

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA

**Desenvolvimento de Bancada Didático-
Experimental de Baixo Custo para Aplicações
em Controle Ativo de Vibrações**

Autor: Maurício José Amorim

Orientador: Prof. Dr. Eurípedes Guilherme de Oliveira Nóbrega

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA
DEPARTAMENTO DE MECÂNICA COMPUTACIONAL

Desenvolvimento de Bancada Didático- Experimental de Baixo Custo para Aplicações em Controle Ativo de Vibrações

Autor: **Maurício José Amorim**

Orientador: **Prof. Dr. Eurípedes Guilherme de Oliveira Nóbrega**

Curso: Engenharia Mecânica

Área de concentração: Mecânica dos Sólidos e Projeto Mecânico

Dissertação de mestrado acadêmico apresentada à comissão de Pós Graduação da Faculdade de Engenharia Mecânica, como requisito para a obtenção do título de Mestre em Engenharia Mecânica.

Campinas, 2006

S.P. - Brasil

IMADA
EX
BC/ 69.120
16.123.06
D X
110
29/06/06

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

Am68d Amorim, Maurício José
Desenvolvimento de bancada didático-experimental de baixo custo para aplicações em controle ativo de vibrações / Maurício José Amorim. --Campinas, SP: [s.n.], 2006.

Orientador: Eurípedes Guilherme de Oliveira Nóbrega

Dissertação (mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Mecânica.

1. Processamento de sinais – Técnicas digitais. 2. Controle em tempo real. 3. Programação em tempo-real. 4. Eletrônica. 5. Engenharia de estruturas – Controle. I. Nóbrega, Eurípedes Guilherme de Oliveira. II. Universidade Estadual de Campinas. Faculdade de Engenharia Mecânica. III. Título.

Título em Inglês: Design of a didactic and experimental testbed of low cost for applications in active control of vibration

Palavras-chave em Inglês: Embedded system, DSP system, Didactic testbed, Real-time programming, Fundamentals of electronics, Mechanical system controls

Área de concentração: Mecânica dos Sólidos e Projeto Mecânico

Titulação: Mestre em Engenharia Mecânica

Banca examinadora: Luiz Otávio Saraiva Ferreira e Elnatan Chagas Ferreira

Data da defesa: 21/02/2006

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA
DEPARTAMENTO DE MECÂNICA COMPUTACIONAL

DISSERTAÇÃO DE MESTRADO ACADÊMICO

**Desenvolvimento de Bancada Didático-
Experimental de Baixo Custo para Aplicações
em Controle Ativo de Vibrações**

Autor: Maurício José Amorim

Orientador: Prof. Dr. Eurípedes Guilherme de Oliveira Nóbrega

Prof. Dr. Eurípedes Guilherme de Oliveira Nóbrega, Presidente
DMC/FEM/UNICAMP

Prof. Dr. Luiz Otávio Saraiva Ferreira
DMC/FEM/UNICAMP

Prof. Dr. Elnatan Chagas Ferreira
DEMIC/FEEC/UNICAMP

Campinas, 21 de fevereiro de 2006.

Dedicatória

Aos meus amados pais, Paulinho e Odete que sempre estiveram ao meu lado transmitindo-me força, paz, palavras de incentivo, amor, alegrias e tão somente alegrias.

Agradecimentos

- A Deus por me fazer um amante da vida.
- Aos meus pais, Paulinho e Odete, grandes amigos. Pelo incentivo em todos os momentos da minha vida.
- Ao meu orientador, Eurípedes Nóbrega, que me mostrou os caminhos a serem seguidos.
- À CAPES pelo apoio financeiro ao desenvolvimento do trabalho.
- Aos meus irmãos e amigos, Alexandro, Juliano e Tiago, pelos bons momentos de convivência e aprendizado.
- A minha querida Lísia, amiga e companheira inseparável. Pela dedicação e inigualáveis momentos de alegria juntos passados.
- Meu especial agradecimento ao amigo Alysson pela amizade, pelo apoio incondicional durante o decorrer de todas as etapas deste trabalho, pelos ensinamentos, conselhos e pelo contagiante alto astral.
- Ao professor e amigo Belisário, que muito me ajudou nos primeiros passos da pesquisa.
- Ao professor Luiz Otávio pelas dicas, ensinamentos e pela oportunidade de trabalharmos juntos.
- Aos professores, membros da banca de qualificação, pelas sugestões a este trabalho.
- Ao professor Janito, pela considerável contribuição.
- Ao professor e amigo Alessandro, meu especial agradecimento pelo apoio e incentivo na conquista desse trabalho.
- Aos professores Deyller e João Carlos, grandes amigos e inspiradores, por fazerem parte da minha história.
- À todos os meus familiares, pela amizade e carinho que têm provocado grandes transformações na minha vida.
- Aos eternos amigos Juliano, André, Marcel e Roberto, pelo apoio e também momentos de diversão e descontração.
- Aos amigos da república Quilombo pelos momentos agradáveis.
- Aos amigos da Engenharia de Alimentos pelos momentos de irreverência compartilhados.
- Aos professores e funcionários do Departamento de Mecânica Computacional e da Coordenação de

Pós-graduação em Engenharia Mecânica da UNICAMP.

- A todos aqueles que contribuíram direta ou indiretamente para elaboração deste trabalho.

*“Viver é um desafio constante.
Sem perceber entramos nas ilusões da realidade superficial,
acreditando nas promessas fáceis de felicidade material e
mergulhamos assim em sonhos fúteis de conforto físico
imaginando estar imunes ao sofrimento.
Mas as ilusões obscurecem nossa lucidez, deturpam situações,
invertem valores,
nos arrastando aos círculos de sofrimento que queríamos evitar.
E chega sempre a hora de descobrir que a vida
é mais do que supúnhamos,
sua sabedoria nos coloca diante da eterna chama da verdade,
que ofusca nossas ilusões, nos forçando a distinguir
o falso do verdadeiro.
Por entre choques da realidade profunda e as lições do dia-a-dia
conquistamos as vantagens da maturidade e entendemos que
TUDO vale a pena.”*

Luiz Gasparetto

Resumo

Amorim, Maurício José. *Desenvolvimento de Bancada Didático-Experimental de Baixo Custo para Aplicações em Controle Ativo de Vibrações*. Campinas: Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 2006, 125 p. Dissertação (Mestrado).

Este trabalho apresenta uma bancada didática destinada ao ensino nos cursos de engenharia de aplicações em projetos de controle, processamento digital de sinais e programação em tempo real. Bancadas didáticas são ferramentas indispensáveis ao ensino, considerando que os conceitos vistos apenas em sala de aula são muitas vezes abstratos. A bancada em questão foi desenvolvida partindo de projeto mecânico já existente, tendo sido colocados na estrutura sensores extensométricos para obter a resposta do sistema como deformação e projetado o circuito condicionador do sinal para essa resposta. Para a primeira fase, envolvendo identificação, controle da estrutura e análise dos resultados, foram projetados dois acionadores para os motores que aplicam o distúrbio e o esforço de controle sobre a planta. Após terem sido satisfeitas as etapas da primeira fase, o próximo passo foi transferir o sistema de controle para uma configuração embarcada utilizando um microcontrolador. Para tanto, algumas adaptações e novos projetos emergiram diante da mudança do ambiente de processamento. A necessidade de adaptar os sinais para a nova placa de aquisição exigiu mudança em alguns circuitos. O acionador de motor para uma saída de controle modulada em largura de pulso foi desenvolvido utilizando componentes discretos. Além disso, é detalhadamente abordada a programação em tempo real do sistema de controle em questão.

Palavras chaves: Sistema embarcado, sistema DSP, bancada didática, programação em tempo real, eletrônica de potência, controle de sistema mecânico

Abstract

Amorim, Maurício José. *Design of a didactic and experimental testbed of low cost for applications in active control of vibration*. Campinas: Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 2006, 125 p. Dissertation (Master's Degree).

This work presents the design of a didactic testbed intended to teaching control system design, digital processing of signals and real-time programming. Didactic testbeds are very valuable tools when applying concepts developed inside the classroom. The testbed developed here is built upon a previous mechanical design. Strain gages were used as measuring devices using a conditioning circuit. Firstly, system identification, structural control and analysis of the results were proceeded, two motor drives were designed to operate motors performing both disturbance and control inputs. Afterwards, the control system was redesigned to operate using a microcontroller unit in a embedded architecture. Some adaptations on the circuits were needed and are covered here. A DC motor circuit driver using PWM signal was developed using discrete components. Besides, real-time programming of the control system is covered in detail.

Keywords: Embedded system, DSP system, didactic testbed, real-time programming, power electronics, mechanical system control

Sumário

1	Introdução	1
1.1	Relevância e motivação	1
1.2	Organização do texto	4
2	Bancadas para controle de estrutura	7
2.1	Bancadas de teste	7
2.2	Bancadas de teste para controle ativo	8
2.3	Controle ativo	12
2.4	Microcontroladores e ferramentas de programação	13
3	Descrição da Bancada	17
3.1	Estrutura mecânica	17
3.2	Sistema DSP	21
3.2.1	Descrição da placa	21
3.2.2	Descrição do programa	23
3.3	Processamento analógico de sinais	24
3.3.1	Sensores	24
3.3.2	Atuadores	27
3.4	Sistema embarcado	32
3.4.1	Descrição da placa	32
3.4.2	Descrição do programa	34
4	Programação do sistema embarcado	37
4.1	Acionamento com sinal em modulação por largura de pulso	37
4.1.1	Divisão do problema	38
4.1.2	Máquinas de estados e diagramas SDL	39

4.1.3	Fluxogramas	45
5	Modelos Analítico e Experimental	55
5.1	Modelo analítico da bancada	55
5.2	Identificação da bancada	59
5.2.1	Sinal de Schroeder	60
5.2.2	Identificação para o sistema DSP	62
5.2.3	Identificação para o sistema embarcado	66
5.3	Controladores H_2 e H_∞	69
5.4	Controle usando o sistema DSP	70
5.4.1	Implementação dos controladores	70
5.4.2	Resultados	72
5.5	Controle usando o sistema embarcado	74
5.5.1	Implementação dos controladores	74
5.5.2	Resultados	75
5.6	Resultados utilizando o sistema DSP para produzir o sinal em PWM	76
6	Conclusão e propostas para trabalhos futuros	79
6.1	Conclusão	79
6.2	Propostas para trabalhos futuros	80
A	Programa para teste do sistema embarcado	87
B	Código para projetar os controladores usando Matlab	99
C	Comunicação serial do sistema embarcado	103
D	Linguagem Gráfica SDL	107

Lista de Figuras

1.1	Diagrama de blocos dos sinais do sistema	3
2.1	Desenho da Ponte Bill Emerson	9
2.2	Bancada de teste com três graus de liberdade e atuador hidráulico	10
2.3	Estrutura com 3 graus de liberdade com atuação hidráulica usando tendões	11
2.4	Montagem de baixo custo utilizando o Kit LEGO aplicada ao ensino	12
3.1	Modelo geral da bancada	18
3.2	Esquema da dinâmica da bancada	20
3.3	Esquema da dinâmica da bancada com a massa para controle ativo	20
3.4	Foto da bancada	21
3.5	Diagrama de blocos do kit DS1102	22
3.6	Fluxo de informação na montagem do sistema DSP	23
3.7	Circuito auxiliar que fornece a deformação da estrutura em níveis de tensão	25
3.8	Filtro antialiasing para o sinal obtido pelos extensômetros	26
3.9	Diagramas de Bode do filtro projetado	26
3.10	Circuito de acionamento do motor da base com realimentação em tensão	27
3.11	Circuito de acionamento do motor de compensação com realimentação em corrente	28
3.12	Diagrama funcional do LMD18200T	29
3.13	Circuito para isolar o sinal em PWM entre acionamento e controle	30
3.14	Circuito para isolar o sinal de direção entre acionamento e controle	30
3.15	Circuito responsável pelo sinal em PWM a partir de componentes discretos	31
3.16	Diagrama em blocos do kit MSC1210EVM	33
3.17	Fluxo de informação na montagem com o sistema embarcado	34
4.1	Diagrama de partição.	38
4.2	Diagrama de partição.	39

4.3	Máquina de estados do módulo Conversor A/D	40
4.4	Diagrama SDL do módulo Conversor A/D	40
4.5	Máquina de estado do módulo Controle	41
4.6	SDL do módulo Controle	41
4.7	Máquina de estados do módulo Relógio amostragem	42
4.8	SDL do módulo Relógio amostragem	42
4.9	Máquina de estados do módulo Transmissão serial	43
4.10	SDL do módulo Transmissão serial	43
4.11	Máquina de estado do módulo Gerenciador	44
4.12	SDL do módulo de Gerenciador	44
4.13	Máquina de estado do módulo Sinal de controle	45
4.14	SDL do módulo Sinal de controle	45
4.15	Fluxograma do módulo Conversor A/D	48
4.16	Fluxograma do módulo Controle	49
4.17	Fluxograma do módulo Relógio amostragem	50
4.18	Fluxograma do módulo Transmissão serial	51
4.19	Fluxograma do módulo Gerenciador	52
4.20	Fluxograma do módulo Sinal de controle	53
5.1	Esquema dos elementos modelados na dinâmica do sistema	56
5.2	Comparação entre as FRFs experimental e a do modelo analítico com o coeficiente de amortecimento do motor ajustado	58
5.3	Sinal de Schroeder com espectro até 6,25 Hz	61
5.4	Módulo da FFT do sinal de Schroeder com espectro até 6,25 Hz	61
5.5	Resposta do sistema ao sinal de Schroeder aplicado ao motor da base	62
5.6	Módulo da FFT da resposta do sistema ao sinal de Schroeder aplicado ao motor da base	63
5.7	Resposta do sistema ao sinal de Schroeder aplicado ao motor superior	63
5.8	Módulo da FFT da resposta do sistema ao sinal de Schroeder aplicado ao motor superior	64
5.9	Diagrama de Simulink para identificar o sistema	65
5.10	Diagrama de valor singular do modelo identificado para o sistema	66
5.11	Diagrama de Simulink para identificação com o sinal em PWM	67

5.12	Resposta do sistema ao sinal de Schroeder como distúrbio no motor superior usando as características do sistema embarcado	68
5.13	Módulo da FFT da resposta do sistema ao sinal de Schroeder como distúrbio no motor superior usando as características do sistema embarcado	69
5.14	Programa utilizando Simulink	71
5.15	Resposta no tempo com controlador H_2	72
5.16	Resposta no tempo com controlador H_∞	73
5.17	FFTs do sistema e das versões com controlador, respectivamente	74
5.18	FFTs do sistema e da versão com controlador na placa MSC1210, respectivamente . . .	76
5.19	FFTs do sistema e da versão com controlador usando o sistema DSP, respectivamente .	77
5.20	Diagrama de Simulink para controle com o sistema DSP usando sinal de controle em PWM para o circuito de acionamento	78
D.1	Decomposição hierárquica de um sistema na linguagem SDL	108
D.2	Representação de um processo numa especificação SDL	109
D.3	Definição dos principais símbolos da linguagem SDL	110

Lista de Tabelas

3.1	Descrição de componentes mecânicos	19
3.2	Descrição Componentes Elétricos	19
5.1	Parâmetros do modelo analítico	59

Capítulo 1

Introdução

1.1 Relevância e motivação

As bancadas experimentais são dispositivos usados didaticamente para avaliar conceitos e validar modelos teóricos. O uso de bancadas experimentais simulando a operação de sistemas reais é também um método amplamente conhecido e extensivamente usado para o desenvolvimento de projetos em geral. A engenharia sempre necessitou testar conceitos e aplicações em escala reduzida nas mais diversas áreas. Com a evolução da eletrônica e, portanto, da instrumentação necessária aos laboratórios, o uso de sistemas experimentais para simulação e teste tornou-se muito mais simples, uma vez que sistemas de aquisição de dados e microprocessadores podem ser usados (Lee et al., 1997). Em conseqüência, a importância da implementação de bancadas de teste cresceu significativamente, tornando-se um apoio essencial para atividades de projeto, ensino e pesquisa. Atualmente, tanto aplicações de engenharia que envolvam eletrônica embarcada e análise de sinais, quanto atividades de ensino fazem uso de bancadas de teste como ferramentas auxiliares. Tais bancadas implementam sistemas bastante completos, apresentando assim, praticamente todas as características técnicas que seriam encontradas no sistema real, devido à instrumentação e ao aparato de tratamento de sinais necessário (Brglez et al., 1989).

Grandes esforços vêm sendo realizados por universidades e empresas de diversos países no sentido de desenvolver bancadas didáticas experimentais para sua implantação no currículo de vários cursos. Elas propiciam aos alunos a ilustração de conceitos importantes, facilitando a compreensão dos modelos matemáticos necessários para a análise dos sistemas, acarretando de modo geral maior eficiência no processo de aprendizado. Desse modo, a motivação dos estudantes aumenta quando têm a oportunidade de aplicar conceitos abstratos em meios concretos, tais como sistemas mecânicos e códigos de programação para a solução de problemas vistos em sala de aula, (Klassner and Anderson, 2003). A utilização de modelos estruturais reduzidos sob a forma de equipamentos didáticos estimula o interesse dos alunos, envolvendo-os de forma ativa no processo de aprendizagem e estabelecendo a

ligação fundamental entre teoria e prática.

O desenvolvimento de equipamentos didáticos de baixo custo, compatíveis com a capacidade de investimento atual das universidades, porém com ênfase na qualidade didática, é uma alternativa viável para se minimizar a carência existente nos atuais cursos de engenharia dos países em desenvolvimento. Bancadas didáticas experimentais são equipamentos indispensáveis em atividades educacionais relativas ao estudo de todas as engenharias e apenas com o aumento significativo da relação benefício/custo será possível atender à demanda.

A bancada experimental, objeto desse trabalho, vem sendo desenvolvida para o atendimento às disciplinas da área de controle de sistemas mecânicos. Trata-se de uma estrutura mecânica flexível que permite a simulação experimental de vários sistemas. Para o caso presente, o sistema de referência é o controle de um edifício de um andar. A estrutura permite assim aplicar um deslocamento à base do edifício e controlando uma massa posicionada em seu topo, minimizar a vibração resultante.

Frente ao alto custo na compra dos equipamentos disponíveis no mercado internacional, a bancada didática em questão foi concebida e construída dentro da filosofia de se dispor de um equipamento versátil em termos de operação, facilidade de instalação e transporte, bem como a melhor relação possível entre benefício e custo.

Nesse contexto, o presente trabalho propõe o desenvolvimento de uma bancada didático-experimental de baixo custo para auxílio no ensino e fundamentação dos conceitos de controle ativo estrutural. Como ilustração da usabilidade e teste do sistema, apresenta também resultados obtidos com a implementação dos controladores H_2 e H_∞ , de grande importância na área de controle estrutural. Para tanto, faz uso, primeiramente, de um sistema com processador digital de sinais (DSP, do inglês “digital signal processor”) e em seguida, de um microcontrolador. Ambos os sistemas são detalhados mais adiante.

Um dos objetivos deste trabalho consiste em desenvolver a parte eletrônica responsável pela interação da estrutura mecânica, ora com a placa do processador digital de sinais, ora com o microcontrolador. O teste e a apresentação dos resultados também é mérito deste trabalho.

Assim como no equipamento didático apresentado em (Heck et al., 2004), a estrutura aqui desenvolvida pode ser usada pelo corpo docente para ensinar conceitos de controle ativo, processamento digital de sinais, programação em linguagem C ou linguagem de montagem (assembly), sistemas com processadores embarcados, e outros mais. O baixo custo e a portabilidade da montagem experimental fazem-na uma boa opção para uso em cursos correlatos. Além das vantagens de utilização apresentadas acima, a montagem eletrônica da bancada proposta neste trabalho permite sua utilização para

outras configurações mecânicas, por exemplo, como um pêndulo invertido ou controle de robôs espaciais flexíveis.

A bancada desenvolvida neste trabalho foi projetada para exercitar a modelagem e o controle ativo de vibrações de estruturas mecânicas flexíveis. Para o teste da bancada foram implementados dois métodos de controle ativo:

- baseado em um DSP;
- baseado em microcontrolador.

O sistema baseado na DSP, referenciado no restante do texto como *sistema DSP*, faz uso do kit modelo DS1102, baseado na terceira geração de processadores digitais de sinais da Texas Instruments (*TMS320C31*) de 40 MFlops e 60 MHz. O custo de uma versão atualizada desse equipamento é da ordem de €4000,00.

O sistema baseado em microcontrolador, referenciado no restante do texto como *sistema embarcado*, faz uso do kit modelo MSC1210 produzido pela Texas Instruments (*TI*), baseado no microcontrolador 8051 com cristal interno de 11,059 MHz. O custo de uma versão desse equipamento é da ordem de algumas dezenas de dólares.

O sistema, para ambos os casos, pode ser representado pelo diagrama de blocos da Figura 1.1.

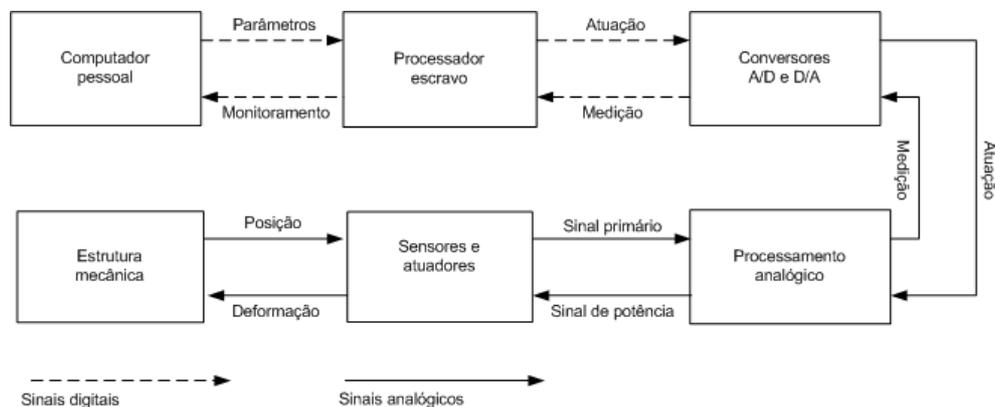


Figura 1.1: Diagrama de blocos dos sinais do sistema

O módulo Computador pessoal é utilizado para programar os sistemas DSP e o embarcado, bem como a monitoração da planta real. Já o Processador escravo processa em tempo real o sinal de controle, além de estabelecer comunicação para monitoramento pelo módulo anterior. O bloco representado por Conversores A/D e D/A faz a ligação do meio físico com o digital, ou seja, converte o sinal analógico vindo da estrutura mecânica para o processamento digital do sinal de controle e o envio desse sinal

processado novamente para a estrutura como esforço de controle. O módulo Processamento analógico, é responsável por acondicionar os sinais vindos do bloco anterior, produzindo um sinal de potência para mover os motores, e também por tornar possível a leitura da resposta do sistema ao distúrbio. O módulo Sensores e atuadores representa os sensores e os motores propriamente ditos, fixados à estrutura metálica.

Os kits DS1102 e MSC1210 compreendem os módulos Processador escravo e Conversores A/D e D/A e são portanto adquiridos no mercado, bem como o módulo Computador pessoal, é claro. Os demais módulos foram projetados e prototipados, com exceção do Estrutura mecânica que foi objeto de estudo de trabalho de graduação (Amorim, 2005).

