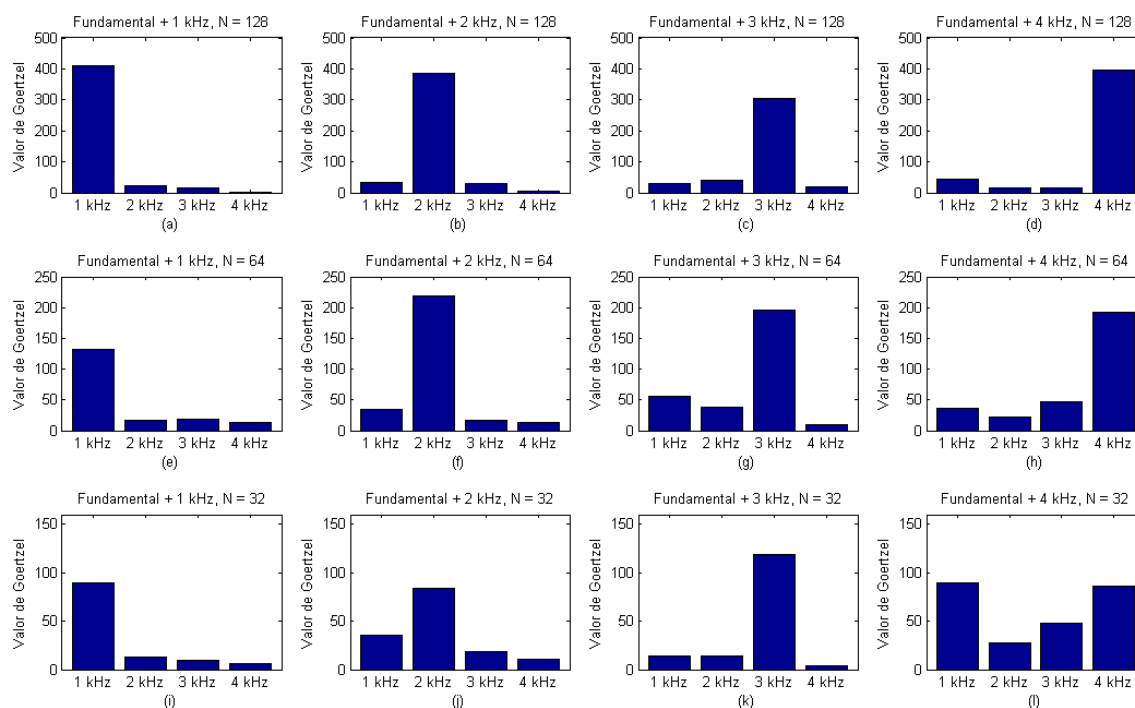


Figura 12 – Resultado para o algoritmo de Goertzel variando-se a quantidade de amostras.

Observa-se que sinais de entrada maiores implicam em valores de Goertzel maiores. A diferença média, entre os valores de Goertzel para frequências que compõem o sinal e o maior dos valores das frequências que

não compõem o sinal, foi de 91% para uma quantidade de amostras igual a 128, e 79% para uma quantidade de amostras igual a 64. Porém, para uma quantidade de amostras igual a 32, obtiveram-se resultados incorretos, como é mostrado pelo gráfico 'l'. O valor de Goertzel para a frequência de 4 kHz, que compõe o sinal de entrada, é menor do que o valor de Goertzel para a frequência de 1 kHz. Desta forma, o tamanho mínimo do sinal de entrada para o algoritmo de Goertzel funcionar corretamente é de 64 amostras.

3.2.2.3 Transformada Rápida de Fourier

Para o algoritmo da FFT, a quantidade de amostras para o sinal de entrada é definida da mesma maneira que para o algoritmo de Goertzel, levando-se em conta a resolução que se deseja alcançar. A resolução do espectro de frequências da FFT é definida pela taxa de amostragem e a quantidade de amostras do sinal de entrada.

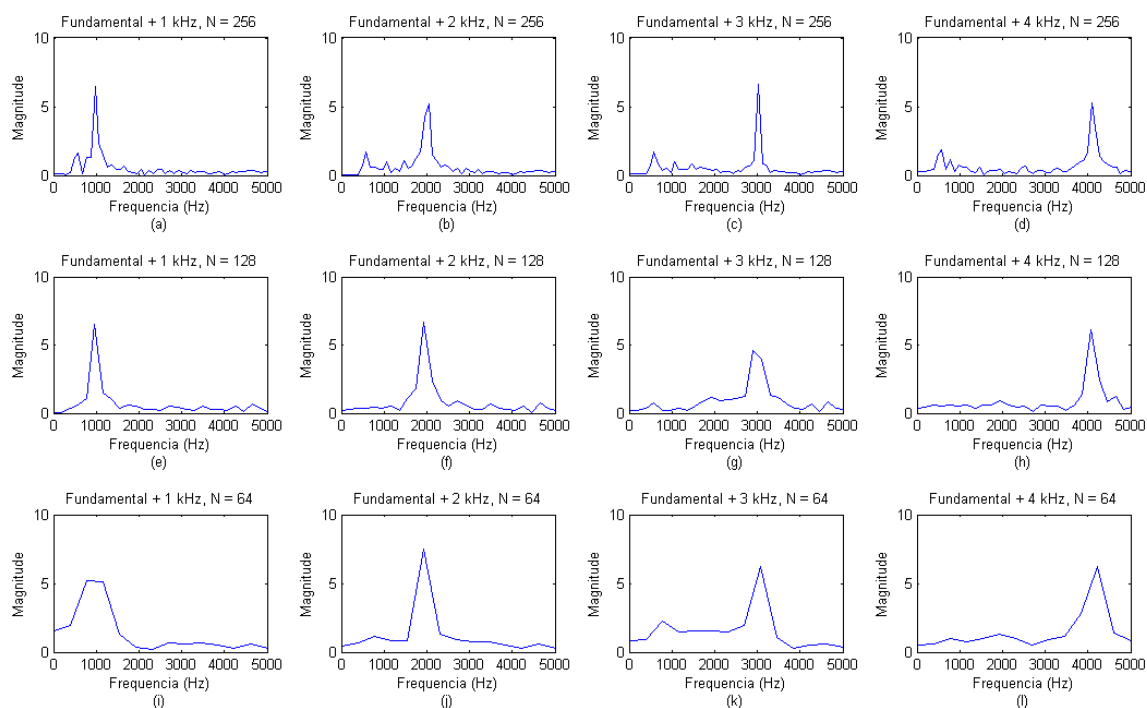


Figura 13 – Resultado da FFT variando-se a quantidade de amostras do sinal de entrada utilizado.

