TÉCNICAS PARA IMPLEMENTAÇÃO DE CONTROLADORES EM PROCESSADORES DE PONTO FIXO

Guilherme Feulo do Espirito Santo e Paulo Henrique da Rocha
Centro Tecnológico da Marinha em São Paulo - CTMSP

INTRODUÇÃO

Por apresentarem baixo custo, alta velocidade de processamento e grande portabilidade, os processadores de ponto fixo têm sido cada vez mais utilizados na implementação de controladores eletrônicos.
O maior desafio da implementação de um processador de ponto fixo em um controlador reside em escolher uma resolução adequada para o processamento das informações que foram pensadas para ponto flutuante.
Nos últimos anos, diversas técnicas de otimização foram propostas para transformar algoritmos implementados em ponto flutuante para ponto fixo, sem com isso degradar o desempenho do controlador [1].
Ainda nesse cenário de busca de técnicas para a implementação de algoritmos em DSP's de ponto fixo os fabricantes de ferramentas computacionais desenvolveram ferramentas para facilitar o manuseio das informações em ponto fixo e o trabalho dos projetistas, como por exemplo, os *ToolBox de ponto fixo no Matlab/Simulink* (da Matworks) e a *biblioteca IQMath (da Texas Instruments)*.

OBJETIVO

O escopo deste trabalho é a otimização computacional e o desenvolvimento de uma interface gráfica para o algoritmo minimização da norma H_∞ do erro entre o dado em ponto flutuante e ponto fixo, por meio do algoritmo de seleção genética, proposto em [1]. Para isso será utilizada a linguagem MATLAB juntamente com o *ToolBox de ponto fixo*.

METODOLOGIA

O desenvolvimento do programa foi dividido em 4 etapas.

1ª Etapa: Entendimento do problema e embasamento teórico.

Para elaborar a otimização computacional do programa foi necessário um conhecimento aprofundado tanto do problema quanto da teoria que o permeia, e por isso foram estudados: sistemas de armazenamento de dados em ponto fixo e em ponto flutuante [1,2], algoritmo genético, obtenção de leis controle biquadráticas através de pólos e zeros [3], além do estudo do próprio programa, sua funcionalidade e variáveis de entrada e saída.

2ª Etapa: Levantamento das necessidades do programa.

Nesta etapa foi feita a apuração do que seria necessário que o programa realiza-se: tipos de entrada e flexibilidade do programa quanto ao problema. Nesta etapa também foi feito um estudo da viabilidade da execução do programa bem como uma estimativa do seu tempo de execução.

3ª Etapa: Elaboração do código

Após a definição de todas as necessidades do programa bem como as limitações do problema, foi dado início ao desenvolvimento do código.

O desenvolvimento do código foi feito tendo como base o algoritmo definido na etapa anterior, e divido em partes independentes:

Corpo do programa: a função principal que recebe as entradas, manipula os dados para sejam utilizados nas funções auxiliares e organiza o funcionamento do programa.

Funções auxiliares: as funções auxiliares além de tornarem o código mais organizado e conciso, têm por objetivo realizar cada parte do processo independentemente, são funções auxiliares as funções de ordenação de vetor, cálculo de erro, definição de formato Q_i dos coeficientes, etc.

Interface gráfica: A interface gráfica tem por objetivo distanciar o usuário do código do programa evitando assim modificações indesejadas. A interface é responsável por coletar os dados de entrada fornecidos pelo usuário e repassar ao programa, e após o termino da execução exibir o resultado obtido.

4ª Etapa: Testes de execução e aperfeiçoamento

Por fim foram realizados alguns testes de execução com leis de controle geradas por pólos e zeros e obtenção dos coeficientes por meio do processo descrito em [3]. Com base nos testes foram feitos pequenos ajustes no código do programa para um melhor desempenho.

RESULTADOS

Foram realizados testes práticos com o programa e a determinação dos formatos Q_i e foi obtido que o programa necessita em média de 200 gerações de indivíduos para encontrar o erro mínimo, o que resulta em um tempo de execução de aproximadamente 7 horas, o que é uma diminuição significativa no tempo de execução que anteriormente era de 12 horas.

CONCLUSÕES

Os resultados obtidos mostram que o programa implementado atende ao esperado, além disso, a interface gráfica facilita em muito o uso desta técnica, por parte do usuário, além de propiciar novas ferramentas para a determinação dos formatos Q_i.
Também vale a pena ressaltar a flexibilidade dada ao programa nesta versão, já que o mesmo aceita um número definido de blocos de biquadráticos (2 pólos e 2 zeros) na lei de controle, e coeficientes reais de qualquer grandeza, calculando com a mesma eficácia seus formatos Q_i.

REFERÊNCIAS BIBLIOGRÁFICAS

[1]Rocha, Paulo Henrique da; Porsh, Michael Cláudio; Ferreira, Henrique Cezar; Sales Roberto Moura. **Implementação de controladores em DSP de ponto-fixo**, Revista Pesquisa Naval nº20 (2007), p. 9-17

[2]Rocha, Paulo Henrique da. **Controle H_∞ não-linear aplicado em sistemas de levitação magnética: projeto e implementação em DSP de ponto-fixo**, Escola Politécnica da Universidade de São Paulo (2009). p. Tese (Doutorado).

[3]Rocha, Paulo Henrique da; Porsh, Michael Cláudio. **Utilização de técnicas de processamento digital de sinais para implementação de funções biquadraticas**, Pesquisa Naval (SDM), v. 16, p. 1-6 (2003)

[4]Rocha, Paulo Henrique da; Porsh, Michael Cláudio; Ferreira, Henrique Cezar; Sales Roberto Moura. **Fixed-point DSP implementation of nonlinear H_∞ controller for large gap electromagnetic suspension system**, Control Engineering Practice 17, (2009), p. 1148-1156.

APOIO FINANCEIRO AO PROJETO

CNPq/PIBITI