Nota-se, assim, que a principal redução de custos está representada na substituição do sistema DSP pelo sistema embarcado. No entanto, o projeto dos demais módulos, incluindo aqui a estrutura mecânica correspondente, já representa uma redução de custos significativa, em comparação às bancadas didáticas comerciais.

Todas as etapas da implementação de um sistema de controle ativo são abordadas, do ponto de vista de utilização da bancada como instrumento de simulação e teste, como base para utilizações futuras em laboratórios didáticos da área de controle de sistemas mecânicos ou de desenvolvimento de sistemas com processamento embarcado em tempo real.

1.2 Organização do texto

O Capítulo 2 compreende uma revisão sobre bancadas experimentais com a apresentação de alguns trabalhos significativos, uma breve revisão de métodos de controle ativo e uma abordagem do micro-controlador e demais ferramentas pertinentes ao desenvolvimento do trabalho.

O Capítulo 3 descreve a estrutura mecânica utilizada para o trabalho. Trata também dos sensores e circuitos eletrônicos desenvolvidos, bem como a descrição dos programas utilizando os sistemas DSP e embarcado. A organização da bancada experimental, no que diz respeito ao fluxo de sinais e informações, é abordada aqui.

O Capítulo 4 apresenta as etapas seguidas para implementação do programa de controle utilizando técnicas de programação em tempo real e fazendo uso da linguagem C como ferramenta de desenvolvimento.

O Capítulo 5 mostra duas formas de modelagem para a bancada: analítica e experimental. O modelo experimental é o utilizado para se calcular os controladores. Os resultados obtidos no sistema DSP e no sistema embarcado são comparados.

Finalmente, o Capítulo 6 contém as conclusões e sugestões para trabalhos futuros.

Capítulo 2

Bancadas para controle de estrutura

Este capítulo está dividido em quatro seções: na primeira, uma visão global sobre bancadas experimentais onde seus conceitos e importância são apresentados; na segunda, algumas bancadas experimentais que influenciaram este trabalho, são referenciadas; na terceira seção, aborda-se brevemente a teoria para o projeto dos controladores usados no trabalho; e, por fim, na última seção, uma revisão sucinta sobre microcontroladores e ferramentas pertinentes ao desenvolvimento deste trabalho.

2.1 Bancadas de teste

Na literatura da língua inglesa encontram-se palavras sinônimas para bancada de teste (Encarta, 2005), *benchmark* e *testbed*, cujos respectivos significados são descritos: usado como padrão para medir ou avaliar alguma coisa; área equipada para testar maquinário sob circunstâncias tão próximas das condições reais de operação quanto possível.

Este trabalho se concentra no desenvolvimento de uma bancada de propósitos principalmente didáticos, porém que permite ainda seu uso em pesquisa para o teste de controladores. Por se tratar de uma estrutura mecânica flexível, o trabalho envolve a área de controle ativo de vibrações.

Nas últimas duas décadas, o controle ativo de estruturas mecânicas tornou-se uma tecnologia viável para aumentar a funcionalidade estrutural e a segurança contra perturbações externas. O sucesso do Primeiro Congresso Mundial em Controle Estrutural, que aconteceu em Pasadena, California, em agosto de 1994, demonstrou o interesse em controle de estruturas mecânicas. A Segunda Conferência em Controle Estrutural, aconteceu em Kyoto, Japão no verão de 1998, mostrando a continuidade dos estudos nesta área (Spencer et al., 1998b).

A proteção de estruturas de engenharia civil, incluindo seu conteúdo material e ocupantes, é, sem dúvida, de grande importância atualmente. Tal proteção proporciona uma estrutura mais confiável e maior conforto para os habitantes. Exemplos de estruturas que investem em proteção incluem edifícios,

torres, estradas, pontes e oleodutos. Da mesma forma, podem-se citar eventos que provocam a necessidade de tal proteção, quais sejam: terremotos, vendavais, ondas marítimas, tráfico pesado e mesmo atos deliberados. Tudo indica que métodos de controle ativo darão uma contribuição significativa na área de proteção estrutural, de grande importância social e econômica (Spencer and Sain, 1997).

Para o desenvolvimento, teste e validação de sistemas de controle e para o gerenciamento de estruturas automatizadas, alguns laboratórios têm usado a opção de altíssimo custo quando escolhem o equipamento real como bancada para testes. Estes sistemas experimentais, além de caros, exigem um grande espaço físico e nem sempre permitem o teste nas condições adequadas. É neste cenário que surgem as bancadas experimentais em escala reduzida, reproduzindo as características de operação do equipamento real. Esse modelo reduzido tem muitas vantagens sobre os modelos tradicionais. Primeiro, eles são mais baratos e podem ser facilmente reproduzidos em outros laboratórios. Segundo, eles podem ser facilmente reconfigurados para considerar cenários alternativos. Terceiro, eles podem considerar um ambiente de emulação bem mais complexo que aqueles que usam o equipamento real. Finalmente, eles são totalmente confiáveis e seguros, exigindo um mínimo de habilidade para operá-los (Gonzalez et al., 2000).

Utilizando-se de modelos experimentais, os desenvolvedores de sistemas de controle podem testar seus programas nestes equipamentos automatizados em escala reduzida, antes de testá-los na planta real, (Gonzalez and Davis, 1997b; Gonzalez and Davis, 1997a). Esse método fornece uma vantagem sobre o método mais comum de testes realizado no ambiente de simulação matemática do computador, já que exige que o programa seja desenvolvido para atuar da mesma forma como será o caso quando o sistema de controle for implementado sobre o sistema real.

2.2 Bancadas de teste para controle ativo

É utilizado em (Lu et al., 2003) um modelo de referência, simulando uma construção real para comparação no estudo de estratégias de controle ativo estrutural. A referida estrutura é uma torre fina de concreto, sensível à ação do vento, proposta para a cidade de Melbourne, Austrália. As cargas de ventos que atuam no modelo são definidas pelos resultados dos testes de túneis de vento conduzidos na Universidade de Sydney. A partir daí é possível projetar um atuador para ser instalado no topo da construção de modo a reduzir a resposta da planta sob a ação de fortes ventos. O modelo usa um controlador ótimo LQG como padrão para avaliação de desempenho. O autor inclui restrições da norma H_∞ na metodologia de projeto H_2 e estuda como a inclusão dessas restrições melhorou o desempenho do controlador H_2 na estrutura em questão sob a excitação do vento.

Em (Caicedo et al., 2003) foi realizado um estudo de maneira a fornecer um modelo de avaliação para o desenvolvimento de estratégias de controle de pontes com cabo, conforme Figura 2.1.

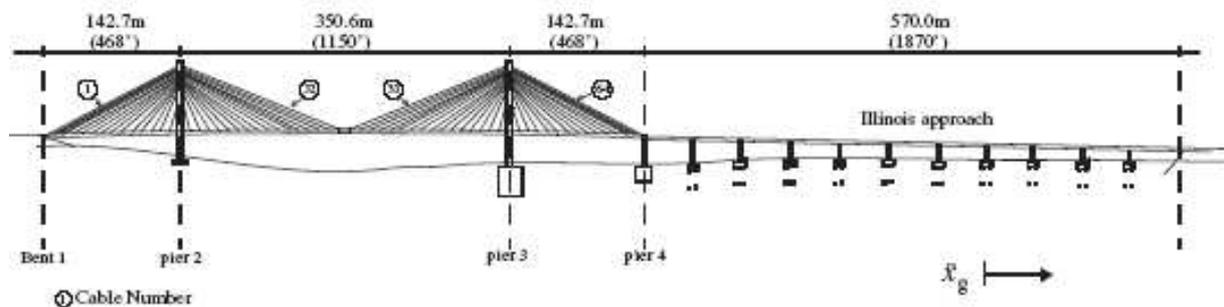


Figura 2.1: Desenho da Ponte Bill Emerson

Baseado no desenho detalhado da Ponte Bill Emerson que liga os extremos do Rio Mississippi, perto de Cape Girardeau, o modelo de avaliação foi desenvolvido para representar o comportamento complexo da estrutura levando em consideração, a princípio, cargas sísmicas de acordo com dados fornecidos pela *American Association of State Highway and Transportation Officials* (AASHTO). Quando os efeitos de temperatura foram considerados, surgiram limitações quanto à grandes níveis de tensões. Avaliando vários projetos, determinou-se que a incorporação de dispositivos de transferência de força resultaria em solução mais eficiente para a implementação. Segundo (Spencer et al., 1998b), controladores que são implementados com sucesso em modelos de avaliação são esperados que desempenhem similarmente na prática.

Em (Spencer et al., 1998b) é apresentado um modelo em controle estrutural que pode ser usado para avaliar a efetividade relativa e a implementabilidade de vários outros algoritmos de controle de estruturas e também uma bancada de teste para avaliação de projetos de controle de interesse tais como redução de ordem, interação estrutura-controlador, atraso computacional, entre outros. É considerada uma estrutura, conforme Figura 2.2, cujo modelo representa um prédio ativamente controlado, com três massas distribuídas representando andares do edifício.

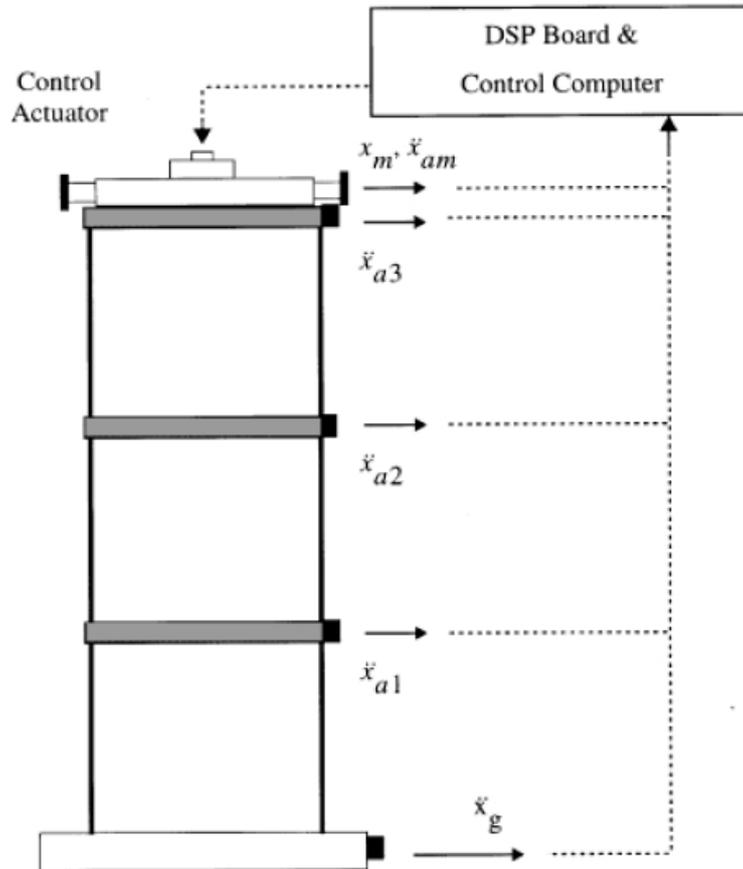


Figura 2.2: Bancada de teste com três graus de liberdade e atuador hidráulico

A armação da estrutura é formada por barras de aço, com 158 cm de altura e pesando 77 quilos, sendo que as massas dos pisos somam 227 quilos distribuídos igualmente entre eles. As três primeiras frequências naturais de interesse são 5,81, 17,68 e 28,53 Hz com respectivos fatores de amortecimento de 0,33, 0,23 e 0,30 por cento.

Para a proposta de controle da estrutura em questão foi implementado, no terceiro andar, um atuador dinâmico (AMD, do inglês “Active Mass Driver”). O sistema consiste de um atuador hidráulico com uma massa de aço acoplada ao eixo do cilindro. O peso total a ser deslocado pelo cilindro hidráulico é de 5,2 quilos, envolvendo a massa de aço, pistão e os discos de aço fixados a ele. A posição do atuador foi obtida usando um transformador diferencial variável linear (LVDT, do inglês “linear variable differential transformer”), rigidamente montado entre o fim do eixo do pistão e o terceiro piso. Para ilustrar restrições e desafios dessa bancada foi apresentado o projeto do controle LQG.

Em (Spencer et al., 1998a) é considerada uma estrutura conforme ilustra a Figura 2.3.

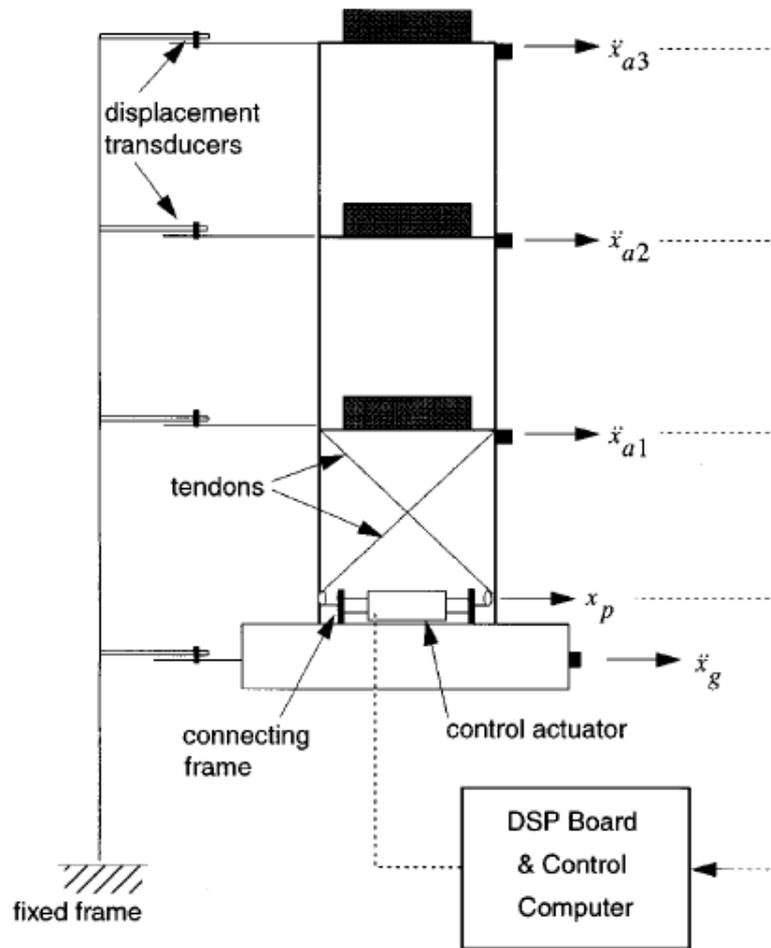


Figura 2.3: Estrutura com 3 graus de liberdade com atuação hidráulica usando tendões

A estrutura na qual o modelo de avaliação é baseada é a de um prédio, também de três andares. A estrutura de teste apresentada acima tem uma massa de 2950 quilos, distribuída entre os andares e uma altura de 254 cm. Um atuador hidráulico, quatro tendões pré-tensionados e um quadro de aço que conecta o atuador aos tendões, são usados para aplicar o esforço de controle à estrutura de teste. Os quatro tendões diagonais transmitem o esforço de controle ao primeiro piso da estrutura e o quadro de aço conecta o atuador aos tendões.

Em (Heck et al., 2004) é usada uma montagem experimental portátil e de baixo custo, comparado com outras disponíveis no mercado, baseada em um *Kit LEGO* e ilustrada na Figura 2.4. A montagem consiste de duas engrenagens, sensor de rotação acoplado no eixo do motor e o microcontrolador dedicado. No trabalho citado, a montagem foi utilizada no ensino de conceitos fundamentais de controle, processamento digital de sinais, programação C e processamento embarcado. O experimento reforça para os estudantes a importância de computação em tempo real, tarefas periódicas e o compromisso

envolvendo custo e capacidade do processador embarcado versus desempenho dentro do contexto de sistemas de controle. O baixo custo e a portabilidade da montagem experimental a torna bem indicada para o uso em cursos introdutórios em sinais e controle.



Figura 2.4: Montagem de baixo custo utilizando o Kit LEGO aplicada ao ensino

2.3 Controle ativo

O objetivo mais importante de um sistema de controle é alcançar certas especificações de desempenho além da condição básica de prover estabilidade interna (Zhou and Doyle, 1998). Como exemplo de técnicas de controle, pode-se citar inicialmente o *LQG* (Skogestad and Postlethwaite, 1997), um controlador ótimo desenvolvido a partir do trabalho de filtragem ótima de Wiener em 1940 e que alcançou maturidade na década de 1960. Segundo (Paganini, 1995), o controlador *LQG* apresenta bom desempenho desde que consiga modelar adequadamente a resposta de um sistema incluindo o ruído branco. É um modelo adequado para muitas perturbações que surgem em aplicações experimentais. O *LQG* assume que a dinâmica da planta pode ser modelada como linear e que o ruído dos sinais de medida e de atuação apresentam comportamento estocástico e com propriedades estatísticas conhecidas.

O desenvolvimento da teoria de controle ótimo H_2 a partir do *LQG* mencionado pode ser encontrado na maioria dos livros de controle avançado. Em anos recentes, problemas de controle ótimo H_2 têm sido considerados em um cenário diferente do tradicional *LQG*. O interesse nesse novo cenário é minimizar a norma H_2 da matriz de transferência a partir de um distúrbio externo para uma saída controlada de um dado sistema linear invariante no tempo através da seleção apropriada de um controlador (Chen et al., 1993).

Conceitos para controle ativo e híbrido ativo/passivo em estruturas tem sido muito explorados por autores da área. Atenção tem sido dada, mais recentemente, à aplicação da teoria de controle robusto

no contexto de projeto H_∞ . Controle robusto está envolvido com a manutenção do desempenho mediante incertezas na dinâmica do sistema. Incertezas são basicamente as discrepâncias entre o modelo matemático da planta a ser controlada e a planta real, como é o caso dos modos de vibração de altas frequências presentes na estrutura, porém descartados no modelo. Outro exemplo é a massa ou rigidez de alguns elementos da dinâmica do sistema, cujos valores apresentam sempre alguma diferença com relação àqueles adotados do modelo, a chamada incerteza paramétrica. Entradas exógenas ou distúrbios de origem externa também perturbam o sistema real com relação ao modelo matemático. Eles afetam o desempenho mas não a estabilidade do sistema de malha fechada. Para uma construção civil controlada ativamente, atividade sísmica, ventos e ruído de sensores são exemplos de distúrbios exógenos. O desempenho é medido, nos métodos que têm como base a norma H_∞ , pelo pior caso na resposta do sistema controlado aos distúrbios (Whorton et al., 1998).

Motivado pela deficiência de desempenho do controle LQG devido à sensibilidade à perturbações causadas pelas incertezas de modelagem, as quais afetam a estabilidade do sistema de malha fechada, houve, na década de 1980, um significativo deslocamento rumo aos métodos H_∞ para o controle robusto. Este desenvolvimento originou-se da influência do trabalho de Zames (1981) que avançou a teoria relacionada ao H_∞ no sentido do desempenho e da estabilidade (Skogestad and Postlethwaite, 1997). Conforme (Sarracini, 2004), o projeto H_∞ pode ser empregado quando se deseja que um sistema mantenha um determinado desempenho em condições adversas tais como erros de modelagem e distúrbios externos. Em termos gerais, espera-se, através de um método de otimização no domínio da frequência, reduzir os efeitos das incertezas, dos distúrbios externos e ruído de sensores tendo em vista o pior caso, ou seja, o valor de pico da função de resposta em frequência.

2.4 Microcontroladores e ferramentas de programação

A rápida redução no tamanho de circuitos integrados (CIs) tem permitido mais e mais funções serem implementadas em uma área muito pequena, ao ponto de sistemas inteiros poderem ser implementados em um único chip. Estes sistemas são referenciados pelos acrônimos em inglês *Systems on Silicon* (SOS), *Systems on a Chip* (SOC) ou *Systems in a Package* (SIAP) (Beetner et al., 2000).

O trabalho em questão utiliza o microcontrolador 8051. Ele foi escolhido por causa da sua boa capacidade computacional, arquitetura simples e ampla popularidade, além da probabilidade de continuar sendo um importante processador por vários anos. Existem vários fabricantes de componentes da família 8051, podendo-se citar Texas Instruments, Philips, Atmel, Dallas, Temic, Siemens, entre outros, bem como uma gama de livros atuais no mercado sobre esse controlador (“Programação C

para Microcontroladores 8051”, “Microcontrolador 8051 - Detalhado (5ª Edição)”, entre outros). Os fundamentos aprendidos nesse controlador podem ser facilmente transferidos para outros processadores ou arquiteturas mais sofisticadas.

Emprega-se na bancada a placa modelo MSC1210 fabricada pela Texas Instruments, contendo um microcontrolador da família 8051 com um conversor analógico digital de alta precisão. O microcontrolador é utilizado para processar digitalmente os sinais, substituindo assim, o sistema formado pelo kit DS1102, *Matlab*[®] e computador, sendo o responsável pela execução, em tempo real, do algoritmo de controle ativo da estrutura.

Uma vez obtidas as matrizes de estado identificadas do sistema, com o sinal de saída de deformação da estrutura usado como realimentação, implementaram-se no 8051 controladores do tipo H_2 e H_∞ .

Para a maioria de sistemas embarcados o esforço para desenvolver a programação é maior que o do projeto do circuito que utiliza o microcontrolador e conseqüentemente mais engenheiros são envolvidos. Geralmente, depurar uma única linha de código em *assembly* toma o mesmo tempo que várias páginas de código em linguagem de alto nível, por exemplo, o C. Como conseqüência, surge o interesse em utilizar linguagens de alto nível como meio para minimizar o trabalho no desenvolvimento de programas para sistemas embarcados. Isso é reforçado aqui pela necessidade de se usar ponto flutuante no processamento dos sinais.

Outra razão para usar uma linguagem de alto nível é o grau de independência arquitetural que ela fornece — dependendo do compilador, um código executável pode ser usado para qualquer microcontrolador que se deseje trabalhar, bastando apenas simples adaptações. Em algumas áreas onde laços de tempo são críticos, faz-se uso de rotinas em *assembly* para atender prioridades de tempo específicas do sistema. Apesar de arquitetura independente e portabilidade serem características desejadas, um programa de microcontrolador não pode ser totalmente separado das características específicas do hardware. Programar um microcontrolador, mesmo em uma linguagem de alto nível, requer acessibilidade a vários registros de periféricos, habilidade para reparar programas e dados, e ainda conhecimento do tipo de memória e sua organização, uso de estruturas de dados e de como lidar com interrupções. O compilador C para cada microcontrolador possui especificidades porque é preciso implementar os aspectos particulares dos periféricos disponíveis (Bannatyne and Viot, 1998).

O compilador C utilizado para desenvolver parte deste trabalho é o μ *Vision 2*[®] desenvolvido pela *Keil*[®]. Esta ferramenta é bem adaptada para encontrar problemas no código de alto nível, mas não para simular a interface entre hardware e software, (Beetner et al., 2000). Portanto, deve-se primeiramente depurar o programa e em seguida compilá-lo, para que seja gerado um arquivo executável em código

de máquina do microcontrolador e com isso, programar sua memória.