A Figura 13 apresenta o resultado do teste para o algoritmo da FFT. Iniciou-se o teste com uma quantidade de amostras igual a 256 (Gráficos 'a',

'b', 'c' e 'd') e a cada iteração diminuía-se o tamanho do sinal de entrada pela metade, até um mínimo de 64 amostras (Gráficos 'i', 'j', 'k' e 'l').

Os picos dos gráficos com quantidades de amostras maiores são mais pontiagudos devido a resolução da FFT. Os picos observados antes de 1kHz nos gráficos 'a', 'b' e 'c', se devem a influência da fundamental da rede elétrica. Diminuindo-se a quantidade de amostras e conseqüentemente a resolução, este pico se funde com o pico em 1 kHz para sinais compostos pela fundamental da rede elétrica mais 1 kHz (gráficos 'e' e 'i'). Como o conjunto de frequências utilizado possui um espaçamento de pelo menos 1 kHz entre as frequências, uma quantidade de amostras igual a 64 demonstrou-se suficiente para identificar as frequências que compõe o sinal.

3.2.3 Análise comparativa

Os três algoritmos se comportaram de maneira adequada nos testes para um sinal injetado a rede elétrica com diferentes potências. Mesmo com sinais injetados com 1V de potência, foi possível identificar as frequências que compunham o sinal. Entretanto a quantidade de amostras utilizada para o sinal de entrada pode ser reduzida para adequar-se melhor a cada algoritmo.

Nos testes de variação do tamanho do sinal de entrada observou-se que, com quantidades de amostras menores e sinais injetados com 5V de potência, ainda é possível identificar as frequências que compõe o sinal. Esta redução no tamanho do sinal de entrada diminui a quantidade de operações realizadas em cada algoritmo e, conseqüentemente, o tempo de execução.

A Tabela 1 apresenta dados sobre a quantidade aproximada de operações necessárias para o cálculo de cada algoritmo. Os tamanhos dos sinais de entrada foram obtidos através dos testes realizados.

Para o algoritmo de Correlação e para o algoritmo de Goertzel, o número de operações refere-se ao cálculo para uma frequência do conjunto de frequências, enquanto que para o algoritmo da FFT, número de operações refere-se ao cálculo de todo o espectro de frequências. Portanto, para se diferenciar "N" frequências em um sinal utilizando o algoritmo de Correlação ou

o algoritmo de Goertzel, o número de operações total deve ser multiplicado por “N”.

Algoritmo	Numero de Operações (Smith, 2003)	Tamanho sinal de entrada	
		N = um ciclo	N = bom
Correlação	N^2 adições N^2 multiplicações	N = 417 347778 operações	N = 25 1250 operações
Goertzel	$4*N$ adições $2*N$ multiplicações	N = 417 2502 operações	N = 64 384 operações
Transformada Rápida de Fourier	$N * \log_2 N$ adições $(N/2) * \log_2 N$ multiplicações	N = 512 6912 operações	N = 64 576 operações

Tabela 1 – Numero de operações algoritmos de Correlação, Goertzel e FFT.

Observa-se que com a adoção de uma quantidade de amostras mais adequada a cada algoritmo, é possível diminuir o número de operações mantendo a eficácia. Houve uma redução de 99,64% na quantidade de operações realizadas para o algoritmo de Correlação, 84,65% para o algoritmo de Goertzel e 91,67% para o algoritmo da FFT. Diminuindo assim, a disparidade entre o número de operações realizadas por cada um.

3.3 Implementação do protótipo

Um protótipo foi desenvolvido para testar a transmissão e identificação de diferentes frequências transmitidas pela rede elétrica. De um lado sinais com diferentes frequências eram injetados na rede elétrica, do outro, capturava-se o sinal e identificaram-se as frequências transmitidas.

3.3.1 Definição do algoritmo

O algoritmo escolhido para fazer a identificação das frequências transmitidas pela rede elétrica foi o algoritmo de Correlação. Este algoritmo foi

escolhido por ter se mostrado eficaz na identificação de diferentes frequências, como demonstraram os testes descritos anteriormente.

Apesar de o algoritmo de Correlação não ser o mais rápido dos três, por realizar mais operações, ele é tão eficaz quanto os outros, se utilizado com os parâmetros adequados. Sua implementação é simples, necessitando apenas de representação numérica inteira, facilitando a implementação em microcontroladores que possuem apenas representação em ponto fixo.

Algumas peculiaridades do algoritmo de Correlação foram levadas em consideração para a definição da frequência de amostragem e a quantidade de amostras utilizadas para o seu processamento. Estas peculiaridades estão descritas em mais detalhes a seguir.

3.3.1.1 Amostras por ciclo

O algoritmo de Correlação pode não funcionar corretamente se utilizado com uma taxa de amostragem próxima ao teorema de Nyquist. A Figura 14 ilustra este problema. O gráfico 'a' apresenta o sinal conhecido de 1.9 kHz (SC1) e o gráfico 'b', o sinal conhecido de 3.8 kHz (SC2). Os gráficos 'c', 'e', 'g' e 'i' representam um sinal de entrada de 3.8 kHz amostrado em diferentes fases. Os gráficos 'd', 'f', 'h' e 'j' apresentam os resultados da correlação para o sinal de entrada dos gráficos exatamente acima a eles com o sinal SC1 e SC2. Os círculos representam o local onde o sinal foi amostrado. A taxa de amostragem utilizada é de 8 kHz.