O microcontrolador utilizado possui 24 bits de resolução para conversão analógico-digital, sendo considerado um microcontrolador com conversor de alta precisão. Isso é mais do que suficiente para o trabalho desenvolvido aqui. Porém, esse dispositivo não possui saídas analógicas. Desse modo, será usado um sinal em PWM como saída para a atuação do controlador.

Outra interface desse equipamento utilizada aqui é a comunicação serial entre o computador de monitoramento e a placa MSC1210, para enviar os dados lidos no sensor, como uma forma de acompanhar os resultados do controle.

Capítulo 3

Descrição da Bancada

A descrição da estrutura em questão é apresentada detalhadamente nesse capítulo. Aqui também são descritos os circuitos eletrônicos desenvolvidos, a configuração do sistema DSP bem como a do sistema embarcado.

3.1 Estrutura mecânica

A bancada construída, conforme ilustração na Figura 3.1, pesa 16 quilos e mede 82,4 cm de altura. É constituída de três partes: a base; a estrutura propriamente dita representando o edifício; e, o subsistema de compensação. A base compreende um fuso acoplado à estrutura mecânica, movimentado por um motor de corrente contínua (CC) de modo a excitar o sistema. A estrutura é formada por uma massa sustentada por quatro colunas presas à base. E o subsistema de compensação por uma segunda massa acoplada ao sistema por intermédio de outro fuso, também movido por um motor de corrente contínua que atua em resposta ao sinal de controle calculado.

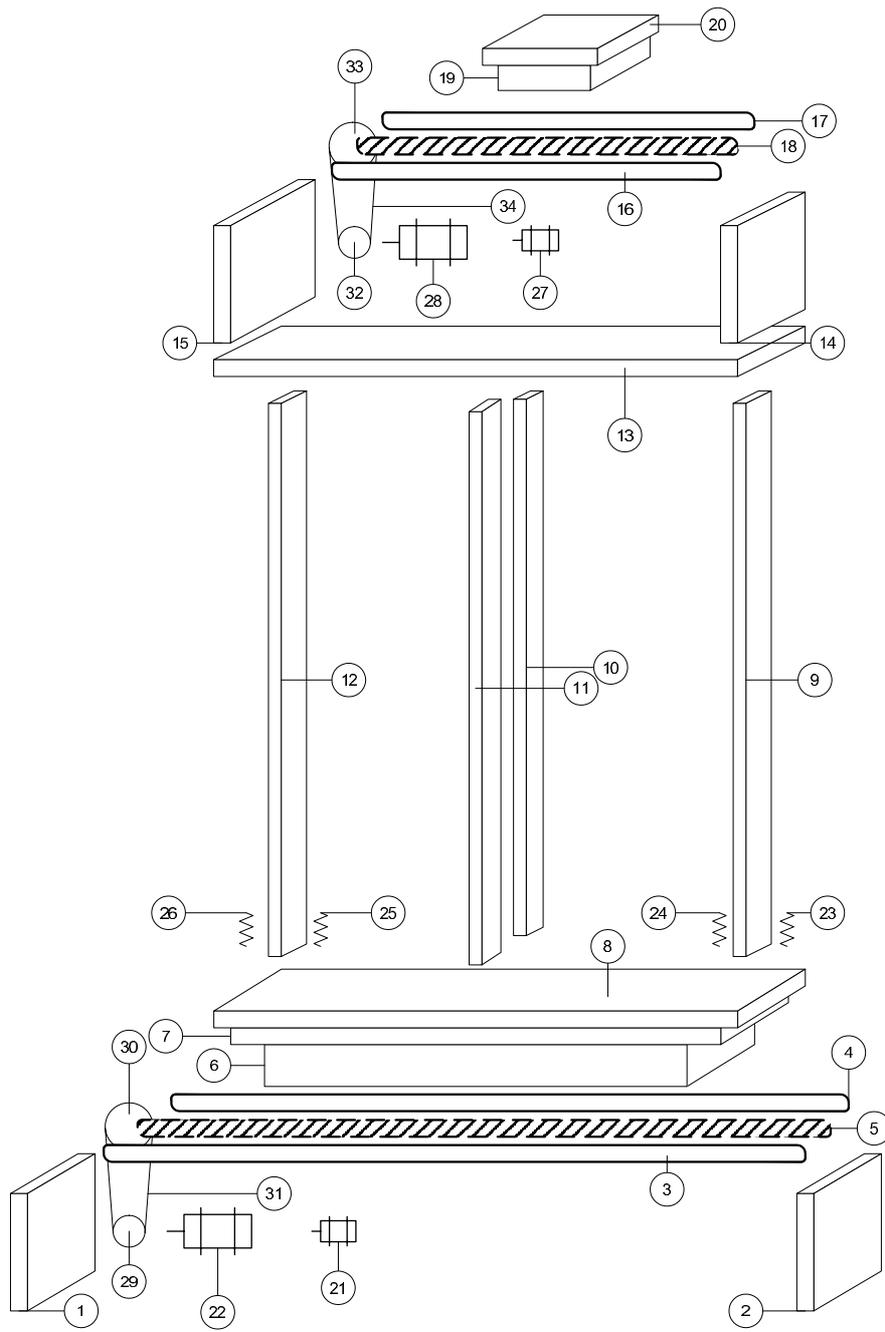


Figura 3.1: Modelo geral da bancada

As Tabelas 3.1 e 3.2 listam os componentes necessários à montagem completa da bancada.

Tabela 3.1: Descrição de componentes mecânicos

Numero ref.	Descrição
1, 2	Apoio lateral massa inferior
3, 4	Guia massa inferior
5	Fuso
6	Chapa dos rolamentos inferiores
7	Chapa integrante da massa inferior
8, 13	Chapa engaste das vigas
9, 10, 11, 12	Vigas
14, 15	Apoio lateral massa superior
16, 17	Guia massa superior
18	Fuso
19	Chapa dos rolamentos superiores
20	Massa superior
29, 32	Polia eixo motor
30, 33	Polia fuso
31, 34	Correia sincronizada

Tabela 3.2: Descrição Componentes Elétricos

Número ref.	Descrição
7, 25	Motor $12V_{cc}$, $35W$, $5500RPM$ MPA 110-5K5 - Motron
30, 31	Encoder $5v_{cc}$, $2500 Pulsos/Rev$ BDG 6460-1-05-2500-65 - BALLUFF
26, 27, 28, 29	Extensômetro $119,8 \pm 0,2 \Omega$, $K_e = 2,12 \pm 1,0 \%$ KFG-5-120-C1-11 - KYOWA

Como foi dito, quatro vigas paralelas são engastadas à base e a uma massa superior m_1 . Sobre m_1 está apoiado um motor CC que movimenta, analogamente à base, uma massa m_2 através de um fuso.

A estrutura é contruída mecanicamente para representar os principais elementos da dinâmica de um edifício de um andar. O modelo simplificado para esse problema mais conhecido é o de uma massa unida à base da estrutura por colunas, que atuam como molas de flexão. O modelo adotado consiste, portanto, em uma massa suspensa por quatro colunas, o qual tem sido utilizado em muitas aplicações, inclusive para simular respostas a terremotos (conforme (Spencer et al., 1998b)) e é representada pelo esquema da Figura 3.2.

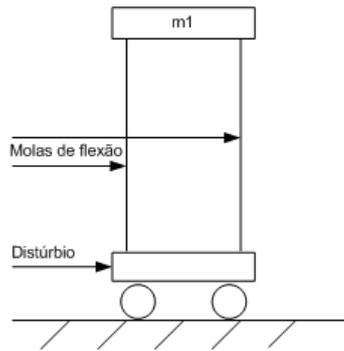


Figura 3.2: Esquema da dinâmica da bancada

A abordagem do controle ativo de vibrações consiste em acrescentar uma segunda massa (m_2) ao sistema que deve ser movida dinamicamente. A influência da inércia dessa massa e das forças dinâmicas que ela produz no restante da estrutura deve ser responsável pela atenuação da vibração. Um esquema possível para essa configuração está na Figura 3.3.

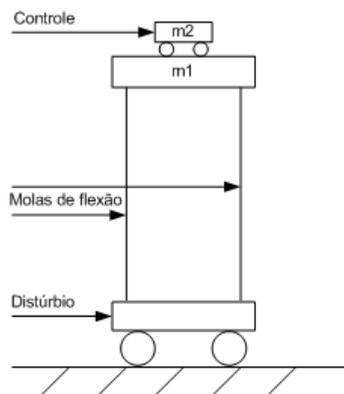


Figura 3.3: Esquema da dinâmica da bancada com a massa para controle ativo

Resumindo, o movimento da base representa um distúrbio que provoca a oscilação da massa superior m_1 que deve ser minimizada movimentando-se a massa m_2 .

A configuração geral da bancada é mostrada na Figura 3.4.

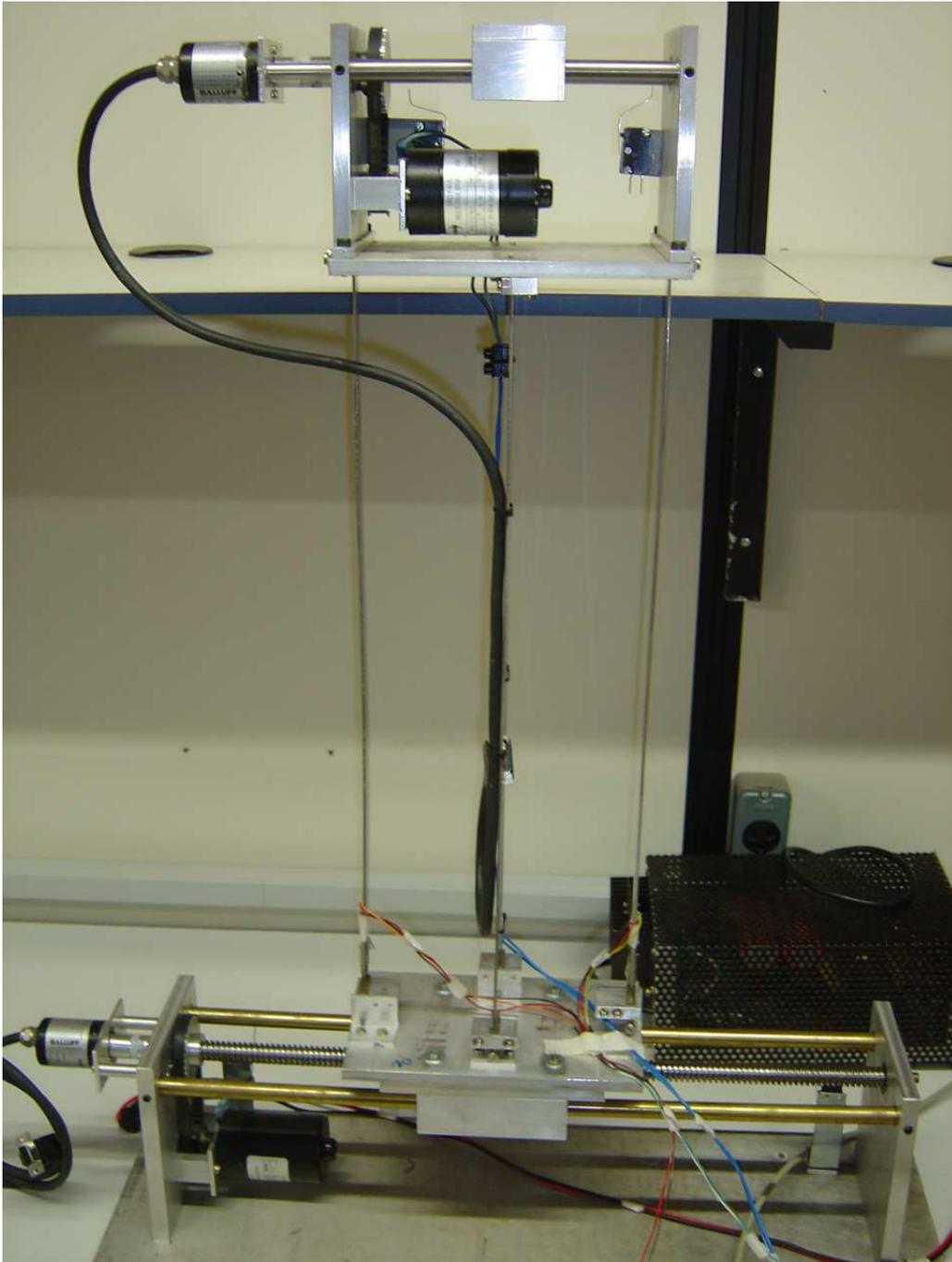


Figura 3.4: Foto da bancada

3.2 Sistema DSP

3.2.1 Descrição da placa

O kit produzido pela dSPACE compreende basicamente uma placa de interface para microcomputadores pessoais a qual possui interface serial, 4 canais A/D sendo dois de 16 bits e dois de 12 bits, 4

canais D/A de 12 bits, 2 entradas incrementais para encoders, 26 canais de entradas e saídas digitais, memória RAM de 128K e processador de 60 MHz.

O sinal lido dos extensômetros e o enviado para os circuitos de acionamento dos motores CC são tratados pelo sistema DSP. Trata-se de um sistema de aquisição acoplado a um computador pessoal utilizando uma placa que contém canais de conversão de sinal A/D e D/A fabricada pela dSPACE (Figura 3.5). Essa placa é comprada conjuntamente com o software Control Desk destinado a operar os parâmetros de aquisição em tempo real. Também estão incluídos módulos de comunicação com o software de simulação Simulink, responsável pela sua programação. É necessário lembrar que a placa de aquisição possui processador próprio, constituindo um sistema de processamento digital de sinais.

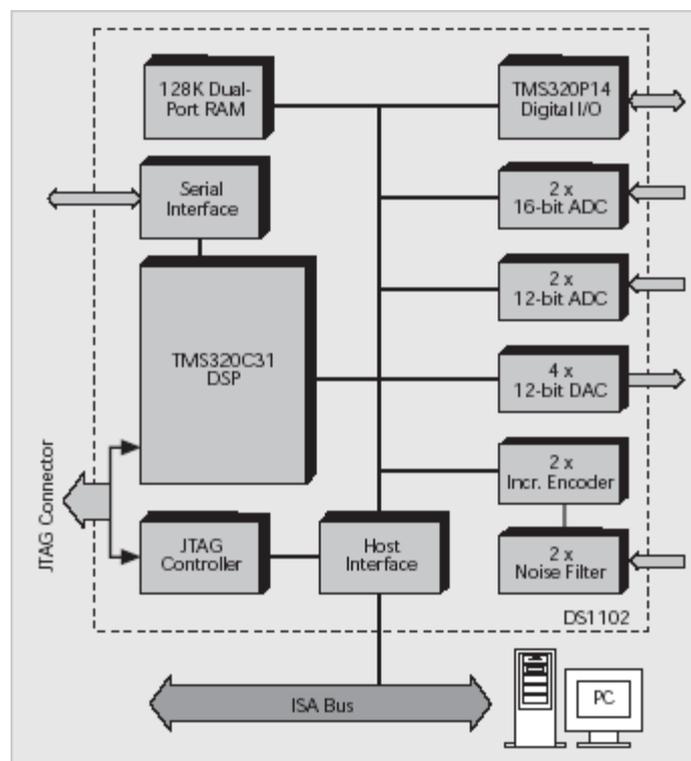


Figura 3.5: Diagrama de blocos do kit DS1102

A operação do sistema é controlada através da programação gerada automaticamente a partir de comandos no Simulink, que transforma o diagrama de blocos respectivo em um programa em C, e o descarrega, após compilado, na memória de programa da placa. O controlador, assim programado, realiza em tempo real a conversão A/D, o cálculo do sinal de controle e a conversão D/A.

Com essas configurações a montagem utilizando o sistema DSP pode ser resumida como segue:

- os extensômetros produzem um sinal de tensão associado à deformação;

- o sinal passa por um amplificador;
- o sinal amplificado é lido pelo conversor analógico-digital;
- o processador da placa dSPACE realiza o processamento do programa feito via Simulink;
- os canais de conversão digital-analógico enviam sinais para os circuitos de acionamento de ambos os motores: do sinal de distúrbio suposto desconhecido e do sinal de controle;
- os circuitos de acionamento alimentam os motores fornecendo a potência para que se movam.

A Figura 3.6 mostra a disposição geral dos equipamentos e o fluxo de informações no sistema DSP.

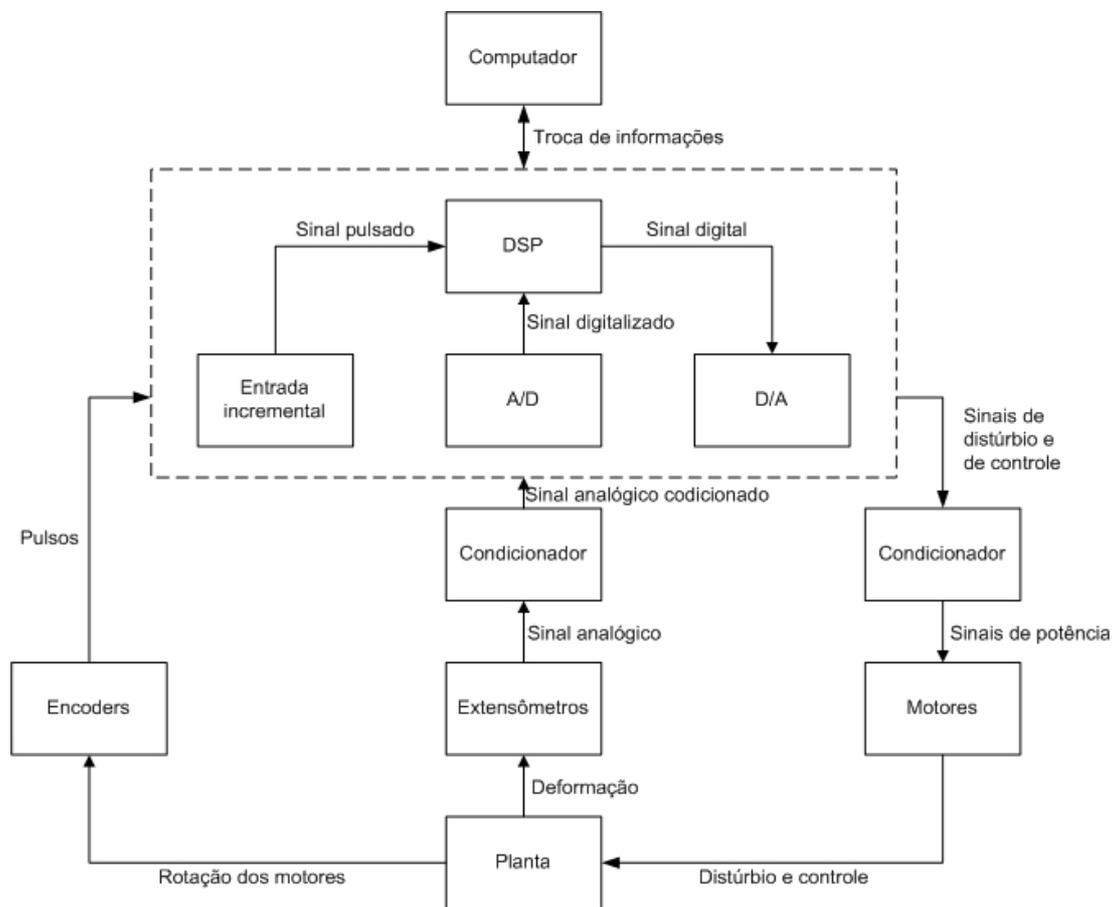


Figura 3.6: Fluxo de informação na montagem do sistema DSP

3.2.2 Descrição do programa

O Simulink foi utilizado para a identificação experimental e posteriormente a implementação do algoritmo de controle. Os diagramas desenvolvidos são descritos no Capítulo 5.

No Simulink foram programadas configurações diferentes. Para identificar o sistema, inicialmente, usando o sistema DSP, sinais de excitação devem ser produzidos para as duas entradas do sistema, que serão utilizadas como distúrbio e como controle.

Para tanto, blocos que produzem a excitação foram ligados aos módulos de conversão D/A e faz-se a leitura dos extensômetros pelo conversor A/D.

Acrescentar controle a essa configuração corresponde a alterar o digrama de blocos de Simulink acrescentando uma realimentação entre a saída medida dos extensômetros e a entrada escolhida para controle intermediada pelo controlador.

O passo seguinte foi testar o controle pelo microcontrolador utilizando o sistema DSP, ou seja, produzir neste o sinal de controle já em PWM para acionar o motor de controle utilizando o circuito de acionamento com ponte H descrito neste capítulo.

Para isso, um diagrama de blocos que converte um sinal em uma versão de amplitude e período constante com largura de pulso ajustável foi elaborado. Esse diagrama deve produzir também o sinal de direção, correspondente ao sinal aritmético cujo módulo foi convertido em PWM.

Esses dois sinais produzidos então no Simulink (sinal em PWM e sinal de direção) são obtidos pela bancada nos canais de saída D/A.

3.3 Processamento analógico de sinais

3.3.1 Sensores

Assim como na simulação em escala reduzida de um terremoto, a excitação é fornecida pelo motor que aciona o fuso da base, produzindo um deslocamento (no caso da bancada, determinado) para excitar a estrutura. Utilizando-se extensômetros (Tabela 3.2) ligados em ponte, colocados próximos ao engaste inferior de duas das vigas, em ambas as faces, é possível medir a variável de interesse do problema (deslocamento da massa superior m_1) através de um sinal que representa a deformação (ε). O arranjo em ponte é o mais recomendado para ser utilizado na construção de transdutores a base de extensometria, tais como células de carga e transdutores de deslocamentos. Para detalhes sobre ligação de extensômetros, ver (Almeida, 1996).

Fazendo uso da ligação em ponte dos extensômetros e seu coeficiente de sensibilidade (K), tem-se (Almeida, 1996):

$$\Delta E_0 = KE\varepsilon. \quad (3.1)$$

Dessa relação pode-se obter a deformação real (ε) a partir da tensão medida (ΔE_0) e da tensão de alimentação (E).

Para condicionar o sinal da ponte de extensômetros de modo a ser lido pela placa de aquisição de dados foi utilizado um circuito integrado da *Analog Devices*, modelo AD524 (Devices, 1999). Esse dispositivo é um amplificador de instrumentação que pode operar com três ganhos preestabelecidos (10, 100 e 1000), mas que também oferece flexibilidade para obter ganhos específicos a partir de arranjos de resistores. O amplificador implementado para fornecer a deformação da estrutura é apresentado na Figura 3.7. A configuração mostrada opera com ganho 100.

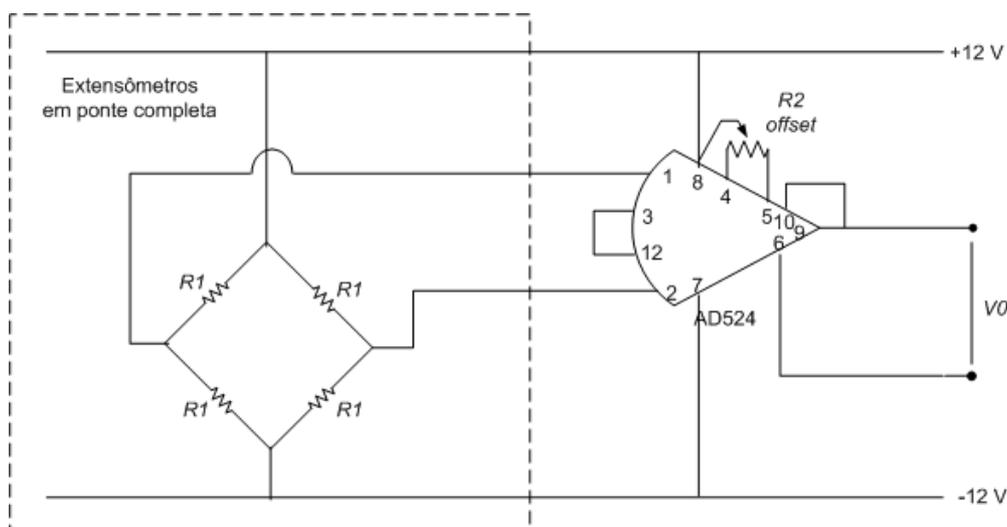


Figura 3.7: Circuito auxiliar que fornece a deformação da estrutura em níveis de tensão

Para medir a rotação dos motores, são utilizados encoders (Tabela 3.2). O encoder é um sensor óptico bastante conhecido e não será abordado aqui. Sua resposta é incremental e é tratada pela placa de aquisição DSP de modo a produzir a posição angular do motor. Porém, esses transdutores não fazem parte da configuração final de controle, uma vez que a variável medida é a deformação nos extensômetros. O sinal medido da rotação do motor foi utilizado para sua identificação e também para verificação do comportamento dinâmico do motor e do sistema. Em um modelo completo, como será visto, pode-se incluir a dinâmica do motor como parte da planta a ser controlada.

Para o sistema utilizando o microcontrolador, o canal A/D do sistema DSP é substituída pelo módulo A/D do sistema embarcado, utilizando uma taxa de amostragem de 25 ms. A precisão do conversor é maior nesse caso, de 24 bits, contra os 16 da placa da dSPACE. O sinal, amplificado pelo AD524, é filtrado por um filtro passa-baixa, de terceira ordem, e em seguida conectado a um par de entradas analógicas diferenciais da placa MSC1210 que constitui a interface do microcontrolador 8051.

O filtro analógico passa-baixa utilizado foi montado com componentes discretos (resistores e capacitores) na configuração de um filtro Pi, quatro resistores e três capacitores iguais, de valores, respectivamente 255 k Ω e 100 nF. O circuito do filtro é mostrado na Figura 3.8.

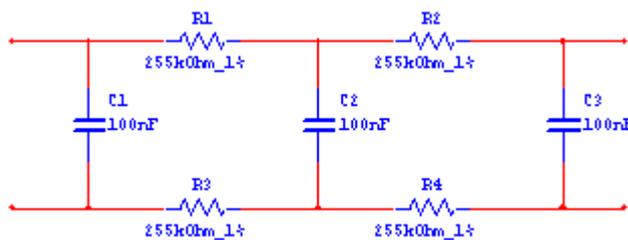


Figura 3.8: Filtro antialiasing para o sinal obtido pelos extensômetros

A frequência natural para o qual o filtro foi projetado, é dada por

$$F = \frac{1}{2\pi RC} = 6,241 \text{ Hz.}$$

Os diagramas de Bode do filtro é mostrado na Figura 3.9.

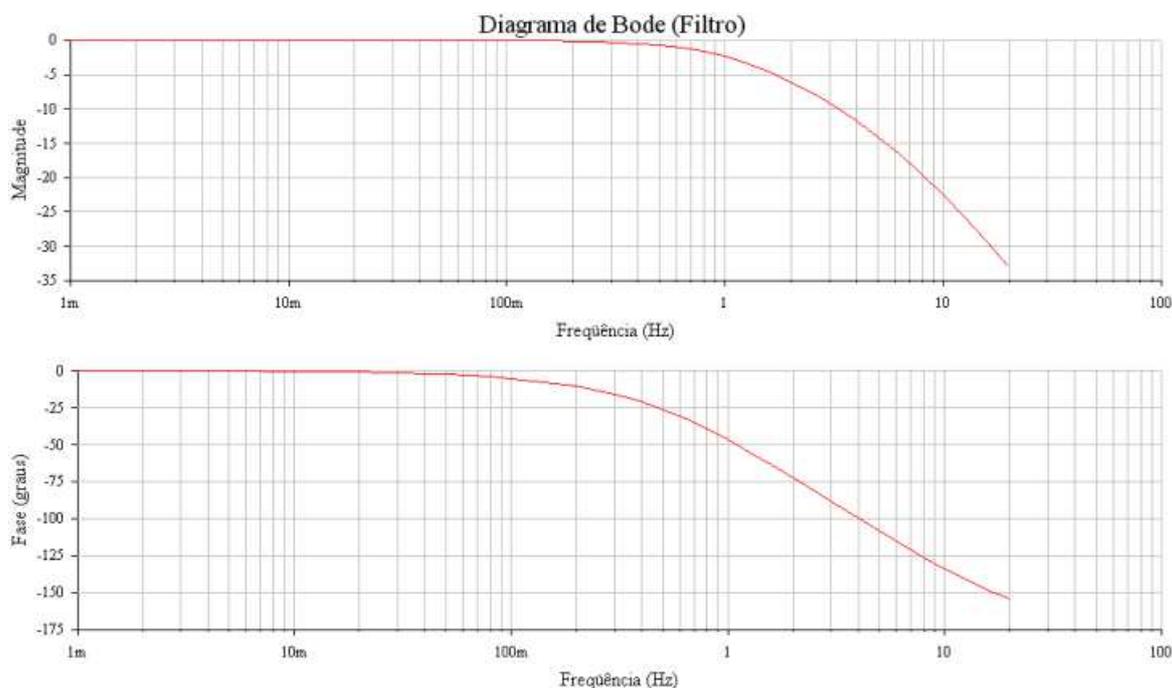


Figura 3.9: Diagramas de Bode do filtro projetado

Conforme será visto, a frequência natural de interesse da estrutura é de 2,54 Hz, desse modo o comportamento dinâmico da estrutura a ser modelado e controlado está contido pela banda de passagem do filtro.

3.3.2 Atuadores

Os motores movimentam os fusos através de polias e são comandados por circuitos de acionamento de potência que por sua vez são controlados por um sistema de aquisição de dados. O esquema geral de fluxo de dados nos equipamentos que compõem a montagem é mostrado na Seção 3.2.1.

O circuito de acionamento do motor da base é apresentado na Figura 3.10. Trata-se de um acionador usual de motores de corrente contínua construído à base de amplificadores operacionais e transistores de potência, com realimentação em tensão. O circuito é alimentado com baterias de 12 V e o sinal de entrada passando por um amplificador operacional como isolador (*buffer*) e por um amplificador inversor (LM324) que, conforme o valor do sinal positivo ou negativo com relação ao terra faz conduzir um ou outro transistor de acionamento (TIP142 ou TIP147) para mover o motor na direção desejada.

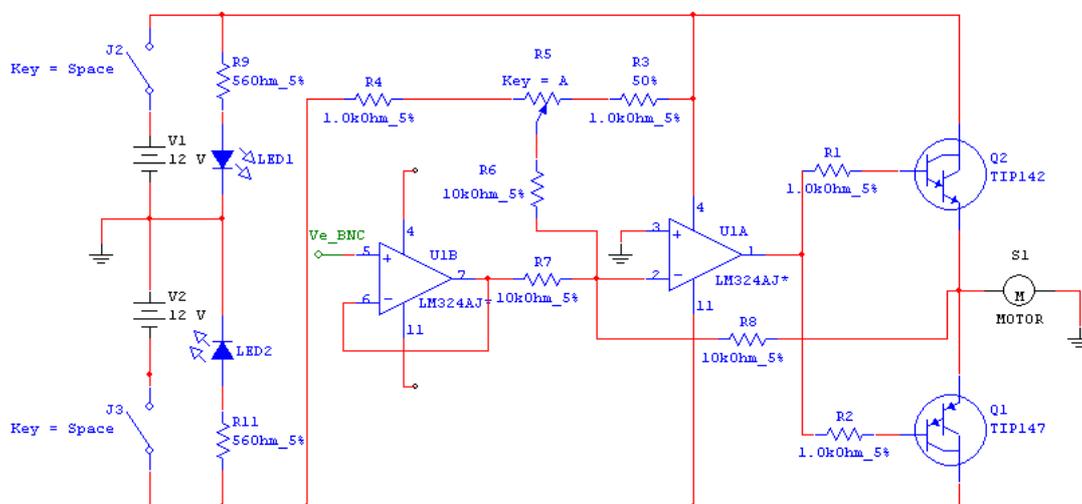


Figura 3.10: Circuito de acionamento do motor da base com realimentação em tensão

O circuito de acionamento do motor acoplado à massa superior, apresentado na Figura 3.11, possui os mesmos componentes, porém é realimentado em corrente. A corrente do motor, na forma da tensão medida sobre o resistor em série com a armadura é realimentada no amplificador para garantir seu ajuste. Ou seja, para forçar uma corrente no motor proporcional ao valor do sinal de controle.

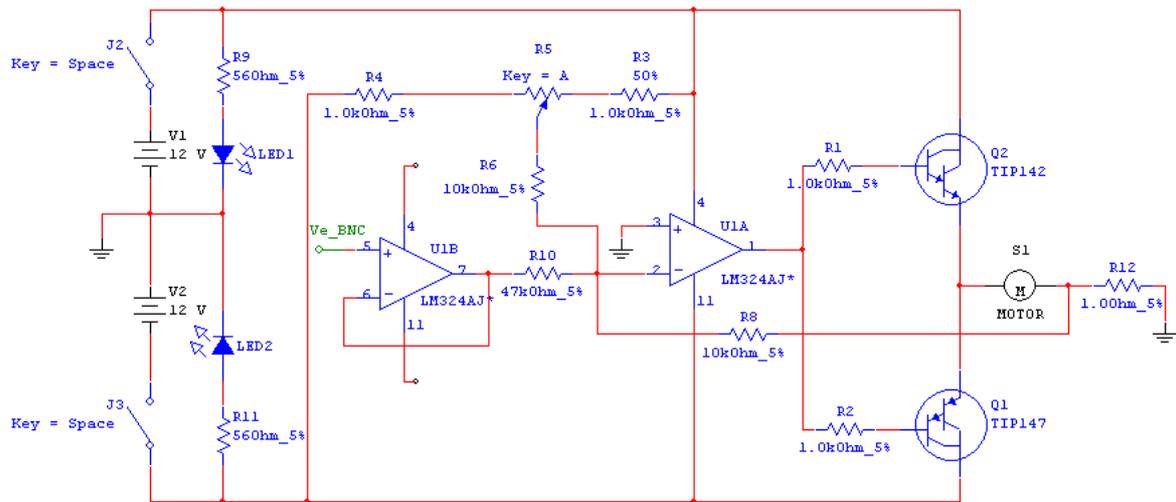


Figura 3.11: Circuito de acionamento do motor de compensação com realimentação em corrente

Para o sistema embarcado, cuja placa microcontrolada possui como saída de controle um sinal modulado em largura de pulso, o motor superior é acionado por um terceiro circuito. Foi utilizado, inicialmente, um circuito de acionamento comercial na forma integrada: o LMD18200T.

Esse circuito integrado é de propósito geral e aciona diversos tipos de motores de corrente contínua e de passo. Possui ainda várias funcionalidades como um pino para leitura de corrente do motor e outro para temperatura. O acionamento é realizado através de dois pinos: para um deles deve ser passado o sinal em PWM e para outro, o de direção. Com esses dois sinais, o circuito produz uma saída de potência em PWM com polaridade variando seguindo indicação do sinal de direção.

O LMD18200T produz uma saída de largura de pulso modulada exatamente como o sinal de entrada fornecido, porém com potência para acionar um motor qualquer dentro dos seus limites de tensão e corrente (3 A e 35 V) com a rotação do motor informada pelo pino de direção. O diagrama funcional é mostrado na Figura 3.12. Capacitores adicionais, não presentes na figura indicada, são colocados no circuito seguindo recomendações do fabricante.

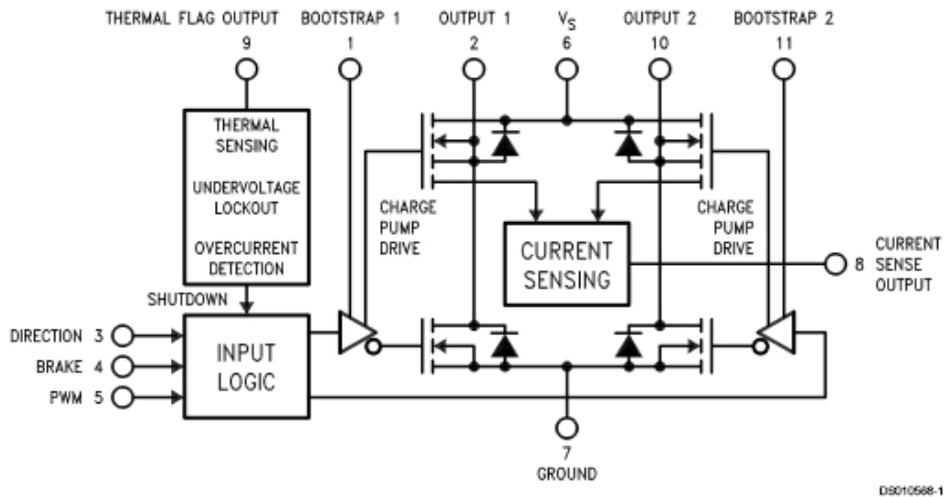


Figura 3.12: Diagrama funcional do LMD18200T

Pode-se notar na figura acima que dois sinais devem ser fornecidos ao circuito integrado de acionamento: o sinal em PWM e um sinal de direção, indicando o sentido de rotação do motor e, conseqüentemente a polaridade da alimentação que deve ser fornecida a ele.

Esses dois sinais são produzidos no sistema embarcado e passam por circuitos de isolamento, para evitar dano por excesso de corrente e para permitir ajuste dos valores de tensão necessários para acionar o circuito e para garantir a estabilidade do sinal.

O sinal de controle na forma de PWM é tratado por um circuito que isola a placa MSC1210 do acionamento do motor na forma ilustrada na Figura 3.13. O circuito de controle é alimentado com 5 V e o de acionamento com 12 V. Há um comparador na entrada desse circuito (LM324) para garantir o nível lógico de tensão e isolar o sistema embarcado da necessidade de corrente do circuito adiante pela alta impedância de entrada do amplificador operacional. Ligado a esse comparador, há um acoplador ótico (H11A1) que isola eletricamente os circuitos de controle e o de acionamento.

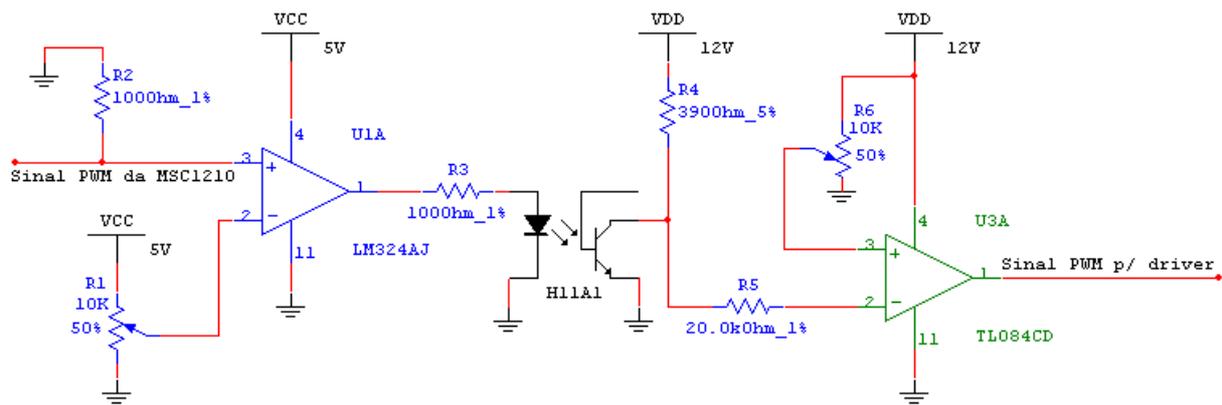


Figura 3.13: Circuito para isolar o sinal em PWM entre acionamento e controle

Uma vez que a tensão lida no acoplador é obtida da polarização do coletor, o sinal de tensão na saída é logicamente invertido com relação à entrada. Para corrigir isso, colocou-se um amplificador na forma comparador inversor (construído com TL084CD) na sua saída. Tanto a saída do acoplador ótico quanto o amplificador são alimentados com 12 V. Dessa forma, o sinal na saída do circuito é de 12 V ou 0 V.

Para o sinal que informa o sentido de rotação do motor, um isolamento semelhante entre controle e acionamento foi elaborado. Conforme se vê na Figura 3.14, usou-se um amplificador operacional como comparador (LM741) ligado a um acoplador ótico. Também neste caso, o sinal na saída do acoplador está invertido com relação à entrada do circuito, porém, a inversão foi corrigida no programa de controle.

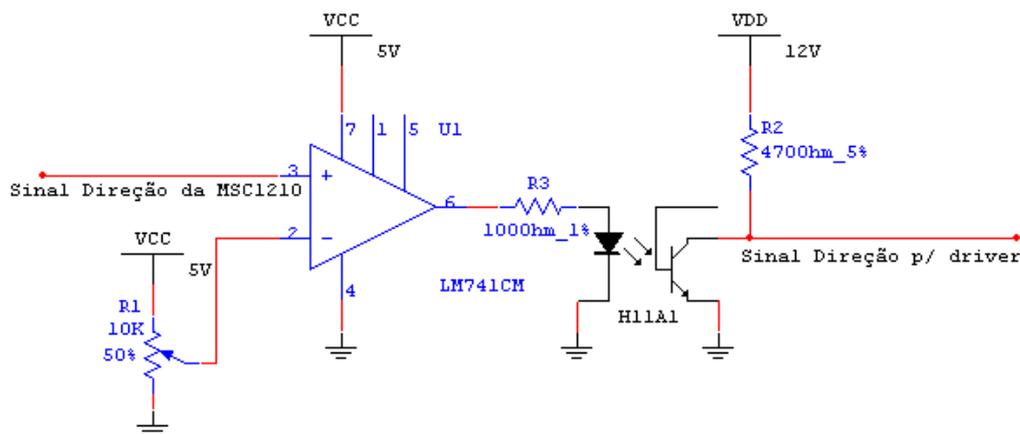


Figura 3.14: Circuito para isolar o sinal de direção entre acionamento e controle

A forma aqui descrita para o acionamento do sistema é, evidentemente, apenas uma das possíveis.

Uma forma de menor custo é implementada inteiramente utilizando componentes discretos e é utilizada para identificação e controle. Para tanto, montou-se uma ponte H para acionamento do motor e um circuito particular de isolamento e tratamento dos sinais de controle (dividido no sinal em PWM e no de direção). Essa outra abordagem é ilustrada na Figura 3.15.

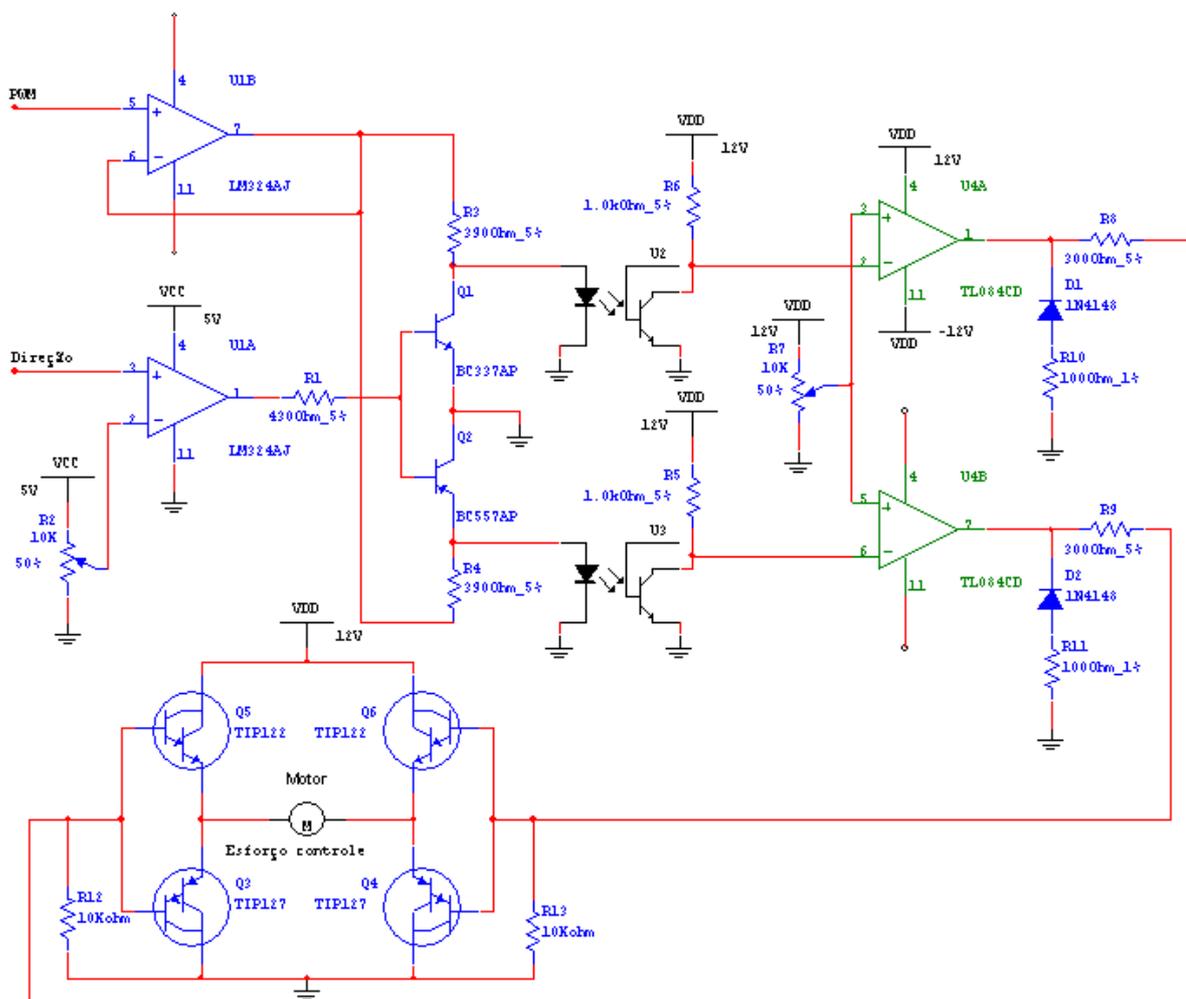


Figura 3.15: Circuito responsável pelo sinal em PWM a partir de componentes discretos

Na forma representada pela figura referenciada acima, há dois amplificadores operacionais (LM324) funcionando como isoladores para alimentar um par de transistores. Esses transistores, um NPN e outro PNP (BC337 e BC557), dividem o sinal em PWM em dois sinais separados. Um dos dois sinais em PWM produzidos está sempre desativado enquanto o outro funciona, trocando de papel conforme o sinal de direção. Novamente para isolamento elétrico, esses sinais passam por optoacopladores e isoladores. Os sinais de PWM divididos são utilizados para alimentar uma ponte H (usando transistores TIP122 e TIP127, que já possuem diodo de segurança interno) que ativa o motor.