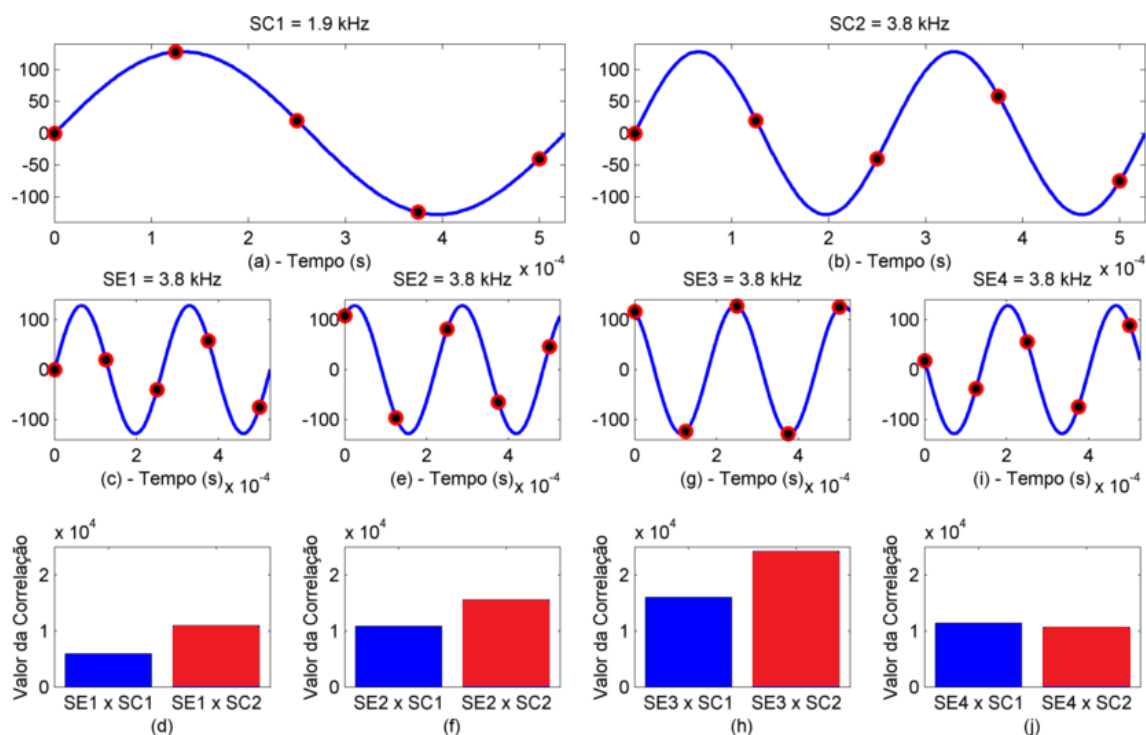


Figura 14 – Influência das amostras no valor da correlação, taxa de amostragem igual a 8 kHz.

Observa-se no gráfico 'j' que, quando se correlacionam o sinal de entrada do gráfico 'i' com os sinais do gráfico 'a' e 'b', obtêm-se valores de correlação incorretos. O valor da correlação do sinal do gráfico 'i' com o sinal do gráfico 'a' é maior do que o valor da correlação do sinal do gráfico 'i' com o sinal do gráfico 'b', sendo que, o sinal do gráfico 'i' e o sinal do gráfico 'b' são compostos pela mesma frequência, porém, em fases diferentes. Isso acontece porque como existem poucos pontos por ciclo, o valor de uma amostra tem grande influência no valor da correlação, e quando este valor fica próximo de zero, ele pode minimizar a influência de um ponto de grande valor no sinal conhecido.

A Figura 15 apresenta o mesmo teste da Figura 14, porém com uma taxa de amostragem maior. Neste teste utilizou-se uma taxa de amostragem de 16 kHz, ou quatro vezes maior do que a frequência mais rápida.

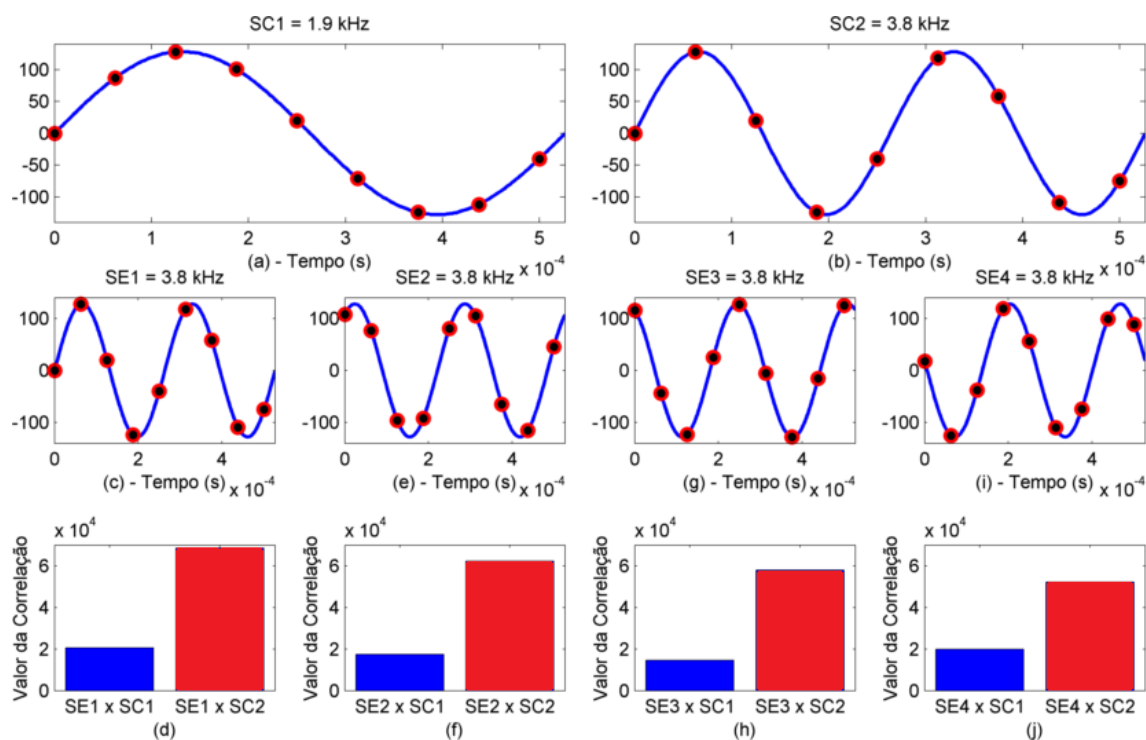


Figura 15 - Influência das amostras no valor da correlação, taxa de amostragem igual a 16 kHz.

Com mais pontos por ciclo da frequência mais alta, observarem-se valores de correlação corretos para o sinal de entrada composto por 3.8 kHz, amostrado em diferentes fases. Além disso, diminui-se a influência de um ponto no valor da correlação e, conseqüentemente, aumenta-se a precisão na identificação das frequências.

3.3.1.2 Quantidade de amostras

A quantidade de amostras mínima para o algoritmo de Correlação é definido por uma relação entre a frequência de amostragem, a frequência mais alta e a frequência mais baixa do conjunto de frequências utilizado.

Para o algoritmo de correlação comportar-se adequadamente, é necessário que existam pontos suficientes para formação de um ciclo da frequência mais baixa. Entretanto, a frequência de amostragem deve ser rápida o suficiente para capturar mais que dois pontos por ciclo da frequência mais rápida (Seção 3.3.1.1).