3.4 Sistema embarcado

3.4.1 Descrição da placa

Nessa configuração, os sinais de leitura dos extensômetros e o de controle em PWM, são tratados pelo sistema embarcado, bem como o monitoramento da resposta da estrutura mecânica sob a ação do controle. Trata-se de um sistema que usa uma placa de conversão A/D de alta precisão (24 bits), maior que a da configuração anterior, e saída em PWM com 16 bits de resolução. A placa ainda disponibiliza memória Flash interna de 32KB e externa de 128KB, duas portas seriais a qual, uma delas, foi usada para o monitoramento da estrutura em resposta ao controle.

Tendo em vista nossa saída de controle na forma de PWM, pôde-se utilizar o sinal da buzina para a sua geração, uma vez que seu acionamento é feito de maneira análoga. Para contemplar o sistema de acionamento, o sinal do LED amarelo foi usado para enviar o sinal de direção ao motor que gira, ora em um sentido, ora em outro, conforme seu estado. O sinal em PWM, o sinal de direção e o transmitido pela porta serial utilizados neste trabalho podem ser observados no esquema representado pela Figura 3.16, como sendo “*Speaker*”, “*YellowLED*” e “*Serial 0*”, respectivamente.

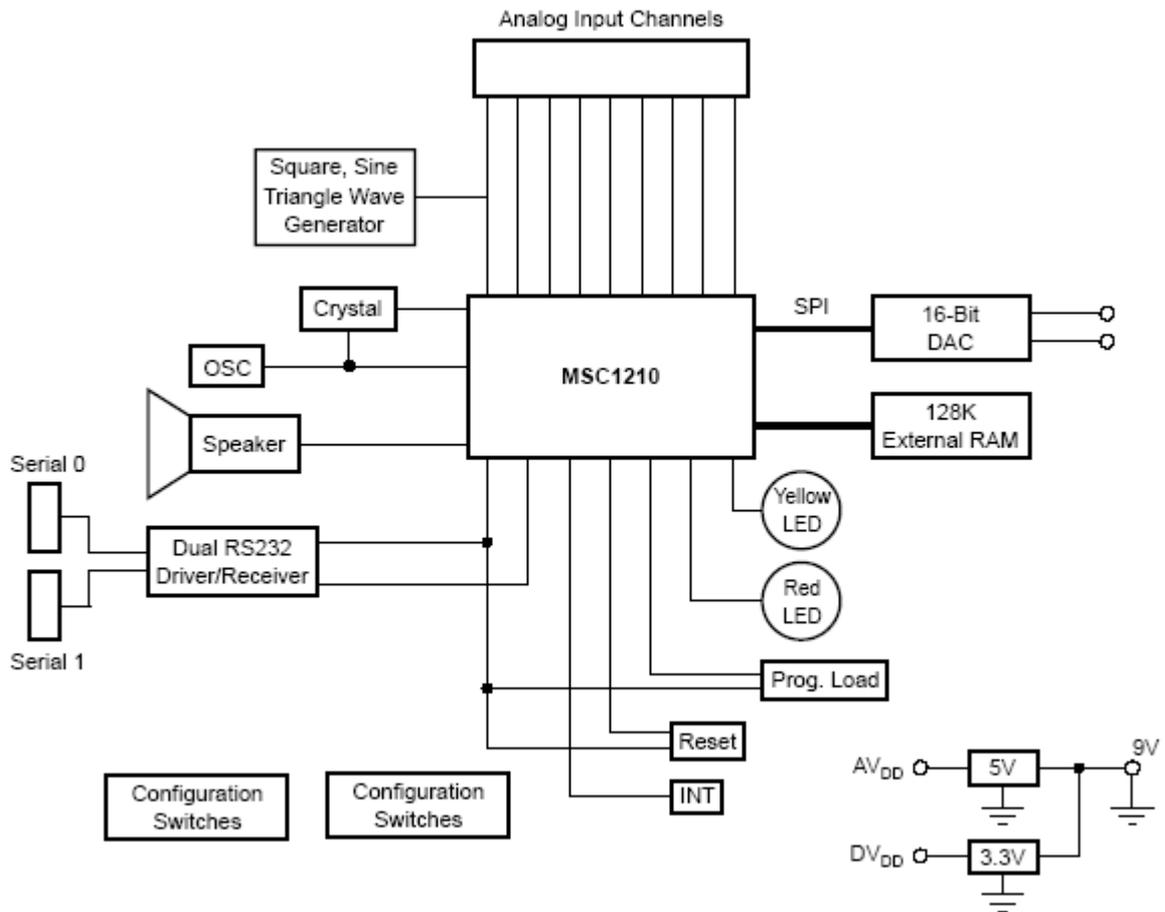


Figura 3.16: Diagrama em blocos do kit MSC1210EVM

Com essas configurações a montagem utilizando o sistema embarcado pode ser resumida como segue:

- os extensômetros produzem um sinal de tensão associado à deformação;
- esse sinal passa por um amplificador de sinal (AD524);
- o sinal é lido pela placa de conversão analógico-digital;
- o sistema embarcado realiza um processamento programado via computador utilizando a linguagem de programação C, que pode ser o cálculo do controlador ou outro qualquer;
- a placa de saída envia o sinal em PWM para o circuito de acionamento de controle;
- o circuito de acionamento alimenta o motor de controle para que se mova.

Quando o controle é realizado pelo sistema embarcado, a montagem é alterada no sentido de substituir o canal A/D de aquisição, bem como o cálculo do controle, ambos do sistema DSP. Desse modo, o esquema de fluxo de dados e montagem é modificado e está ilustrado pela Figura 3.17.

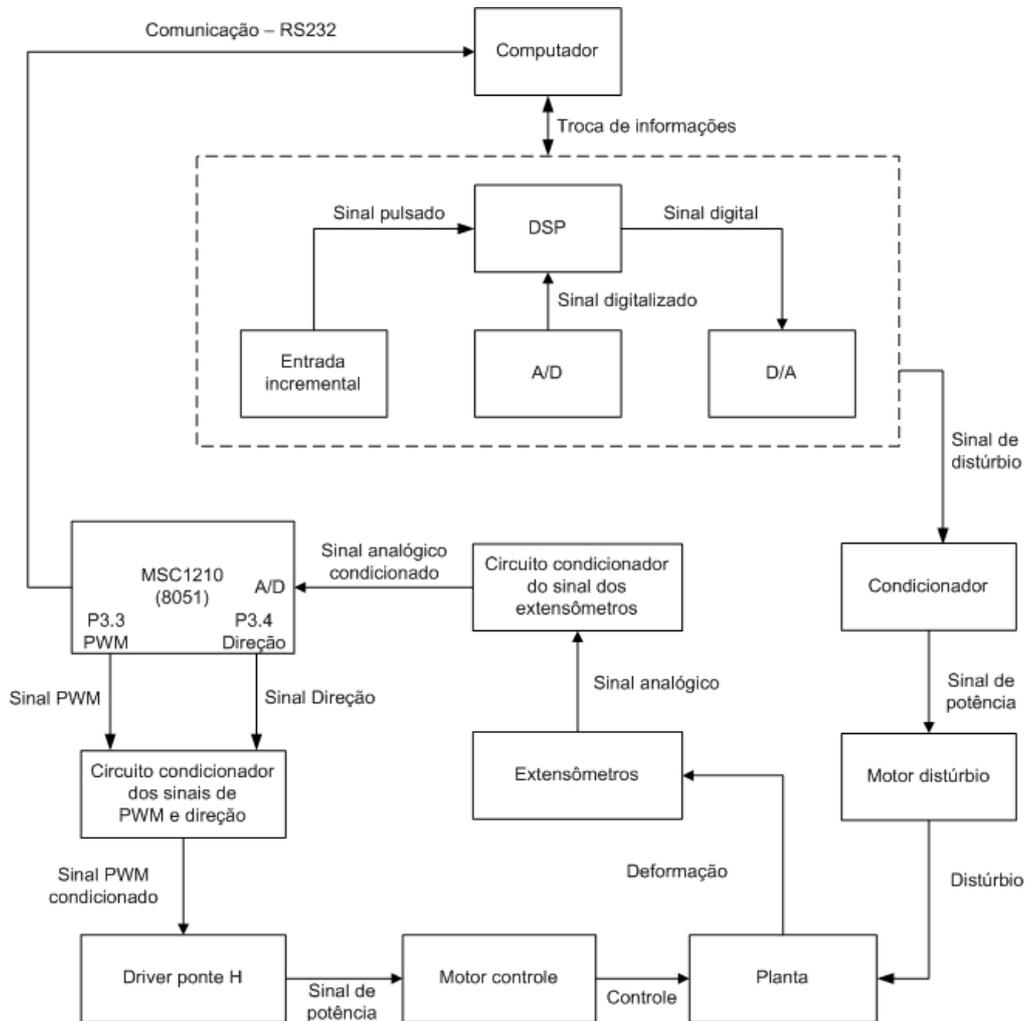


Figura 3.17: Fluxo de informação na montagem com o sistema embarcado

3.4.2 Descrição do programa

Nessa instância, o ambiente de programação μ Vision IDE desenvolvido pela Keil software foi utilizado para o desenvolvimento do programa. A linguagem C foi adotada para programar o microcontrolador aplicando métodos de programação em tempo real, descrito passo a passo no Capítulo 4.

A programação em tempo real no modo como implementada nessa variante da linguagem C separa o programa em funções a serem executadas de forma concorrente, dividindo o problema em aquisição de dados, cálculo do controle, geração do sinal de controle em PWM e transmissão pela porta serial.

O cálculo do controle utiliza as facilidades da linguagem de alto nível de operações em ponto flutu-

ante, vetores e matrizes. Desse modo, o controlador projetado, na sua forma discreta, é implementado transcrevendo as operações matriciais que o definem no espaço de estados diretamente para linguagem C.

O sinal de controle, conforme já esclarecido, deve ser ajustado e escalado para transmissão na forma de PWM pelo módulo da placa preparado para isso. Nesse módulo o sinal em PWM pode ser ajustado basicamente com dois parâmetros: período e ciclo ativo.

Esse módulo evita o trabalho de programação para produzir um sinal de largura de pulso ajustável regulando tempos de alto e baixo usando uma lógica de interrupções.

A versão do compilador utilizada era estudantil e possuía limitações na capacidade de otimização do código e no tamanho que permitia para o programa (abaixo da memória da placa). Dessa forma, algumas alterações no código tiveram de ser feitas. A transmissão serial, que poderia ter sido executada usando uma função de alto nível da linguagem transmitindo os dados na forma final, teve de ser feita em baixo nível, transmitindo blocos de bits usando um procedimento detalhado no Apêndice C.

Capítulo 4

Programação do sistema embarcado

Neste capítulo é apresentada a programação em tempo real para o microcontrolador do sistema embarcado. O programa é descrito como um conjunto de máquinas de estados e os diversos diagramas usando a metodologia SDL (do inglês “specification and description language”) são apresentados e explicados.

4.1 Acionamento com sinal em modulação por largura de pulso

Conforme já mencionado, foi utilizado para o sistema embarcado uma saída modulada em largura de pulso para o sinal de controle. A largura de pulso determina o tempo durante o qual se mantém energizadas as bobinas do motor que efetua o esforço de controle. Isso permite a variação necessária para a energia transmitida a ele. Assim, o maior valor utilizado do sinal de controle deve corresponder no máximo a 100% do ciclo de trabalho do pulso, ou seja, um nível constante de voltagem. O programa implementado realiza a aquisição do dado em um intervalo de amostragem predeterminado, aciona a sua transmissão serial, calcula o percentual do ciclo de trabalho correspondendo a cada nível do sinal computado pelo algoritmo de controle respectivo e o aplica na saída PWM.

Utilizando a metodologia usual de projeto de sistemas em tempo real, são descritos: o diagrama de partição, o diagrama de interação, as máquinas de estados, os diagramas SDL e os fluxogramas de funcionamento para o sistema construído neste trabalho. Referências relevantes sobre o projeto de sistemas em tempo real podem ser encontradas em (Farines et al., 2000), (Belina et al., 1991) e (Ferreira, 2004).

A metodologia SDL foi utilizada assim para especificar completamente os diversos módulos do programa, os quais foram em seguida implementados usando a linguagem C com o compilador μ Vsion2. Para mais detalhes sobre a linguagem SDL consultar Apêndice D.

4.1.1 Divisão do problema

O diagrama de partição representado pela Figura 4.1, indica a hierarquia entre os principais módulos componentes do sistema e a sua divisão em entradas e saídas. A simplicidade desse sistema não requer grande hierarquização. Consiste dos seguintes blocos:

- conversor A/D, como única entrada, é responsável pela aquisição do dado a cada intervalo de amostragem;
- temporizador, que realiza a contagem do tempo que define o intervalo entre as amostragens;
- gerenciador, o responsável por disparar, no momento adequado, o início do processamento de cada módulo;
- controlador, executando a tarefa de cálculo do controle e armazenamento do resultado;
- sinal de controle, que faz o papel de ajustar a saída de controle em PWM;
- porta serial, faz a transmissão do sinal lido dos extensômetros para o computador.

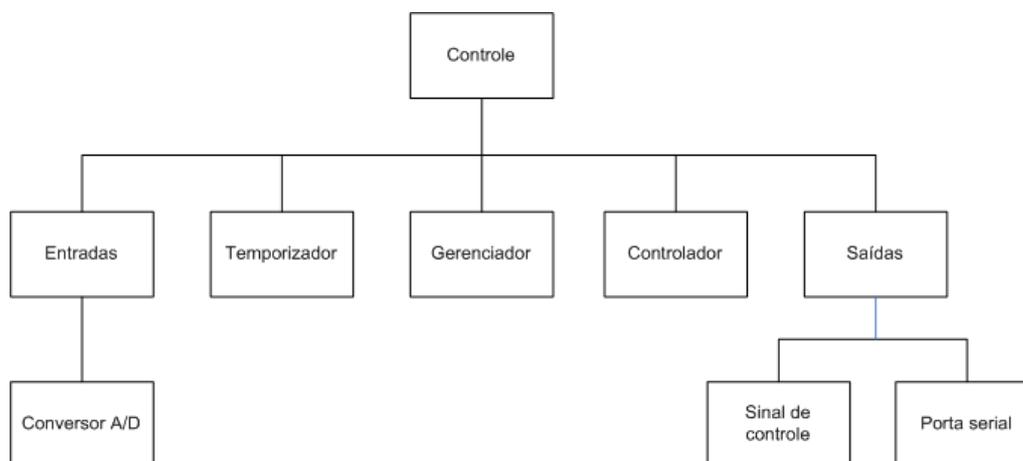


Figura 4.1: Diagrama de partição do sistema de controle em tempo real

O diagrama de interação visto na Figura 4.2, mostra como os módulos acima listados se comunicam. O módulo Gerenciador, através de sinais, ordena o início das rotinas de conversão A/D, cálculo do controle e seu ajuste para saída em PWM, bem como a comunicação serial. Cada um desses módulos sinaliza o fim de suas próprias atividades. O Gerenciador espera então por um sinal vindo da temporização (que deve ser programada para ocorrer após o fim das atividades dos blocos citados) para

ordenar o início no próximo ciclo. A temporização do sistema, portanto, é realizada por um contador de tempo (contando pulsos do relógio do sistema) que deve ser ajustado para corresponder ao tempo de amostragem desejado para o sistema em sua representação discreta.

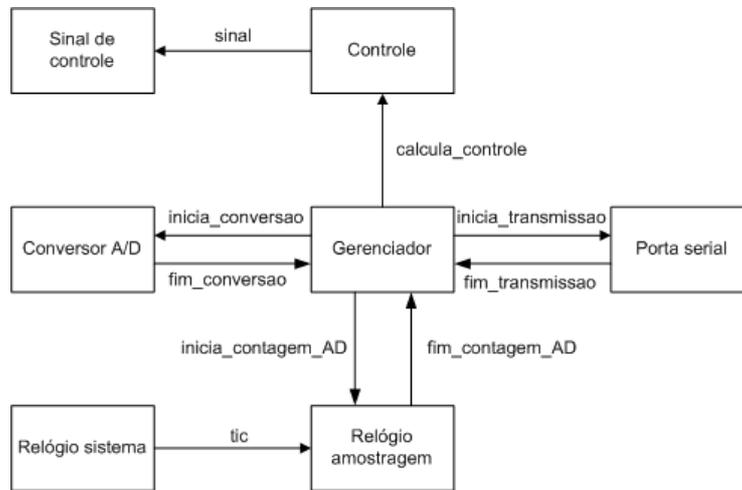


Figura 4.2: Diagrama de interação do sistema

4.1.2 Máquinas de estados e diagramas SDL

As máquinas de estados representam de forma clara, o funcionamento do código de programa. Uma vez bem definidas, facilitam o trabalho do programador que tem a possibilidade de desenvolver programas mais consistentes. Outra metodologia utilizada para a programação em tempo real foi o SDL (Belina et al., 1991), citado anteriormente. Essa metodologia é desenvolvida de maneira mais fácil quando se tem as máquinas de estados definidas. Os diagramas SDL representam as esperas e os módulos que constituem o programa.

Na Figura 4.3 está representada a máquina de estados para o Conversor A/D com dois estados: Espera inicia_conversao e Espera fim_conversao. No Estado inicia_conversao, a máquina de estados aguarda o sinal de sincronismo que determina o intervalo de amostragem, gerado pelo Gerenciador. No Estado fim_conversao, fica aguardando pelo sinal fim_conversao gerado pelo hardware do módulo Conversor A/D. A Figura 4.4 é o diagrama usando a metodologia SDL que representa essa máquina de estados.



Figura 4.3: Máquina de estados do módulo Conversor A/D

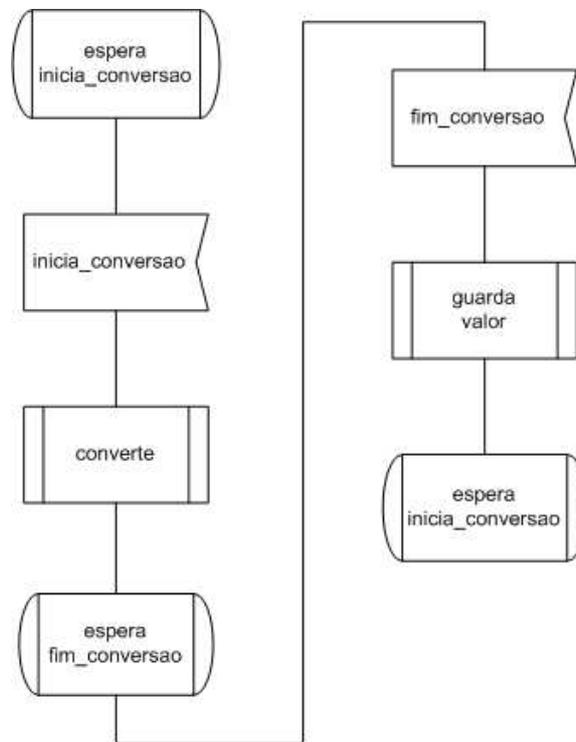


Figura 4.4: Diagrama SDL do módulo Conversor A/D

O módulo chamado Controle, representado na Figura 4.5, aguarda pelo sinal `calcula_controle` vindo do Gerenciador para executar sua tarefa de cálculo e armazenamento do resultado, bem como o sinal para o módulo Sinal de controle. O cálculo consiste das multiplicações matriciais do modelo discretizado do controlador. O resultado é enviado como sinal produzido pelo módulo. O diagrama SDL apresentado na Figura 4.6 representa a máquina de estados do módulo Controle.

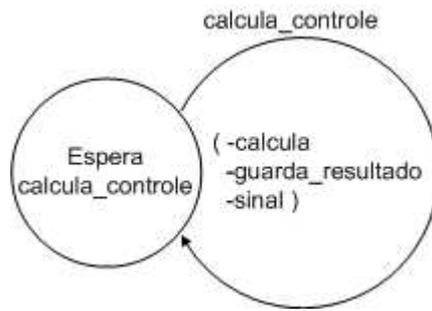


Figura 4.5: Máquina de estado do módulo Controle

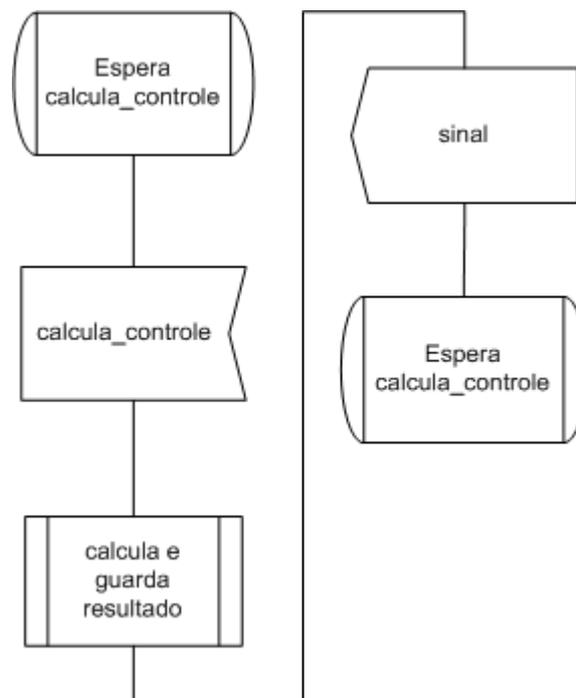


Figura 4.6: SDL do módulo Controle

A Figura 4.7 mostra o contador que garante um intervalo de tempo constante entre duas amostras do Conversor A/D.

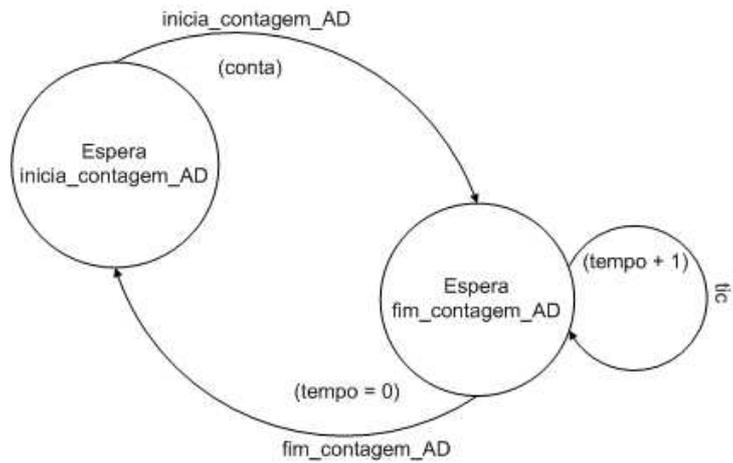


Figura 4.7: Máquina de estados do módulo Relógio amostragem

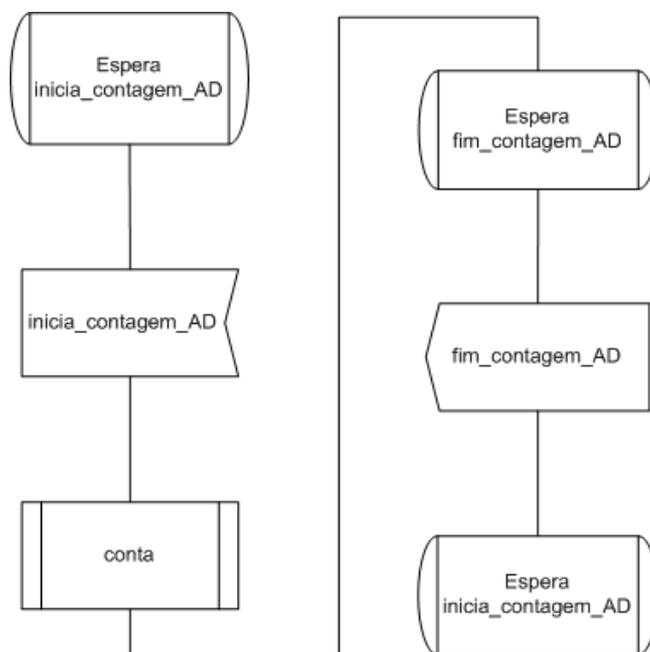


Figura 4.8: SDL do módulo Relógio amostragem

Ele está dividido em dois estados: `Espera inicia_contagem_AD` e `Espera fim_contagem_AD`. O Estado `inicia_contagem_AD` fica esperando pelo sinal `inicia_contagem_AD` vindo do Gerenciador para começar a contagem e passa para o próximo estado, `Espera fim_contagem_AD`. Nele, espera-se pelo sinal `fim_contagem_AD` ajustado pelo circuito do módulo de conversão A/D do microcontrolador. Em princípio, pode-se simplesmente testar pelo fim da conversão nesse bit vindo do circuito do conversor. Porém, este contador garante que a espera seja determinada por uma constante de modo a se ter controle sobre o tempo entre as amostras. A Figura 4.8 apresenta o diagrama SDL para essa máquina

de estados.