A Figura 16 apresenta a influência da escolha do conjunto de frequências na quantidade de amostras necessárias para o sinal de entrada. A

frequência mais alta do conjunto é 4 kHz (Gráfico 'b'). Para capturar mais de duas amostras por ciclo de 4 kHz utilizou-se uma taxa de amostragem de 16 kHz. A frequência mais baixa é 1 kHz (Gráfico 'a'). Para formação completa de um ciclo de 1 kHz (1 milissegundo), são necessárias 17 amostras com uma taxa de amostragem de 16 kHz.

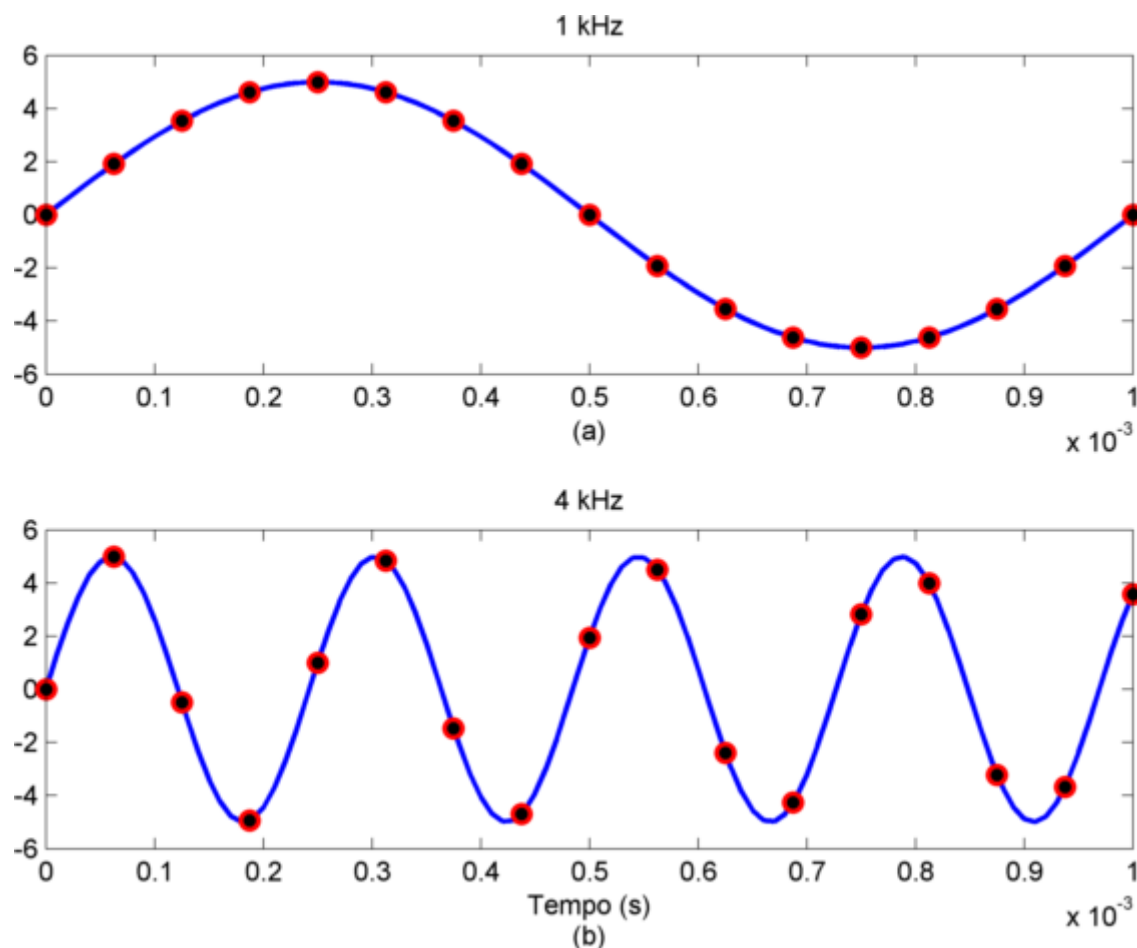


Figura 16 – Quantidade de amostras X Conjunto de frequências

Quanto maior for a diferença entre a frequência mais alta e a frequência mais baixa escolhidas, maior será o tamanho do sinal de entrada e, conseqüentemente, maior será o número de operações necessárias para realizar a correlação dos sinais.

3.3.1.3 Fator de corte

Para determinar se uma frequência esta presente ou não no sinal de entrada, é preciso definir um valor mínimo a ser atingido pelo algoritmo de correlação. A Figura 17 apresenta o fator de corte (linha vermelha), para o conjunto de frequências 1 kHz, 2 kHz, 3 kHz e 4 kHz. Os gráficos ‘a’, ‘b’, ‘c’ e ‘d’ apresentam os valores de correlação para um sinal composto pela fundamental da rede elétrica mais uma frequência de 1 kHz, os gráficos ‘e’, ‘f’, ‘g’ e ‘h’ para a frequência de 2 kHz, os gráficos ‘i’, ‘j’, ‘k’ e ‘l’ para a frequência de 3 kHz e os gráficos ‘m’, ‘n’, ‘o’ e ‘p’ para a frequência de 4 kHz, amostrados em diferentes tempos.

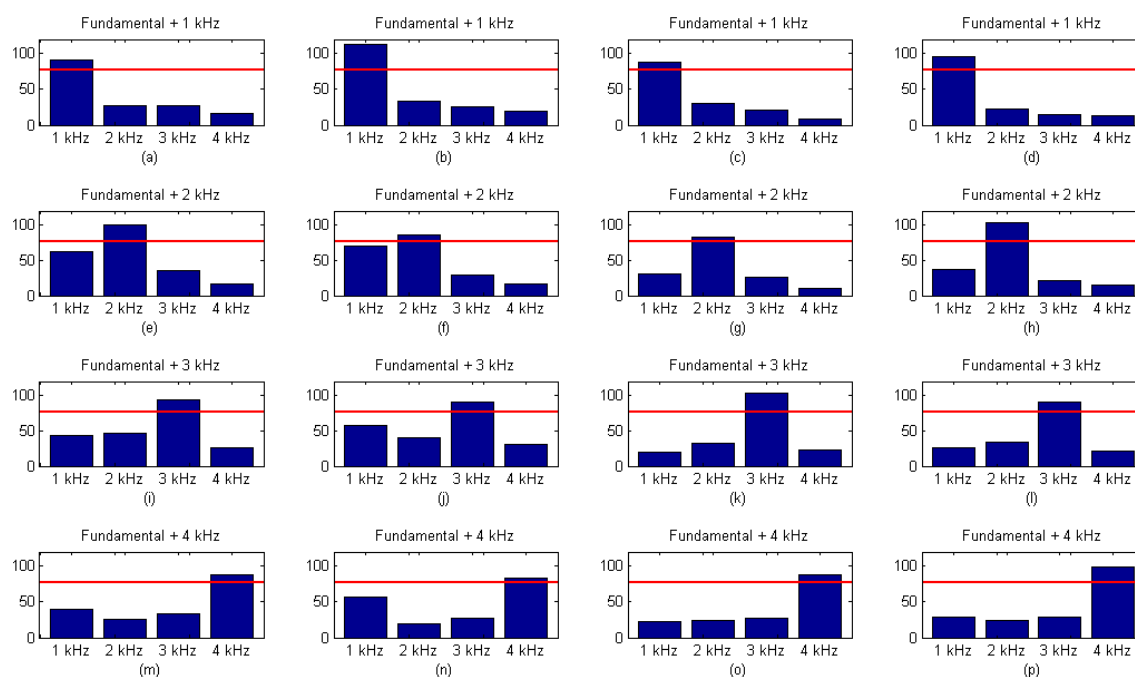


Figura 17 – Fator de corte para o algoritmo de Correlação.