A transmissão serial, representada pela Figura 4.9, também é dividida em dois estados: Espera inicia_transmissao e Espera fim_transmissao.

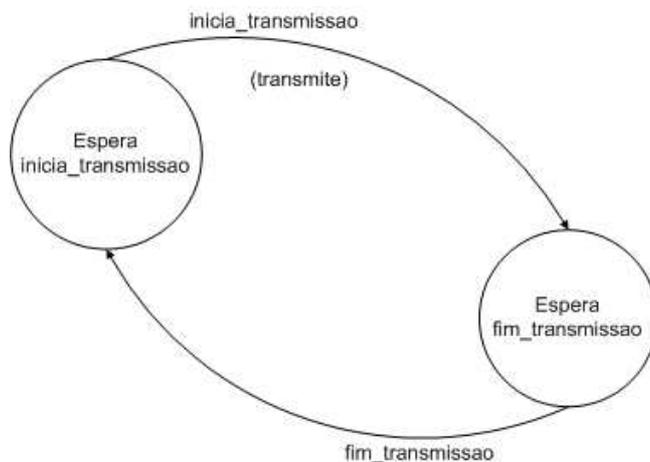


Figura 4.9: Máquina de estados do módulo Transmissão serial

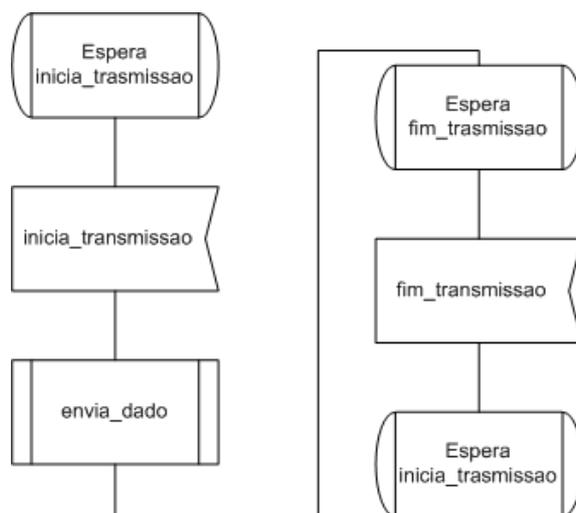


Figura 4.10: SDL do módulo Transmissão serial

No estado Espera inicia_transmissao, fica-se aguardando pelo sinal de inicia_transmissao vindo do Gerenciador para enviar os dados lidos da estrutura mecânica ao computador. No estado Espera fim_transmissao, um sinal ajustado pelo circuito do módulo de transmissão serial do microcontrolador é aguardado para permitir o reinício da transmissão. A representação dessa máquina de estados na forma de diagrama SDL é mostrada na Figura 4.10.

O Gerenciador, representado na Figura 4.11, aguarda pelo fim dos procedimentos dos outros

módulos em conjunção (todos eles devem ter terminado) antes de ordenar o reinício de todas as atividades. Portanto, espera-se, primeiramente, pelos sinais de fim_contagem_AD, fim_conversao e fim_transmissao, que devem acontecer a cada intervalo de amostragem, para enviar os sinais de inicia_conversao, inicia_transmissao, calcula_controle e inicia_contagem_AD. O diagrama SDL, da Figura 4.12, representa essa máquina de estado.

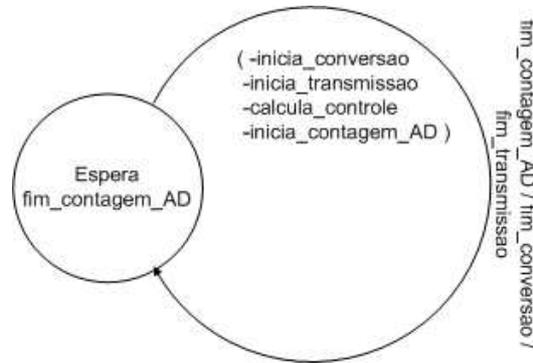


Figura 4.11: Máquina de estado do módulo Gerenciador

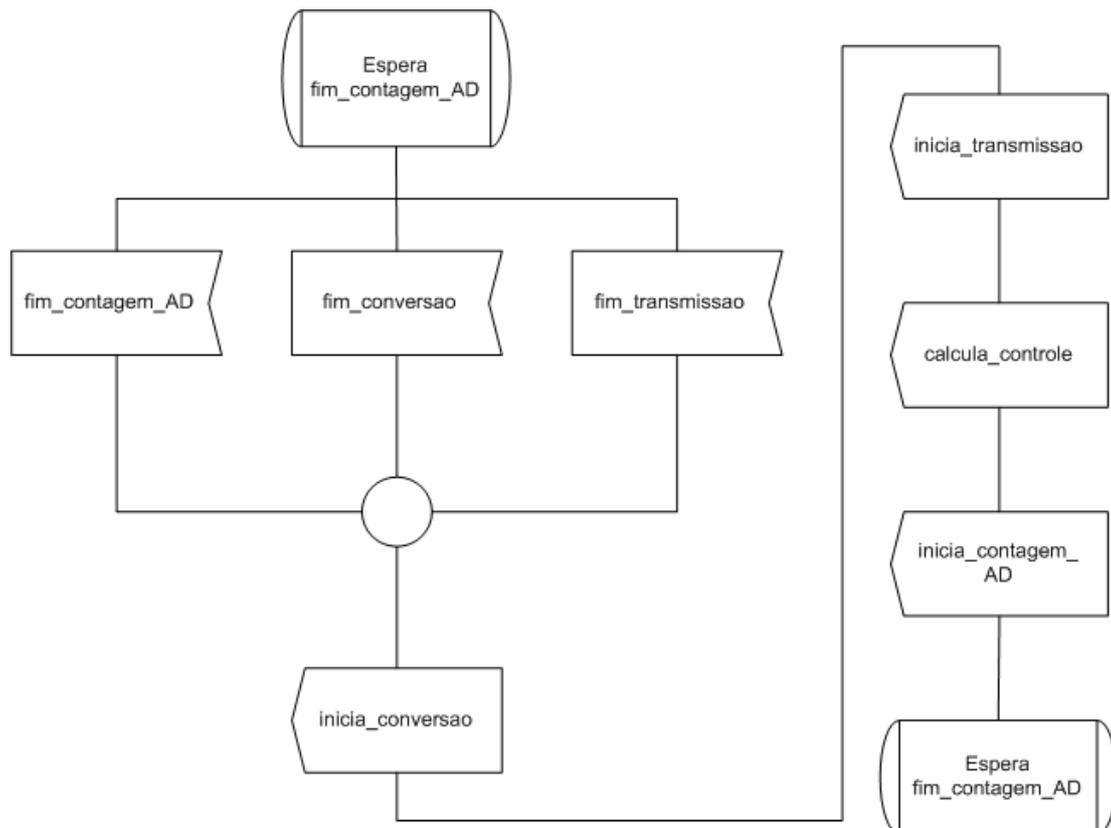


Figura 4.12: SDL do módulo de Gerenciador

O módulo ilustrado na Figura 4.13 simplesmente converte o sinal de controle produzido na saída

modulado em largura de pulso. Espera-se por sinal, vindo do módulo Controle, para que seja realizada a conversão do sinal de controle calculado. A Figura 4.14 representa essa máquina de estado na forma de diagrama SDL.

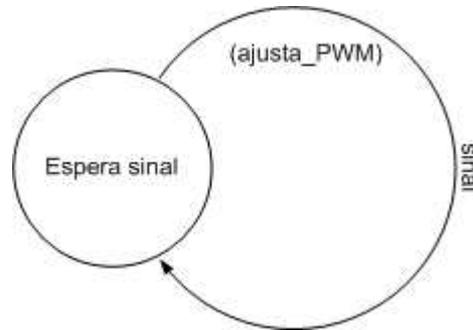


Figura 4.13: Máquina de estado do módulo Sinal de controle

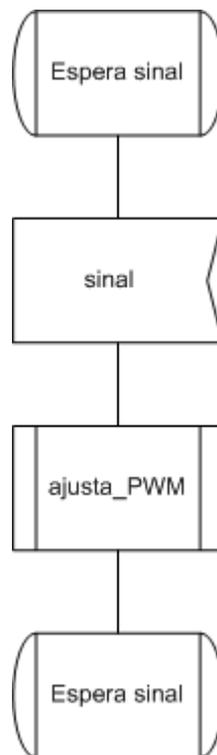


Figura 4.14: SDL do módulo Sinal de controle

4.1.3 Fluxogramas

O fluxograma representa a seqüência de execução dos comandos dos programas e deve seguir as operações conforme são descritas pelo diagrama SDL respectivo. Uma vez que o algoritmo de controle

em tempo real é usualmente interpretado como uma máquina de estado, elabora-se um fluxograma para cada estado de espera dos diagramas SDL, que representam subsistemas do sistema completo.

A implementação final do sistema na forma de um programa exige, portanto, transferir os diagramas SDL para a forma de fluxogramas. O funcionamento dos módulos de programa em um sistema de tempo real é concorrente: cada módulo opera independentemente, controlado apenas pelos sinais recebidos dos outros. Esses sinais alteram o comportamento dos fluxogramas fazendo cada módulo *trocar de estado*. Dessa forma, cada interrupção do microcontrolador utilizado executa todos os módulos com uma seqüência qualquer. Cada execução é determinada pelo estado de cada módulo (no contexto de uma máquina de estados) que muda conforme os sinais que troca com os módulos restantes. Reproduz-se então o funcionamento exato explicitado para os diagramas SDL, preparando agora o sistema para implementação direta, na forma de programa, a partir dos fluxogramas.

O Conversor A/D, como pode ser visto na Figura 4.15, é dividido em dois estados: estado_AD_0 e estado_AD_1. Quando `inicia_conversao` e `fim_conversao` são iguais a um respectivamente, o módulo sai do modo de espera e executa os blocos subseqüentes. Cada módulo é responsável por consumir o bit que o tirou da espera, como é o caso quando faz-se `inicia_conversao` igual a zero.

Na Figura 4.16, o módulo Controle fica esperando um bit do Gerenciador para começar sua execução. Como nos outros módulos, ele deve consumir o sinal que o tirou da espera. Em seguida, ele calcula o controle e envia um sinal que será tratado pelo módulo Sinal de controle. Consumir o sinal significa reinicializar a variável que o representa.

O módulo Relógio amostragem, na Figura 4.17, mostra também sua divisão em dois estados. No primeiro estado, a contagem é iniciada quando do recebimento do sinal `inicia_contagem_AD`. A partir daí, inicia-se um procedimento de contagem até um valor constante escolhido. Ao fim da contagem, o estado é trocado para o seguinte. No próximo estado, espera-se por um sinal de fim da contagem que, quando recebido e consumido, passa-se novamente para o estado inicial, esperando pelo sinal de início da contagem.

O módulo Transmissão serial tem dois fluxogramas, mostrados na Figura 4.18. No primeiro deles, aguarda-se pelo sinal de início da transmissão. Ao recebê-lo, o envio de dados é iniciado e o estado é trocado para o segundo. No segundo estado espera-se pelo sinal de que a transmissão teve fim, para então voltar ao estado inicial.

O módulo Gerenciador possui apenas um estado, mostrado na Figura 4.19, que consiste em esperar pelos sinais de fim dos módulos Contagem AD, Conversão e Transmissão serial. Quando essa situação ocorre, o Gerenciador lança os sinais que iniciam novamente o ciclo de operações que ocorrem dentro

do tempo de amostragem: `inicia_conversao`, `inicia_transmissao`, `calcula_controle`, `inicia_contagem_AD`.

Por fim, na Figura 4.20, o módulo responsável pelo sinal de controle, aguarda por um sinal que ordena seu início para executar a rotina que ajusta a largura de pulso do sinal em PWM de modo proporcional ao sinal de controle calculado no controlador.

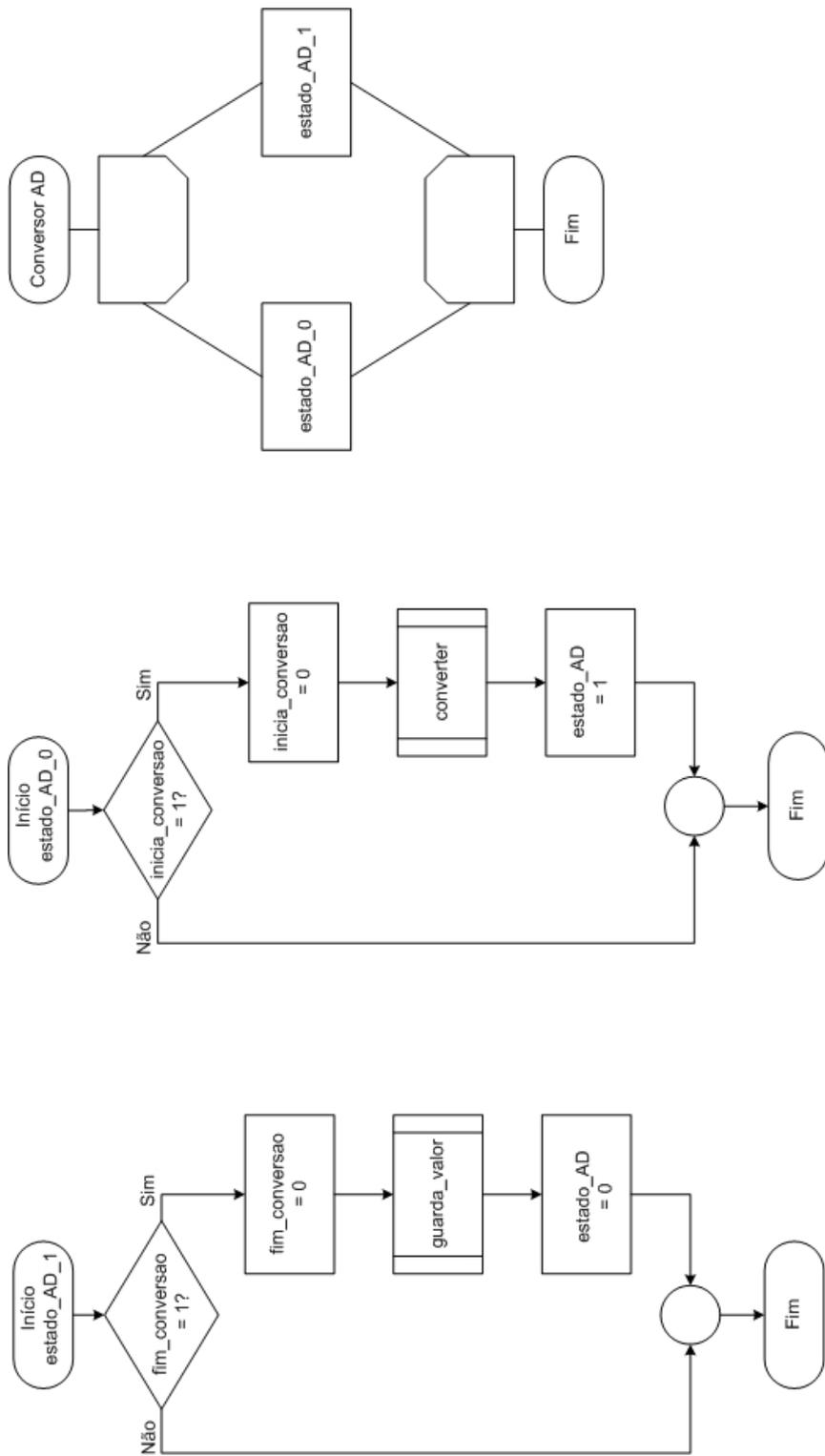


Figura 4.15: Fluxograma do módulo Conversor A/D

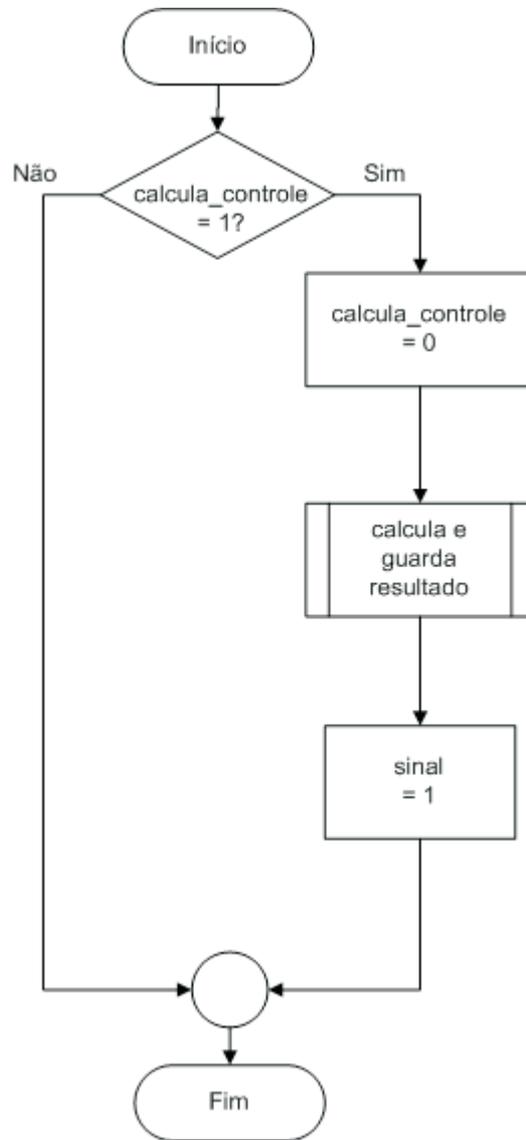


Figura 4.16: Fluxograma do módulo Controle

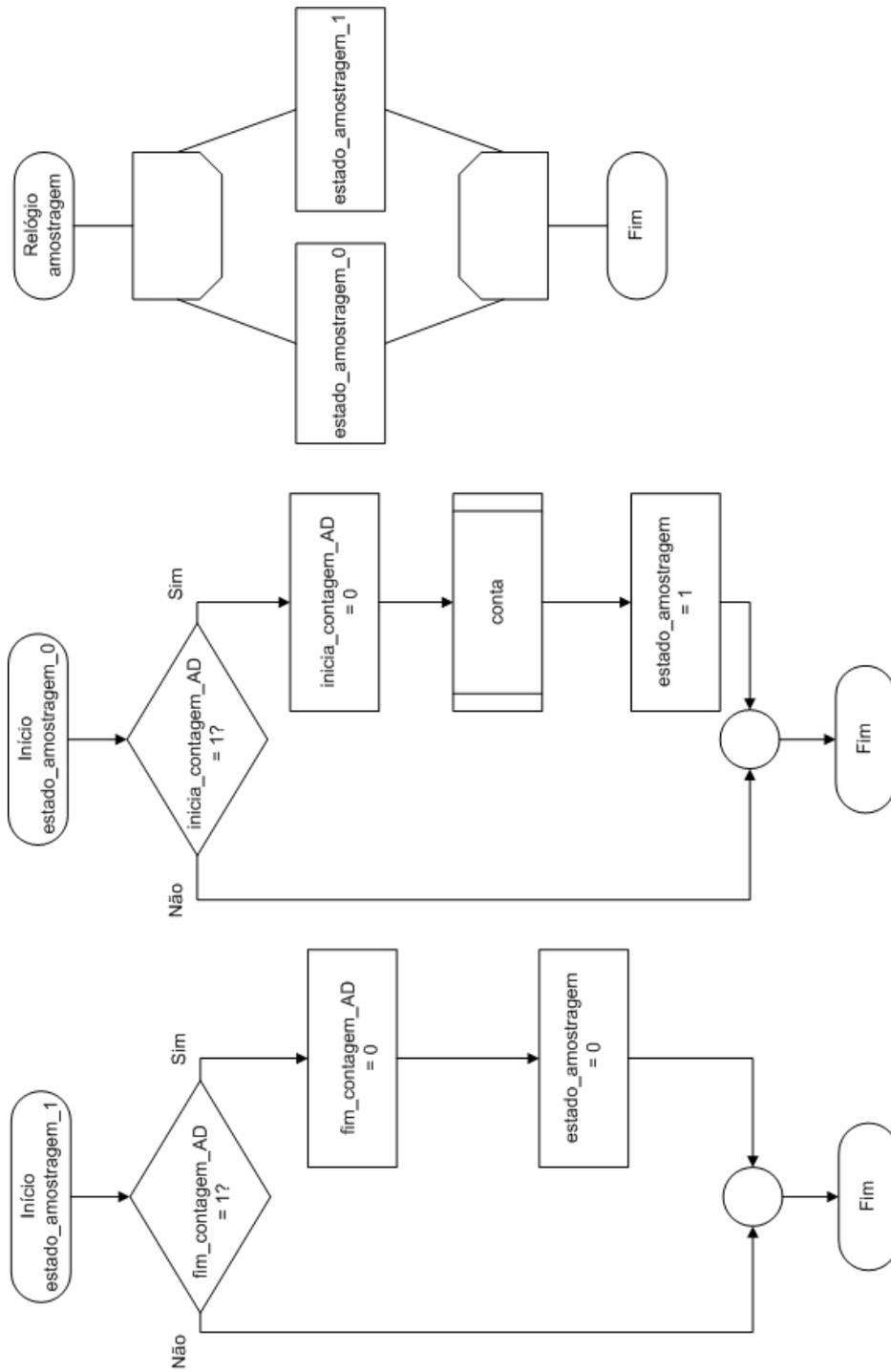


Figura 4.17: Fluxograma do módulo Relógio amostragem

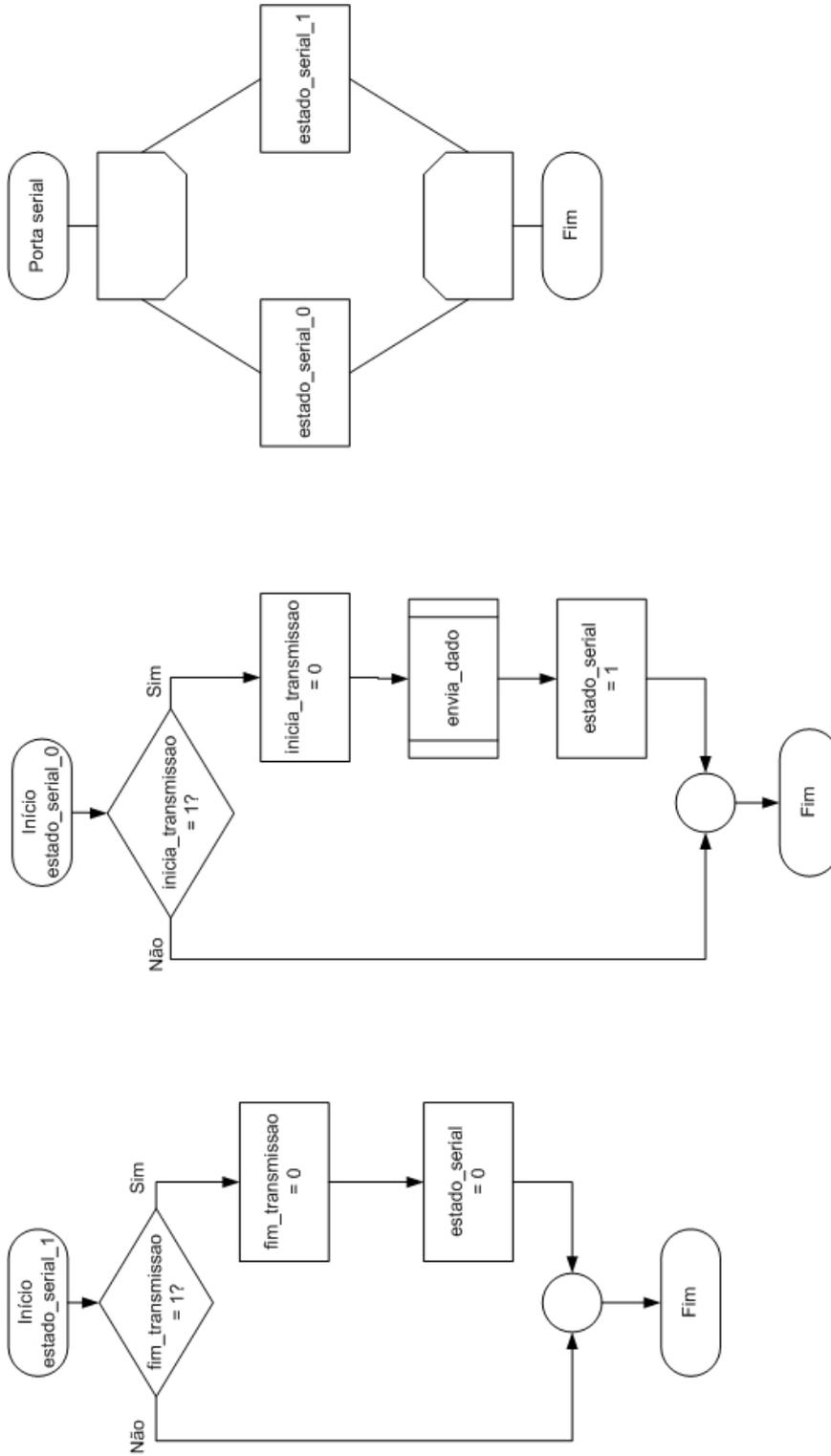


Figura 4.18: Fluxograma do módulo Transmissão serial

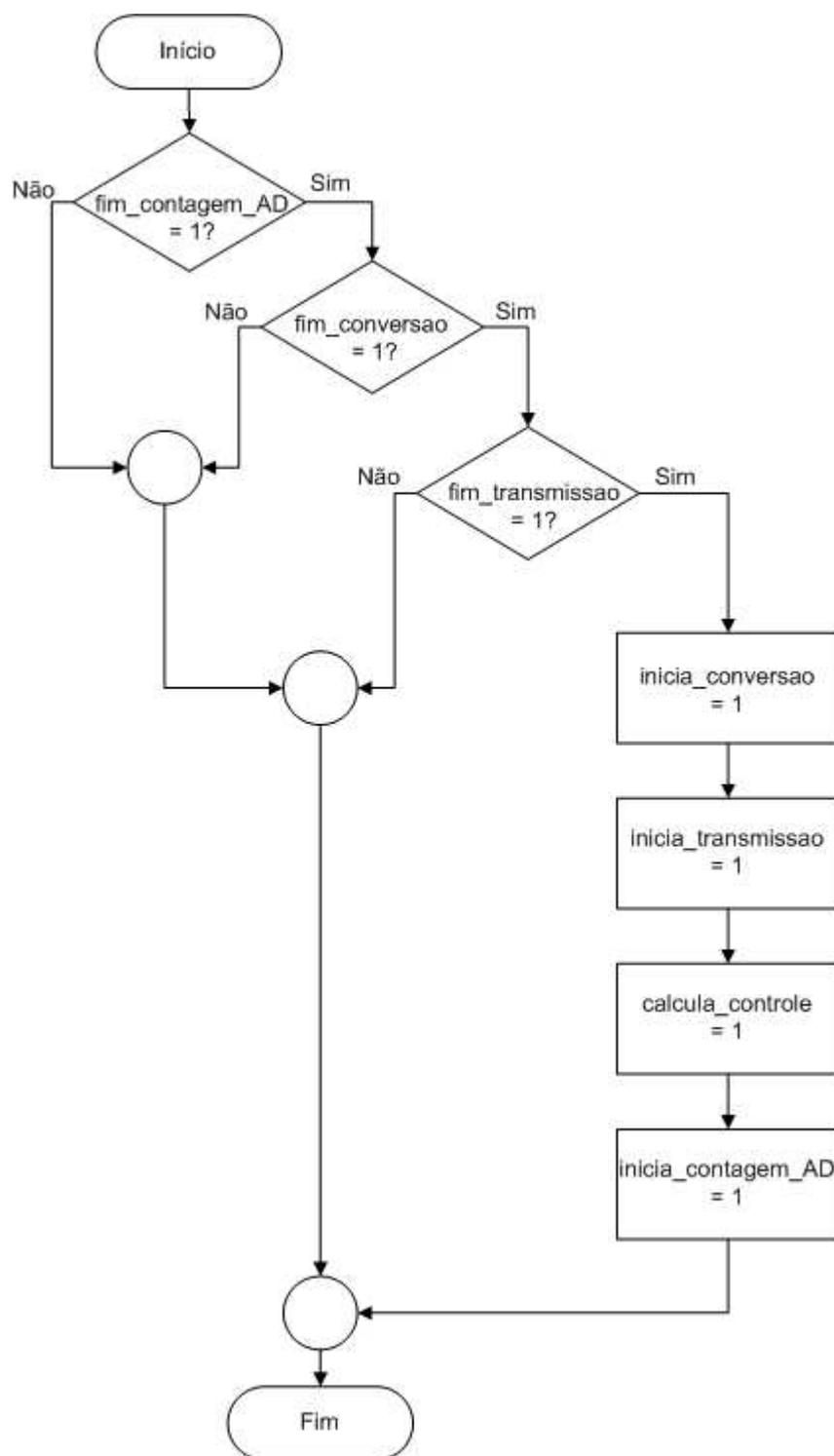


Figura 4.19: Fluxograma do módulo Gerenciador

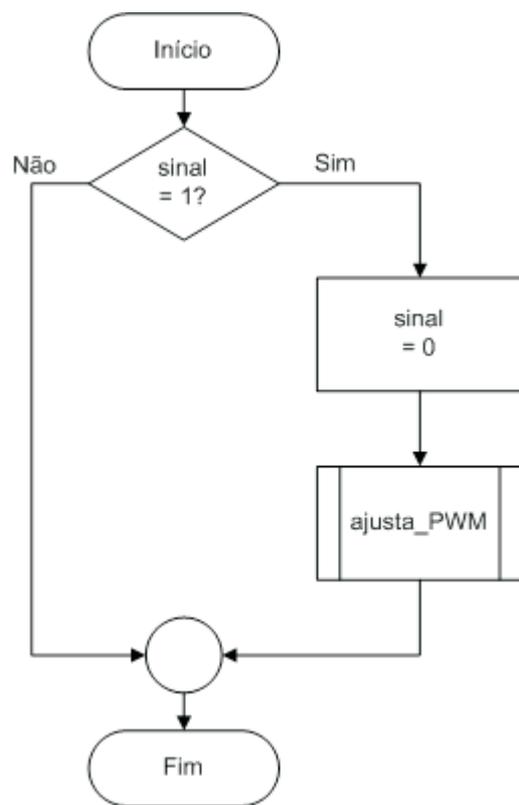


Figura 4.20: Fluxograma do módulo Sinal de controle

Capítulo 5

Modelos Analítico e Experimental

Neste capítulo são apresentados os modelos matemáticos utilizados no trabalho e a implementação dos algoritmos de controle. A modelagem analítica obtida a partir da aplicação do método de Lagrange é desenvolvida inicialmente. A seguir as identificações realizadas com a placa dSPACE são descritas. Nas seções seguintes apresentam-se as implementações para os sistemas DSP e embarcado.

5.1 Modelo analítico da bancada

A modelagem analítica dessa bancada de teste foi baseada em (Amorim, 2005).

A contribuição da massa inferior (base da estrutura) será modelada como um distúrbio. Ou seja, o motor inferior move a base de modo a perturbar a posição de repouso da massa superior (m_1). O modelo simplificado da dinâmica do sistema é composto de duas massas, sendo uma delas (m_2) destinada a ser controlada dinamicamente para atenuar a vibração. Entre o distúrbio e a massa m_1 há rigidez e amortecimento (k e c , respectivamente) devidos às quatro colunas unindo a massa superior e a base. Nesse modelo, utilizando-se o acionador com realimentação em corrente, não é necessário incluir a dinâmica do motor que move a massa m_2 uma vez que o torque é proporcional à corrente imposta. Desse modo, o distúrbio representa uma posição da base, chamada d , e a entrada do sistema para controle é a corrente no motor. Esses elementos são ilustrados na Figura 5.1.

A abordagem adotada aqui para a modelagem é a da Equação de Lagrange. Acrescenta-se que os graus de liberdade considerados são: a posição da massa m_1 , chamada de z , e a posição da massa m_2 , chamada de z_m . A posição da massa m_2 pode ser relacionada com a da massa m_1 e com a rotação do motor, chamada de θ , como $z_m = z + p\theta$. Em que p representa uma constante mecânica do fuso relacionando avanço e ângulo de rotação (passo do fuso em metros por radiano). O amortecimento angular do motor também deve ser considerado, representado na constante C .

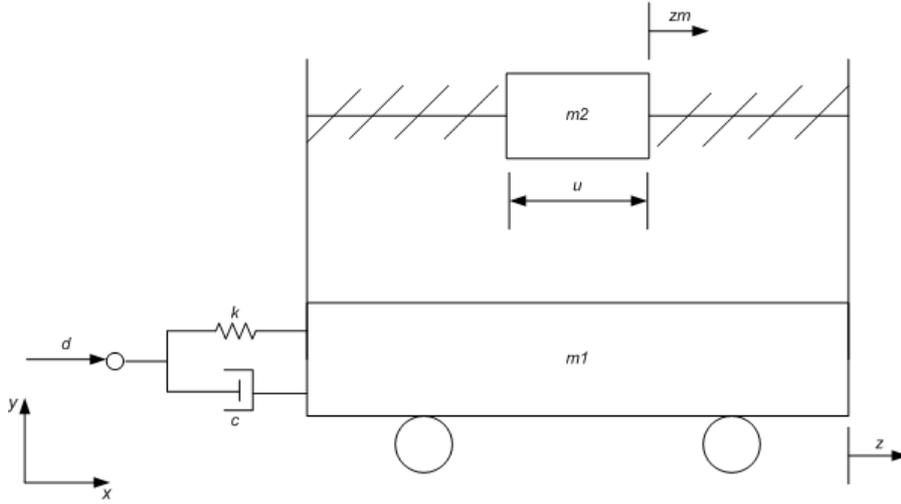


Figura 5.1: Esquema dos elementos modelados na dinâmica do sistema

Com esses parâmetros, podem-se escrever as energias cinética e potencial como:

$$E_c = \frac{1}{2}m_1\dot{z}^2 + \frac{1}{2}m_2\dot{z}_m^2 + \frac{1}{2}J\dot{\theta}^2 = \frac{1}{2}(m_1 + m_2)\dot{z}^2 + m_2p\dot{z}\dot{\theta} + \frac{1}{2}(m_2p^2 + J)\dot{\theta}^2.$$

$$E_p = \frac{1}{2}k(d - z)^2.$$

Aplicando a Equação de Lagrange a essas fórmulas da energia,

$$\frac{d}{dt} \left(\frac{\partial E_c}{\partial \dot{q}_i} \right) - \frac{\partial E_c}{\partial q_i} + \frac{\partial E_p}{\partial q_i} = Q_i,$$

onde q_i representa cada um dos graus de liberdade e Q_i representa as forças generalizadas sobre o sistema. As equações resultantes obtidas com z e θ como graus de liberdade são:

$$(m_1 + m_2)\ddot{z} + c\dot{z} + kz + m_2p\ddot{\theta} = kd,$$

$$m_2p\dot{z} + (m_2p^2 + J)\ddot{\theta} + C\dot{\theta} = \tau,$$

onde foi utilizada como força generalizada sobre o grau de liberdade θ o torque devido ao motor (τ) subtraído do amortecimento: $Q = \tau - C\dot{\theta}$.

Para finalizar o modelo, a equação dinâmica do motor, relacionando a corrente que recebe com o torque produzido deve ser acrescentada: $\tau = Ki$, em que K é a constante eletromecânica usual deste motor de corrente contínua. Realizando a substituição $\tau = Ki$ nas equações do modelo encontradas, tem-se:

$$(m_1 + m_2)\ddot{z} + c\dot{z} + kz + m_2p\ddot{\theta} = kd, \quad (5.1)$$

$$m_2 p \ddot{z} + (m_2 p^2 + J) \ddot{\theta} + C \dot{\theta} = K i. \quad (5.2)$$

As equações 5.1 e 5.2 podem ser convertidas para a forma de estados usando a entrada como a corrente do motor superior ($u = i$) e o vetor de estados conforme definido a seguir:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} z \\ \dot{z} \\ \dot{\theta} \end{bmatrix}.$$

Uma representação no espaço de estados possível,

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}_d d + \mathbf{B}_u u,$$

pode ser obtida isolando as derivadas segundas nas equações 5.1 e 5.2.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -\frac{k}{\alpha} & -\frac{c}{\alpha} & \frac{C m_2 p}{\alpha(m_2 p^2 + J)} \\ \frac{k m_2 p}{\alpha(m_2 p^2 + J)} & \frac{C m_2 p}{\alpha(m_2 p^2 + J)} & \beta \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{k}{\alpha} \\ -\frac{k m_2 p}{\alpha(m_2 p^2 + J)} \end{bmatrix} d + \begin{bmatrix} 0 \\ -\frac{K m_2 p}{\alpha(m_2 p^2 + J)} \\ \gamma \end{bmatrix} u,$$

em que foram realizadas as substituições:

$$\begin{aligned} \alpha &= (m_1 + m_2) - \frac{(m_2 p^2)}{(m_2 p^2 + J)}, \\ \beta &= -\frac{C(m_2 p)^2}{\alpha(m_2 p^2 + J)^2} - \frac{C}{(m_2 p^2 + J)}, \\ \gamma &= \frac{K(m_2 p)^2}{\alpha(m_2 p^2 + J)^2} + \frac{K}{(m_2 p^2 + J)}. \end{aligned}$$

Os parâmetros de rigidez e amortecimento foram obtidos pela resposta mensurada do sistema em comparação com os parâmetros de resposta padrão de um sistema de segunda ordem, pelas fórmulas:

$$c = 2\xi\omega_n(m_1 + m_2) \quad e$$

$$k = \omega_n^2(m_1 + m_2).$$

Excitando o sistema com o motor superior desligado, obtêm-se os seguintes valores:

$$\xi = 0,011 \quad e \quad \omega_n = 16 \text{ rad/s},$$

$$c = 1,0798 \text{ N.s/m} \quad e \quad k = 785,33 \text{ N/m}.$$

Neste caso, devido à repetibilidade do experimento e independência das condições iniciais que têm esses parâmetros, a excitação pode ser aplicada manualmente de forma a se aproximar de um impulso.

O parâmetro restante, o coeficiente de amortecimento do motor superior (C), pode ser obtido excitando o motor superior e calculando a função de resposta em frequência (FRF) experimental da resposta. Ou seja, obtém as transformadas de Fourier da entrada e da saída e faz-se a razão entre elas, saída pela entrada.

O modelo produzido com os parâmetros mostrados não possui C determinado, varia-se o valor de C iterativamente até que haja concordância entre a FRF obtida pelo modelo e a obtida experimentalmente. A Figura 5.2 mostra as duas FRFs, considerando como entrada apenas o motor superior, em comparação para o ajuste do fator de amortecimento do motor ($C = 6.10^{-3}$ N.m.s).

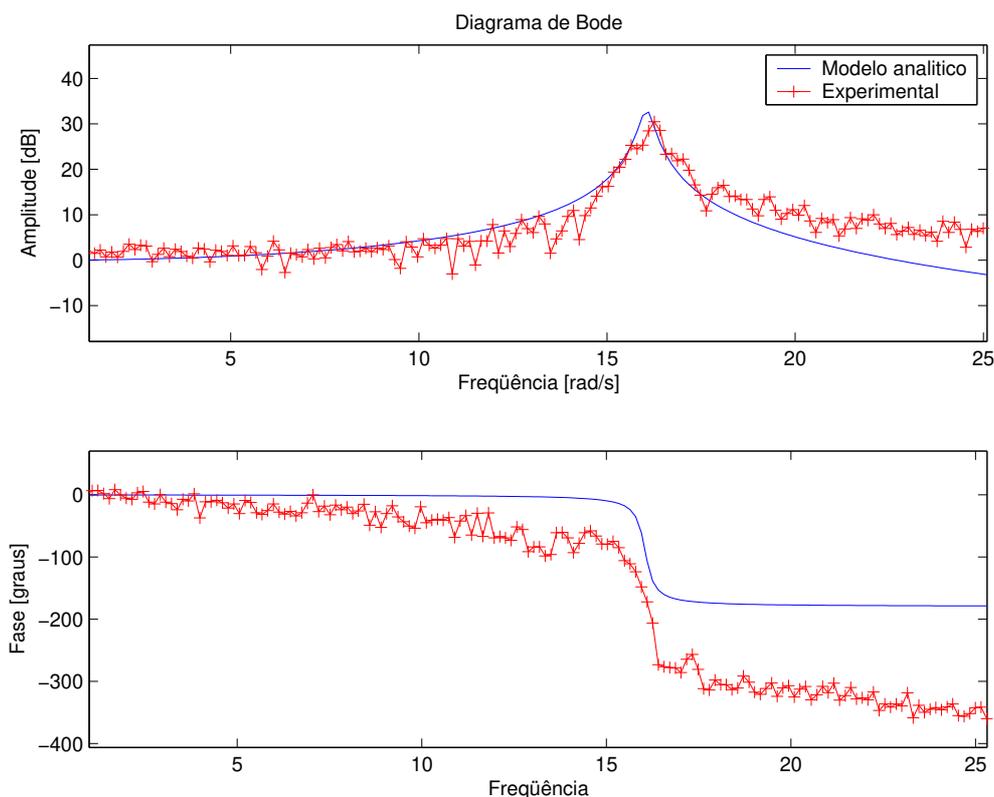


Figura 5.2: Comparação entre as FRFs experimental e a do modelo analítico com o coeficiente de amortecimento do motor ajustado

Os parâmetros para esse modelo são mostrados na Tabela 5.1 (a inércia do motor (J) e a sua constante mecânica (K) são informadas na placa do motor).

Tabela 5.1: Parâmetros do modelo analítico

m_1	=	2,7684 Kg
m_2	=	0,2993 Kg
p	=	0,003/(2 π) m/rad
K	=	0,131 N.m/A
J	=	10 ⁻⁶ kg.m ²
c	=	1,0798 N.s/m
k	=	785,33 N/m
C	=	6.10 ⁻³ N.m.s

Evidentemente, os instrumentos de medição utilizados possuem dinâmica própria que será desprezada. A relação entre a grandeza medida pelo extensômetro (deformação) e o sinal de tensão que ele produz é um fator descrito na seção anterior. Porém, esse fator foi calibrado experimentalmente e a relação deformação e tensão foi ajustada por uma reta com várias amostras. Essa calibração é necessária para obter os sinais produzidos pelos aparelhos em unidades físicas.

5.2 Identificação da bancada

A identificação experimental consiste em obter um modelo matemático que mais se aproxime, de acordo com algum critério de erro, da resposta real encontrada. É utilizada fazendo um cálculo de otimização sobre as entradas e saídas reais gravadas no sistema tendo os parâmetros de representação do modelo analítico como variáveis. Neste trabalho, uma vez que o sistema é suposto linear, um método de otimização que faça uso dessa hipótese é preferível.

O método utilizado basea-se em um processo iterativo de minimização das somas dos quadrados dos erros entre a resposta obtida pelo modelo candidato e a resposta efetivamente obtida. Esse já se encontrava implementado como uma função (`pem`) no pacote *System Identification* do programa Matlab (Ljung, 2004).

Todos os métodos de identificação encontrados na literatura para sistemas lineares, inclusive este utilizado, produzem representações no espaço de estados em tempo discreto. A teoria dos sistemas lineares pode ser escrita tanto em tempo contínuo quanto em tempo discreto (Ogata, 1990). Para o projeto dos controladores, os sistemas identificados serão convertidos para o domínio do tempo contínuo.

5.2.1 Sinal de Schroeder

Conforme será visto adiante, tanto para identificação quanto para os testes de controle o sinal de entrada utilizado foi o chamado sinal de Schroeder (Schroeder, 1970). Esse sinal possui vantagens sobre uma varredura senoidal simples porque possui o espectro em frequência plano até uma frequência estabelecida e nulo após ela. Assim, todas as frequências de zero até a desejada, são excitadas com igual potência. Ainda, o fator de pico é bastante baixo comparado com uma soma de senóides qualquer, conforme mostra a referência acima.

O sinal é obtido como uma soma de senos cujas fases são ajustadas para obter espectro plano conforme as expressões que seguem. Seja o sinal escrito como:

$$r(t) = \sum_{k=1}^N \left(\frac{p_k}{2}\right)^{\frac{1}{2}} \cos\left(\frac{2\pi kt}{T} + \phi_k\right);$$

em que $p_k = 1/N$, N é o número de componentes em frequência e as fases θ_k são obtidas a partir de uma fase inicial θ_1 como:

$$\theta_k = \theta_1 - \frac{\pi k^2}{N}.$$

Para os testes realizados neste trabalho o sinal de Schroeder foi produzido até uma frequência de 6,25 Hz. Esse sinal é ilustrado na Figura 5.3 e o módulo de sua transformada de Fourier, na Figura 5.4. A transformada de Fourier é calculada pelo algoritmo usual da Transformada Rápida de Fourier (FFT, do inglês Fast Fourier Transform).