Nota-se que o valor de correlação se altera ao longo do tempo. Para definir um fator de corte é preciso encontrar um valor que fique acima do valor máximo obtido para frequências que não compõem o sinal, e abaixo do valor mínimo obtido para a frequência que compõem o sinal. O valor máximo de correlação encontrado para as frequências que não compõem o sinal é 71, gráfico ‘f’, e o valor mínimo para frequências que compõem o sinal é 83, gráfico ‘n’. O fator de corte (linha vermelha) foi definido pela média dos valores encontrados, obtendo-se um valor de correlação mínimo de 77.

Os valores de correlação podem variar de acordo com os ruídos da rede elétrica e, por isso, decidiu-se por determinar o fator de corte de maneira experimental, observando os valores de correlação obtidos pelo protótipo para cada frequência.

3.3.2 Mensagens

As mensagens trocadas entre transmissor e receptor são formadas por duas frequências. Uma frequência representa um identificador e a outra uma função. As frequências de 1 kHz e 2 kHz foram definidas como identificadores, e as frequências 3 kHz e 4 kHz foram definidas como funções (Ligar/Desligar, por exemplo). Cada frequência é transmitida por um período de tempo de 50 milissegundos, obtendo-se um tempo total de 100 milissegundos por mensagem.

3.3.3 Hardware utilizado

Para o desenvolvimento do protótipo foram utilizados dois microcontroladores da família AVR, AT90CAN128, um conversor digital-analógico da Microchip, MCP4921, e dois kits de desenvolvimento da Atmel, STK500.

O microcontrolador AT90CAN128 possui recursos além dos utilizados na implementação do protótipo, mas estes foram os escolhidos por existirem amostras disponíveis no LISHA. Ele possui um ADC de 10 bits de resolução integrado, utilizado para fazer a captura do sinal no receptor. Uma interface serial para comunicação via SPI com o conversor digital-analógico. Além de temporizadores para controle das rotinas de captura e comunicação.

O conversor digital-analógico MCP4921 foi escolhido por ter um tempo de troca pequeno entre os valores de voltagem fornecidos pelo microcontrolador (4.5 μ s), e por possuir comunicação serial via SPI (Serial Peripheral Interface).

3.3.4 Ambiente de teste

O protótipo foi testado conectando-se o gerador de ondas diretamente ao sistema de correlação. A Figura 18 apresenta um diagrama esquemático do ambiente de teste.

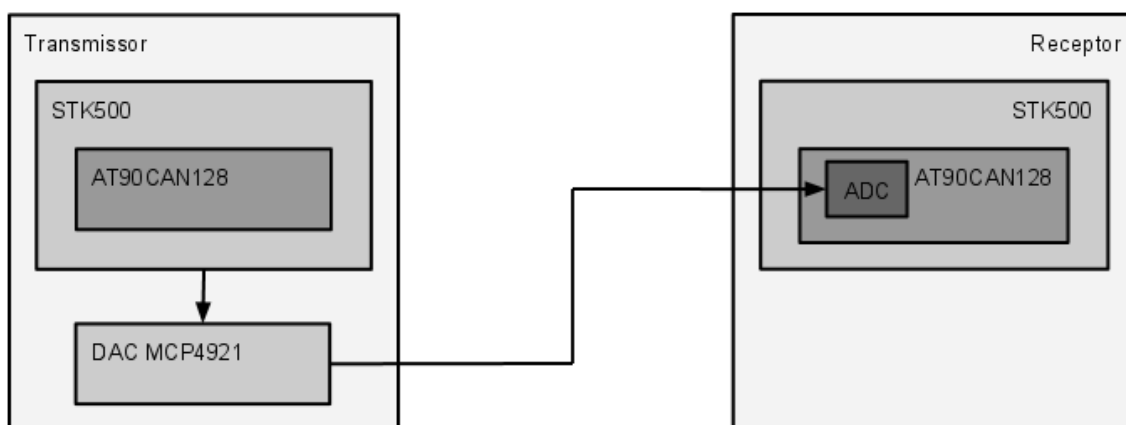


Figura 18 – Diagrama esquemático do ambiente de teste

O transmissor é responsável por gerar as frequências, utilizando para isso o DAC MCP4921. No microcontrolador que controla a geração de ondas, foi criada uma tabela que mantém os valores das senóides das frequências utilizadas. Estes valores são transferidos para o DAC que pode gerar frequências com potências de até 5V de amplitude.

O receptor é responsável por capturar e determinar quais frequências estão presentes no sinal de entrada. A taxa de amostragem utilizada foi de 32 kHz. O clock utilizado no microcontrolador foi de 16 MHz.

Como o sistema não pôde ser testado em um ambiente real devido a falta de um circuito que fizesse a filtragem das frequências e a transformação da tensão da rede elétrica, situações adversas foram criadas para verificar o funcionamento do protótipo. A potência do sinal gerado pelo transmissor foi reduzida pela metade, de 5V para 2.5V, e um ruído de 5mV de potência foi adicionado a cada frequência gerada pelo transmissor.

A atenuação do sinal e o ruído foram gerados sinteticamente pelo Matlab, no momento em que se geravam os valores das senóides que são utilizadas no transmissor. Porém, não foi possível determinar o quanto estas situações seriam comparáveis a ruídos existentes em um ambiente real.

3.3.5 Testes e resultados

No primeiro teste, uma única frequência era transmitida por vez, e o receptor procurava identificá-la. Primeiramente observaram-se os valores de correlação obtidos para cada frequência, para determinação de um fator de corte para o conjunto de frequências. Este foi definido como sendo 50% da média entre os valores de correlação encontrados. O teste foi executado 1000 vezes para cada frequência do conjunto de frequências. Obteve-se uma taxa de erros de menos de 1%.

No segundo teste, mensagens compostas por duas frequências eram transmitidas e o receptor buscava identificar as frequências recebidas. A duração de transmissão de cada frequência foi de 50 milissegundos. Transmitiram-se cada mensagem 1000 vezes, com um pequeno intervalo de tempo entre cada transmissão. A taxa de erros obtida foi de 7%. Esta taxa de erros encontrada, maior do que no primeiro teste, acontece devido ao transmissor gerar ruído quando troca-se a frequência transmitida. Este problema foi resolvido fazendo-se múltiplas identificações antes da determinação da detecção de uma mensagem, reduzindo a taxa de erros para 2%.

Mediu-se também o tempo de execução do algoritmo de Correlação para identificação das quatro frequências do conjunto de frequências. A medição foi feita somente da função que calculava a correlação das frequências. O tempo de execução encontrado foi de 20 milissegundos.

Os resultados obtidos demonstram que o sistema implementado é eficaz na identificação das frequências transmitidas. Além disso, as taxas de erros de menos de 2% demonstram a confiabilidade do sistema. Taxa de erros menores podem ser alcançadas, porém, com um custo ao tempo de execução.

4. CONSIDERAÇÕES FINAIS

Este trabalho apresentou um estudo sobre o funcionamento dos algoritmos de Correlação, Goertzel e Transformada Rápida de Fourier na identificação de mensagens transmitidas em uma rede de energia elétrica. Os três algoritmos se comportaram de maneira adequada nos testes realizados, identificando as frequências que compunham o sinal.

O algoritmo de Correlação foi o escolhido para o desenvolvimento de um protótipo por ser eficaz na identificação das mensagens. Sua simplicidade de implementação e a utilização de somente representação numérica em ponto fixo foram critérios que auxiliaram para a escolha do algoritmo.

Os testes realizados mostraram que para o algoritmo de Correlação funcionar corretamente, é necessária uma taxa de amostragem que capture mais que dois pontos por ciclo da frequência mais alta que se deseja identificar. A quantidade de amostras mínima é definida pelo tempo de formação de um ciclo da frequência mais baixa. Quanto maior for a diferença entre a frequência mais alta e a frequência mais baixa que se deseja identificar, maior será a quantidade de amostras necessárias para realização da correlação.

O protótipo foi desenvolvido fazendo-se as otimizações levantadas pelos estudos realizados. Não foi possível testar o protótipo em um ambiente real, devido a falta de um circuito que fizesse a filtragem das frequências e a transformação da tensão da rede elétrica. Entretanto, criou-se um ambiente de testes onde situações adversas foram inseridas aos sinais gerados, para representar situações do ambiente real. Nestes testes observou-se uma taxa de erros de 2% para transmissão de mensagens compostas por duas frequências e para identificação de frequências únicas o erro foi menor que 1%.

O algoritmo de Correlação se mostrou eficaz na detecção das mensagens. Com uma configuração adequada, foi possível diminuir o número de operações necessárias à sua execução mantendo a eficácia. Mesmo com a utilização de um microcontrolador de baixo desempenho foi possível atingir taxas de erros de 2% com tempos de execução aceitáveis, mostrando-se uma boa opção para detecção de mensagens transmitidas em uma rede de energia elétrica.

4.1 Trabalhos futuros

No decorrer do trabalho surgiram algumas indagações que podem ser contempladas em futuros trabalhos sobre o assunto:

- Estudo de uma forma de sincronia entre transmissor e receptor para adoção de modulação FSK (Frequency-shift keying).
- Estudo para definição de um fator de corte dinâmico para o algoritmo de correlação.
- Estudo sobre o impacto da atenuação do sinal injetado a rede de energia elétrica no sistema de correlação em transmissões em distâncias grandes.
- Implementação do sistema de correlação em microcontroladores mais simples do que o utilizado neste trabalho, que disponham apenas dos recursos necessários a sua execução.

REFERÊNCIAS

SMITH, Steven W. **Digital Signal Processing: A Practical Guide for Engineers and Scientists**. Burlington, MA, EUA. Newnes, 2003.

CHASSAING, Rulph. **Digital Signal Processing and Applications with the C6713 and C6416 DSK**. Hoboken, New Jersey: Wiley Interscience, 2005.

YOUSUF, M. S.; EL-SHAFEI, M. **Power Line Communications: An Overview – Part I**. Innovations in Information Technology, 2007 IIT 07. 4th International Conference on, 2007. 18-20 Nov. 2007. p.218-222.

JOSHI, R. P.; BOSHALE S. et al. **Analysis and Simulation of Noise in Power Line Communication Systems**. Emerging Trends in Engineering and Technology, 2008. ICETET 07. First International Conference on.

YU-JU, L.; LATCHMAN H. A.; et al. **A power line communication network infrastructure for the smart home**. Wireless Communications, IEEE 9(6): 104-111, 2002.

MAJUMDER, A.; CAFFERY J. Jr.. **Power line communications**. Potentials, IEEE 23(4): 4-8, 2004.

PRESS, P. H.; TEUKOLSKY, S. A.; et al. **Numerical Recipes in C: The Art of Scientific Computing**. Second Edition. Disponível em: <<http://apps.nrbook.com/c/index.html>>. Acesso em: 08 jun. 2011.

MELO, M.A.; LEONARDI, F.; LA NEVE, A.; , **"Digital signal processing with MatLab and DSP kits"**, *Digital Signal Processing Workshop, 2004 and the 3rd IEEE Signal Processing Education Workshop. 2004 IEEE 11th*. 2004.

PRANDONI, P.; VETTERLI, M.; **Signal Processing for Communications**. Broken Sound Parkway, NW, EUA. CRC Press, 2008.

SmartHome. Disponível em: <<http://smarhome.lisha.ufsc.br/>>. Acesso em: 02 jul. 2010.

WANNER, Lucas. **CSM213A Project Report: Adaptive Sampling.** 2008.

GOERTZEL, Gerald. **An Algorithm for the Evaluation of Finite Trigonometric Series.** The American Mathematical Monthly, 1958.

MATHWORKS. **Matlab.** Disponível em: <<http://www.mathworks.com>>. Acesso em: 08 jun. 2011.

APÊNDICE A – RECEPTOR – CÓDIGO-FONTE C

```

#include <avr/interrupt.h>

#define SIZE_SE          33
#define SIZE_H           33
#define SIZE_CORR       SIZE_SE + SIZE_H - 1
#define N_DET           2

int8_t freq_1k[] = {
25,49,71,91,106,118,126,127,126,118,106,91,71,49,25,0,-25,-49,-71,-
91,-106,-118,-126,-127,-126,-118,-106,-91,-71,-49,-25,0,25};

int8_t freq_2k[] = { 49,91,118,127,118,91,49,0,-49,-91,-118,-127,-
118,-91,-49,0,49,91,118,127,118,91,49,0,-49,-91,-118,-127,-118,-91,-
49,0,49};

int8_t freq_3k[] = { 71,118,126,91,25,-49,-106,-127,-106,-
49,25,91,126,118,71,0,-71,-118,-126,-91,-25,49,106,127,106,49,-25,-
91,-126,-118,-71,0,71};

int8_t freq_4k[] = {93,127,83,-13,-101,-127,-73,25,108,124,63,-38,-
115,-120,-51,50,120,115,39,-61,-124,-109,-27,72,126,102,14,-82,-127,-
94,-2,92,127};

int8_t buffer[SIZE_SE];
int8_t ig = SIZE_SE-1;
int8_t det_freq[4];

void PORT_Init();
void UART_Init();
void ADC_Init();
void correlation();

int main()
{
    PORT_Init();
    UART_Init();
    ADC_Init();
    sei();
    while (1);
    return 0;
}

void PORT_Init()
{
    PORTA = 0xFF;    DDRA = 0xFF;
    PORTE = 0x00;    DDRE = 0x02;
}

void UART_Init()
{
    uint8_t BaudRate = UBRR_VALUE;
    UBRROH = (uint8_t) (BaudRate>>8);
    UBRROL = (uint8_t) BaudRate;
    UCSROB = UCSROB | 0x18;
    UCSROC = UCSROC | 0x86;
}

void ADC_Init()
{
    TCNT1    = 65036;

```

```

    TCCR1B = 0x01;
    ADCSRA = 0x38;
    ADCSRB = 0x06;
    ADMUX  = 0xE0;
    ADCSRA |= 0x80;
    TIFR1  = 0x01;
}

ISR(ADC_vect)
{
    TIFR1 = 0x01;
    TCNT1 = 65103;
    buffer[ig] = ADCH;

    if((buffer[ig] > 30) || (ig != SIZE_SE-1)){
        ig = ig - 1;
    }

    if(ig == 0)
    {
        ig = SIZE_SE-1;
        correlation();
    }
}

void correlation()
{
    int8_t i; int8_t j; int8_t aux = 0;
    int16_t corr0[SIZE_CORR]; int16_t corr1[SIZE_CORR];
    int16_t corr2[SIZE_CORR]; int16_t corr3[SIZE_CORR];
    int16_t max[] = {0,0,0,0};

    for (i = 0; i < SIZE_CORR; ++i)
    {
        corr0[i] = 0; corr1[i] = 0; corr2[i] = 0; corr3[i] = 0;

        for (j = 0; j <= i; j++)
        {
            aux = i-j;

            if(aux >= SIZE_SE)
                continue;

            if(j >= SIZE_H)
                continue;

            uint8_t b = buffer[aux];

            corr0[i] = corr0[i] + ( ((int16_t) freq_1k[j]*b) >> 5 );
            corr1[i] = corr1[i] + ( ((int16_t) freq_2k[j]*b) >> 5 );
            corr2[i] = corr2[i] + ( ((int16_t) freq_3k[j]*b) >> 5 );
            corr3[i] = corr3[i] + ( ((int16_t) freq_4k[j]*b) >> 5 );
        }

        if(corr0[i] > max[0])
            max[0] = corr0[i];

        if(corr1[i] > max[1])
            max[1] = corr1[i];

        if(corr2[i] > max[2])
            max[2] = corr2[i];
    }
}

```



```

        if(corr3[i] > max[3])
            max[3] = corr3[i];
    }

    int8_t i_aux = 3;
    int16_t tmp = max[3];
    for(i = 0; i < 3 ; i++)
    {
        if(max[i] > tmp)
        {
            i_aux = i;
            tmp = max[i];
        }
    }
    if(det_freq[i_aux] < 254)
        det_freq[i_aux] = det_freq[i_aux] + 1;

    if((det_freq[0] >= N_DET) && (det_freq[2] >= N_DET))
    {
        PORTA &= ~0x30;
        det_freq[0] = 0;
        det_freq[1] = 0;
        det_freq[2] = 0;
        det_freq[3] = 0;
    }

    if((det_freq[0] >= N_DET) && (det_freq[3] >= N_DET))
    {
        PORTA |= 0x30;
        det_freq[0] = 0;
        det_freq[1] = 0;
        det_freq[2] = 0;
        det_freq[3] = 0;
    }

    if((det_freq[1] >= N_DET) && (det_freq[2] >= N_DET))
    {
        PORTA &= ~0xC0;
        det_freq[0] = 0;
        det_freq[1] = 0;
        det_freq[2] = 0;
        det_freq[3] = 0;
    }

    if((det_freq[1] >= N_DET) && (det_freq[3] >= N_DET))
    {
        PORTA |= 0xC0;
        det_freq[0] = 0;
        det_freq[1] = 0;
        det_freq[2] = 0;
        det_freq[3] = 0;
    }
}

```

APÊNDICE B – TRANSMISSOR – CÓDIGO-FONTE C

```

#include <avr/interrupt.h>

#define SELECT_DAC_YES PORTB &= ~0x01
#define SELECT_DAC_NO PORTB |= 0x01
#define TIMER1_INT 55000

#define SIZE_FREQ_1K 81
#define SIZE_FREQ_2K 41
#define SIZE_FREQ_3K 27
#define SIZE_FREQ_4K 20
#define SIZE_FREQ_0K 5

uint16_t freq_1k [] = {
2046,2207,2366,2524,2679,2830,2976,3116,3250,3376,3494,3603,3703,3792,
3871,3938,3994,4037,4069,4088,4094,4088,4069,4037,3994,3938,3871,3792,
3703,3603,3494,3376,3250,3116,2976,2830,2679,2524,2366,2207,2046,1885,
1726,1568,1413,1262,1116,976,842,716,598,489,389,300,221,154,98,55,23,
4,0,4,23,55,98,154,221,300,389,489,598,716,842,976,1116,1262,1413,1568
,1726,1885,2046 };

uint16_t freq_2k [] = {
2046,2366,2679,2976,3250,3494,3703,3871,3994,4069,4094,4069,3994,3871,
3703,3494,3250,2976,2679,2366,2046,1726,1413,1116,842,598,389,221,98,2
3,0,23,98,221,389,598,842,1116,1413,1726,2046 };

uint16_t freq_3k [] = {
2046,2524,2976,3376,3703,3938,4069,4088,3994,3792,3494,3116,2679,2207,
1726,1262,842,489,221,55,0,55,221,489,842,1262,1726 };

uint16_t freq_4k [] = {
2048,2700,3285,3740,4019,4093,3954,3617,3116,2504,1844,1206,655,249,31
,23,226,618,1159,1793 };

uint16_t freq_0k [] = {0,0,0,0,0};

uint16_t * current_freq = freq_0k;
uint16_t * second_freq = freq_0k;

uint8_t size_current_freq = SIZE_FREQ_0K;
uint8_t size_second_freq = SIZE_FREQ_0K;

uint8_t n_int_timer1 = 0;
uint8_t i;

void PORT_Init();
void Timer0_Init();
void ExternalInterrupt_Init();
void SPI_MasterInit();
void SPI_MasterTransmit(uint8_t cData);

int main()
{
PORT_Init();
SPI_MasterInit();
Timer0_Init();
ExternalInterrupt_Init();
sei();
while(1);
return 0;
}

```

```

void PORT_Init ()
{
    PORTA = 0x00;    DDRA = 0xFF;
    PORTD = 0x00;    DDRD = 0x00;
}

void Timer0_Init ()
{
    TCCR0A = 0b00000001;
    TCNT0 = 90;
    TIMSK0 = 0b00000001;
}

void Timer1_Init ()
{
    TCCR1B = 0x03;
    TCNT1 = TIMER1_INT;
    TIMSK1 = 0x01;
}

void ExternalInterrupt_Init ()
{
    EIMSK |= (1 << INT4) | (1 << INT5) | (1 << INT6) | (1 << INT7);
    EICRB |= (1 << ISC40) | (1 << ISC41);
}

void SPI_MasterInit ()
{
    DDRB = (1<<DDB2) | (1<<DDB1) | (1<<DDB0);
    SPCR = (1<<SPE) | (1<<MSTR);
}

void SPI_MasterTransmit(uint8_t cData)
{
    SPDR = cData;
    while(!(SPSR & (1<<SPIF)));
}

/* Timer0 Overflow*/
ISR (SIG_OVERFLOW0)
{
    TCNT0 = 90;
    uint16_t aux = current_freq[i++];
    SELECT_DAC_YES;
    SPI_MasterTransmit(0x70 | (aux >> 8));
    SPI_MasterTransmit((uint8_t) aux);
    SELECT_DAC_NO;

    if (i==size_current_freq)
        i=0;
}

/* Timer1 Overflow */
ISR(SIG_OVERFLOW1)
{
    TCNT1 = TIMER1_INT;
    n_int_timer1 += 1;

    if(n_int_timer1 == 1)
    {
        current_freq = second_freq;
        size_current_freq = size_second_freq;
    }
}

```

```

    }

    if(n_int_timer1 == 2)
    {
        current_freq = freq_0k;
        size_current_freq = SIZE_FREQ_0K;
        n_int_timer1 = 0;
        TCCR1B = 0x00;
        TIMSK1 = 0x00;
    }
}

/* Button 4 */
ISR(INT4_vect)
{
    Timer1_Init();
    current_freq = freq_1k;
    size_current_freq = SIZE_FREQ_1K;
    second_freq = freq_3k;
    size_second_freq = SIZE_FREQ_3K;
    i = 0;
    PORTA = ~0x10;
}

/* Button 5 */
ISR(INT5_vect)
{
    Timer1_Init();
    current_freq = freq_1k;
    size_current_freq = SIZE_FREQ_1K;
    second_freq = freq_4k;
    size_second_freq = SIZE_FREQ_4K;
    i = 0;
    PORTA = ~0x20;
}

/* Button 6 */
ISR(INT6_vect)
{
    Timer1_Init();
    current_freq = freq_2k;
    size_current_freq = SIZE_FREQ_2K;
    second_freq = freq_3k;
    size_second_freq = SIZE_FREQ_3K;
    i = 0;
    PORTA = ~0x40;
}

/* Button 7 */
ISR(INT7_vect)
{
    Timer1_Init();
    current_freq = freq_2k;
    size_current_freq = SIZE_FREQ_2K;
    second_freq = freq_4k;
    size_second_freq = SIZE_FREQ_4K;
    i = 0;
    PORTA = ~0x80;
}

```

ANEXO A – ALGORITMO DE CORRELAÇÃO

```

function corr = correlation(input_signal,known_signal)
    size_corr = length(input_signal)+length(known_signal)-1;
    corr = zeros(1,size_corr);
    y=input_signal(length(input_signal):-1:1);
    input_signal = y;
    for i=1:length(corr)
        for j=1:i
            if (i - j + 1 > length(input_signal))
                continue
            end
            if(j > length(known_signal))
                continue
            end
            corr(i) = corr(i) + known_signal(j)*input_signal(i-j+1);
        end
    end
end

```

ANEXO B – ALGORITMO DE GOERTZEL

```

function result = goertzel(n, sampling_rate, input_signal,
                           target_frequency)
    k = floor(0.5+((n*target_frequency)/sampling_rate) );
    omega = (2*pi*k)/n;
    sine = sin(omega);
    cosine = cos(omega);
    coeff = 2*cosine;
    q(1:3) = 0;
    for i = 1:n
        r(1) = coeff*q(2)-q(3)+input_signal(i);
        r(3) = q(2);
        r(2) = r(1);
        q = r;
    end
    m = q(2)*q(2)+q(3)*q(3)-q(2)*q(3)*coeff;
    result = sqrt(m);
end

```

ANEXO C – ALGORITMO DA TRANSFORMADA RÁPIDA DE FOURIER

```

function result = fft(input_signal)
    n = length(input_signal); t = log2(n); nv2 = n/2; j=1;
    for i=1:(n-1)
        if i<j
            temp = input_signal(j);
            input_signal(j) = input_signal(i);
            input_signal(i) = temp;
        end
        k = nv2;
        while k<j
            j = j-k;
            k = k/2;
        end
        j = j+k;
    end
    w = zeros(n-1,1); k=1; L=2;
    while L<=n
        theta = 2*pi/L;
        for j=0:(L/2 - 1)
            w(k) = complex(cos(j*theta), -sin(j*theta));
            k = k+1;
        end
        L = L*2;
    end
    for q=1:t
        L = 2^q; r = n/L; L2 = L/2;
        for k=0:(r-1)
            for j=0:(L2-1)
                temp = w(L2-1+j+1)*input_signal(k*L+j+L2+1);
                input_signal(k*L+j+L2+1) = input_signal(k*L+j+1)-temp;
                input_signal(k*L+j+1) = input_signal(k*L+j+1)+temp;
            end
        end
    end
    result = input_signal;
end

```