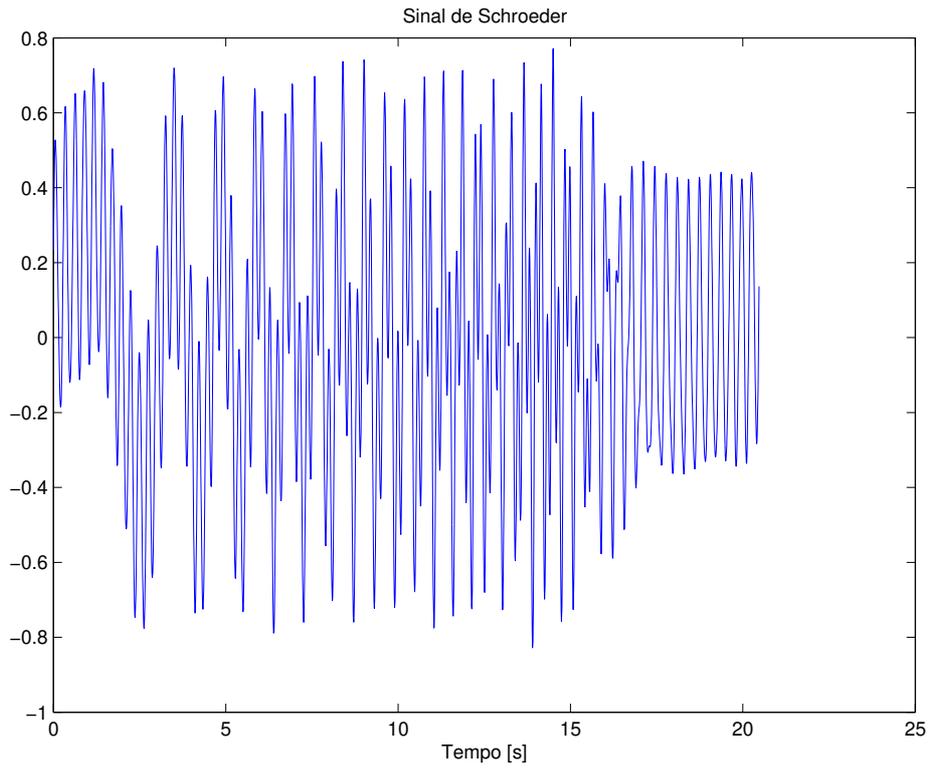


Figura 5.3: Sinal de Schroeder com espectro até 6,25 Hz

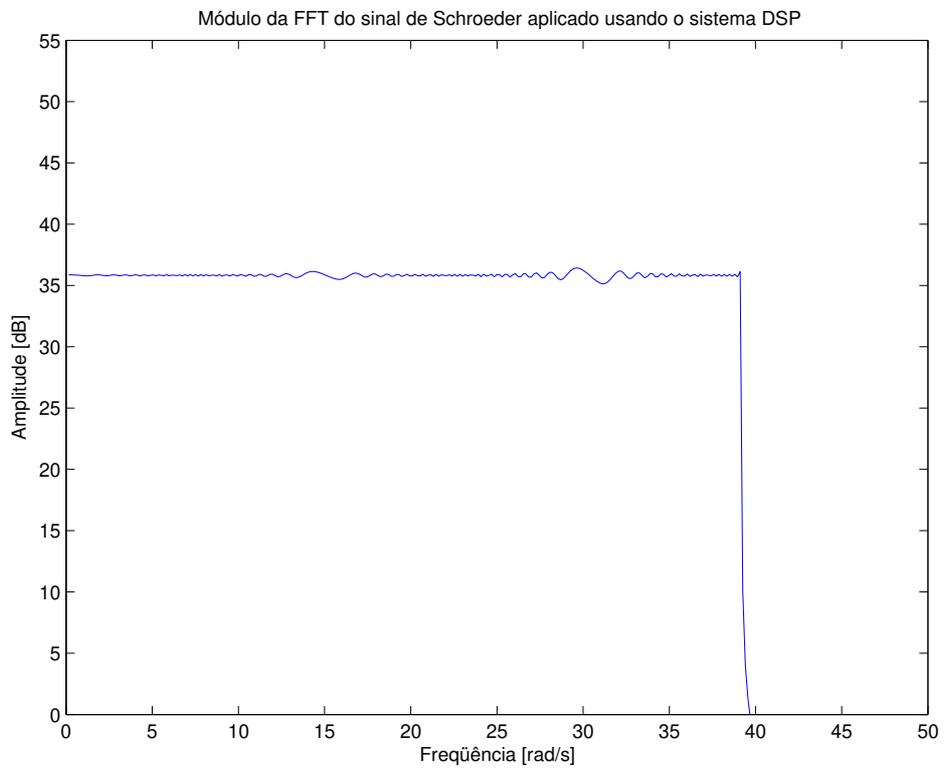


Figura 5.4: Módulo da FFT do sinal de Schroeder com espectro até 6,25 Hz

Foram seguidas duas abordagens para obter um modelo matemático que descrevesse a bancada. Uma delas analítica e a outra experimental. Esta última consiste em encontrar um modelo de estados para o sistema a partir de dados de entrada e saída obtidos na prática. O modelo utilizado para projetar os controladores foi o experimental. Porém, o modelo analítico foi utilizado para comparação.

5.2.2 Identificação para o sistema DSP

A identificação utilizando o sistema DSP é mais simples, pois o acionamento dos motores que causam o distúrbio é diretamente proporcional ao sinal de Schroeder produzido. Para esse caso, a resposta às excitações, utilizando os motores da base e o da massa m_2 , são mostradas nas Figuras 5.5 e 5.7, respectivamente. Os módulos das transformadas de Fourier para esses mesmos gráficos são mostrados em 5.6 e 5.8.

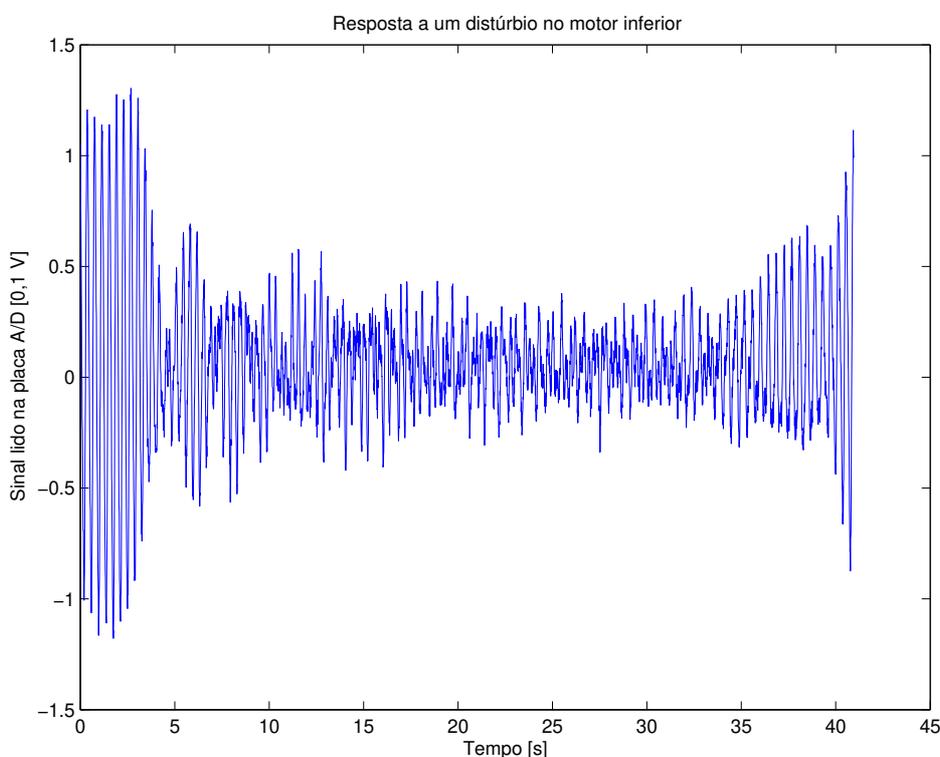


Figura 5.5: Resposta do sistema ao sinal de Schroeder aplicado ao motor da base

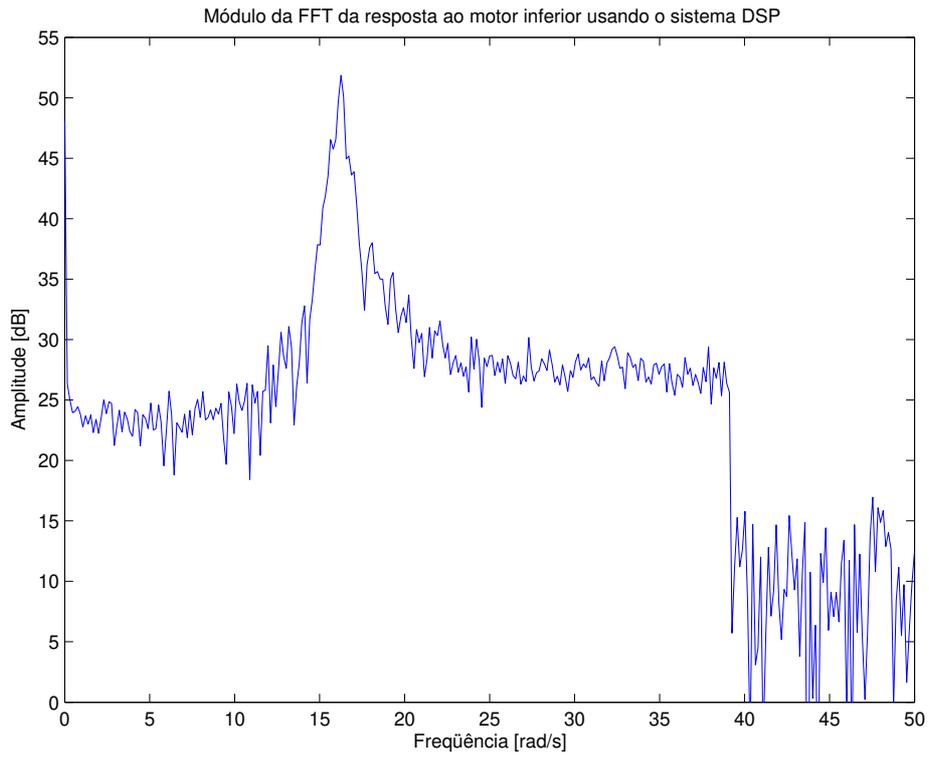


Figura 5.6: Módulo da FFT da resposta do sistema ao sinal de Schroeder aplicado ao motor da base

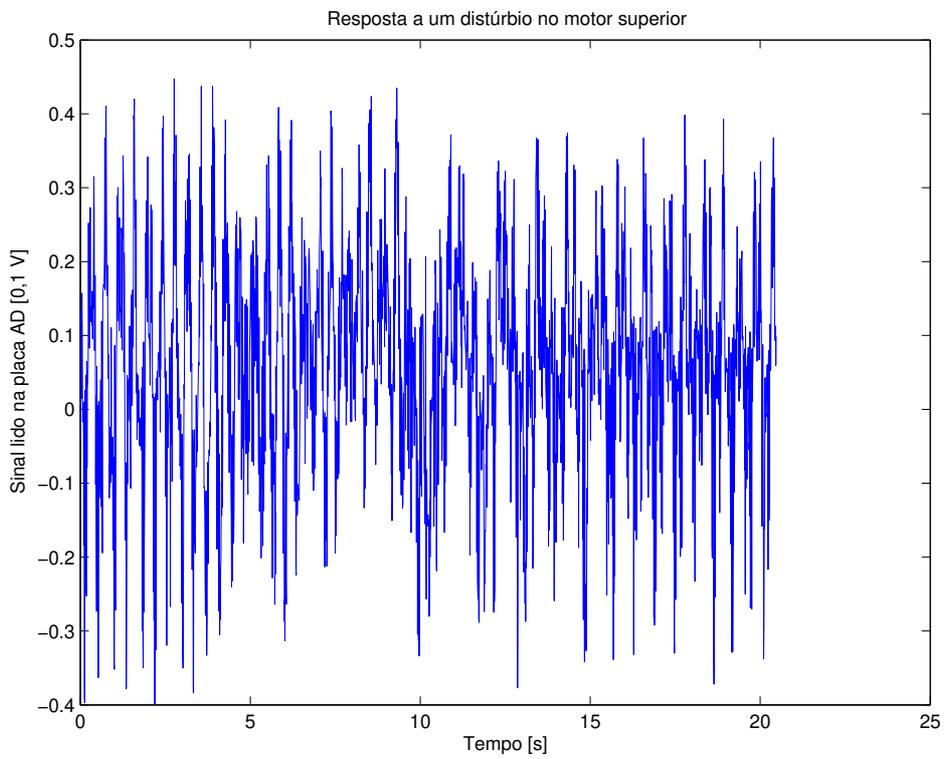


Figura 5.7: Resposta do sistema ao sinal de Schroeder aplicado ao motor superior

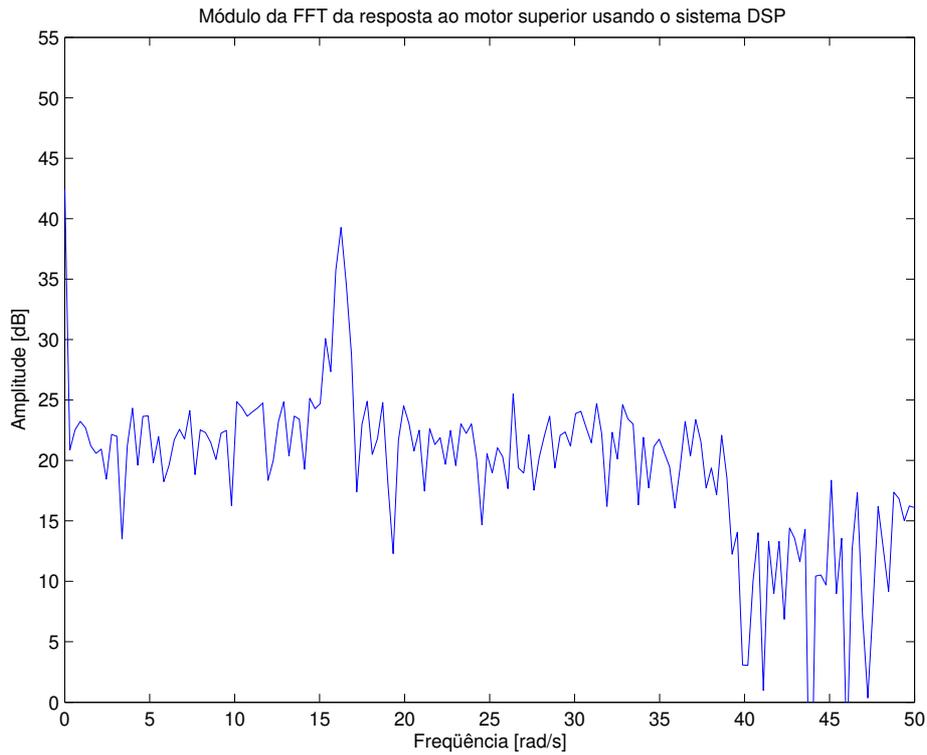


Figura 5.8: Módulo da FFT da resposta do sistema ao sinal de Schroeder aplicado ao motor superior

Conforme já dito, o Simulink permite programar a placa do kit DS1102 na forma de diagrama de blocos. Em linhas gerais, basta inserir um bloco correspondendo à excitação, outro para a conversão D/A, cujo canal de saída é determinado por qual entrada do bloco é usada e um terceiro para a conversão A/D, escolhendo similarmente a entrada e bloco de leitura do sinal convertido para a forma digital, conforme Figura 5.9

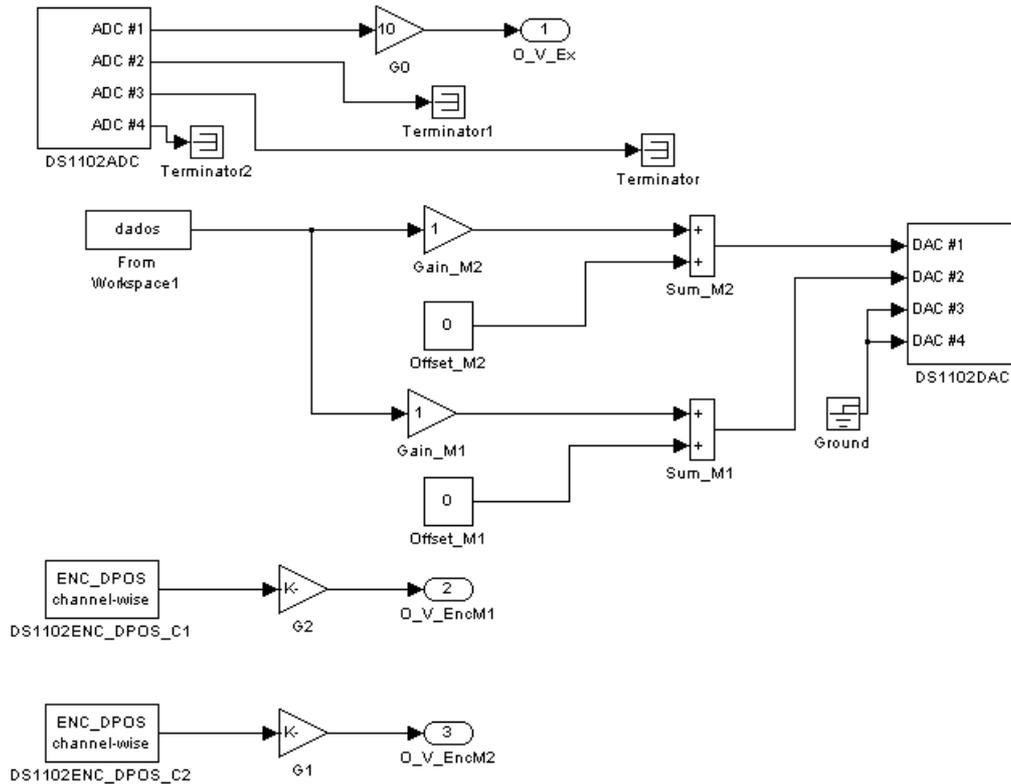


Figura 5.9: Diagrama de Simulink para identificar o sistema

Após a identificação, o modelo foi convertido para o tempo contínuo utilizando as fórmulas conhecidas de conversão (descritas em (Ogata, 1990) e implementadas no Matlab na função `d2c`).

O modelo contínuo obtido é mostrado nas matrizes a seguir:

$$A = \begin{bmatrix} -4.5799e-1 & 1.7415e1 & -9.7769e-1 \\ -1.5107e1 & 1.0013 & 1.1686e1 \\ 2.5553 & -1.5326e1 & -1.5573e2 \end{bmatrix},$$

$$B = \begin{bmatrix} 4.9029e-1 & 1.0667e2 \\ -3.2641 & -5.3002e2 \\ 4.4218e1 & 7.1989e3 \end{bmatrix},$$

$$C = [1.9681e1 \quad -4.6561 \quad 6.6412e-1] \text{ e}$$

$$D = [0 \quad 0].$$

É necessário notar que a primeira entrada corresponde à perturbação imposta ao sistema e a segunda, ao sinal de controle produzido pelos controladores explicitados adiante. Nesse modelo é encontrada a frequência de pico de oscilação de 16 rad/s.

O diagrama de valor singular associado a essas matrizes é mostrado na Figura 5.10.

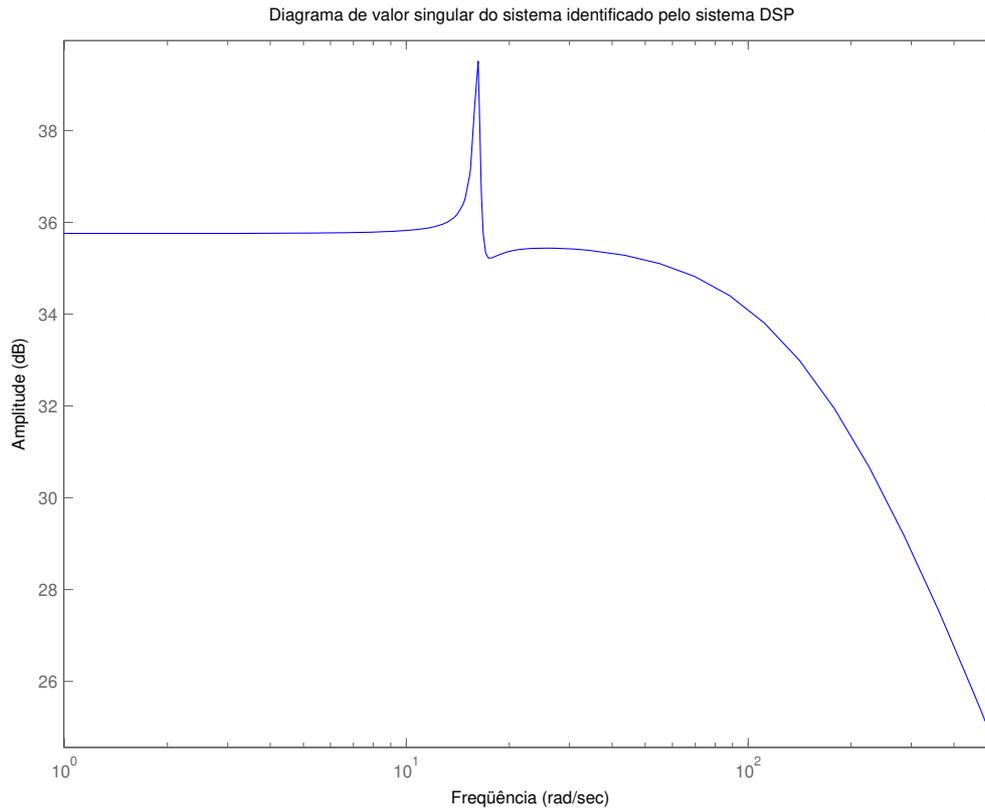


Figura 5.10: Diagrama de valor singular do modelo identificado para o sistema

5.2.3 Identificação para o sistema embarcado

Para o controle usando o sistema embarcado, devido ao uso de uma saída de controle modulada em largura de pulso, foi realizada uma nova identificação, também utilizando o sistema dSPACE.

A mesma identificação anterior pode ser conduzida para obter um modelo que será utilizado para controle a partir do sistema embarcado. A diferença mais relevante aqui é que o controle usando o microcontrolador pressupõe uma excitação usando um sinal em PWM. Assim, é preciso converter o sinal de Schroeder para essa representação, o que significa produzir um sinal de largura de pulso com ciclo de trabalho (razão entre tempo de tensão alta e tempo total em um período) proporcional ao sinal de distúrbio.

O diagrama de Simulink usado para programar a placa dSPACE é bastante semelhante ao anterior com a diferença de que é preciso acrescentar blocos que convertem a excitação contínua em um sinal modulado em largura de pulso. Como se ilustra na Figura 5.11, o primeiro canal de conversão recebe o próprio sinal de distúrbio contínuo.

No segundo, faz-se o seguinte cálculo: o sinal de distúrbio é multiplicado por 100; uma constante de escala que, para um sinal de distúrbio de amplitude máxima 1, produz um valor de no máximo

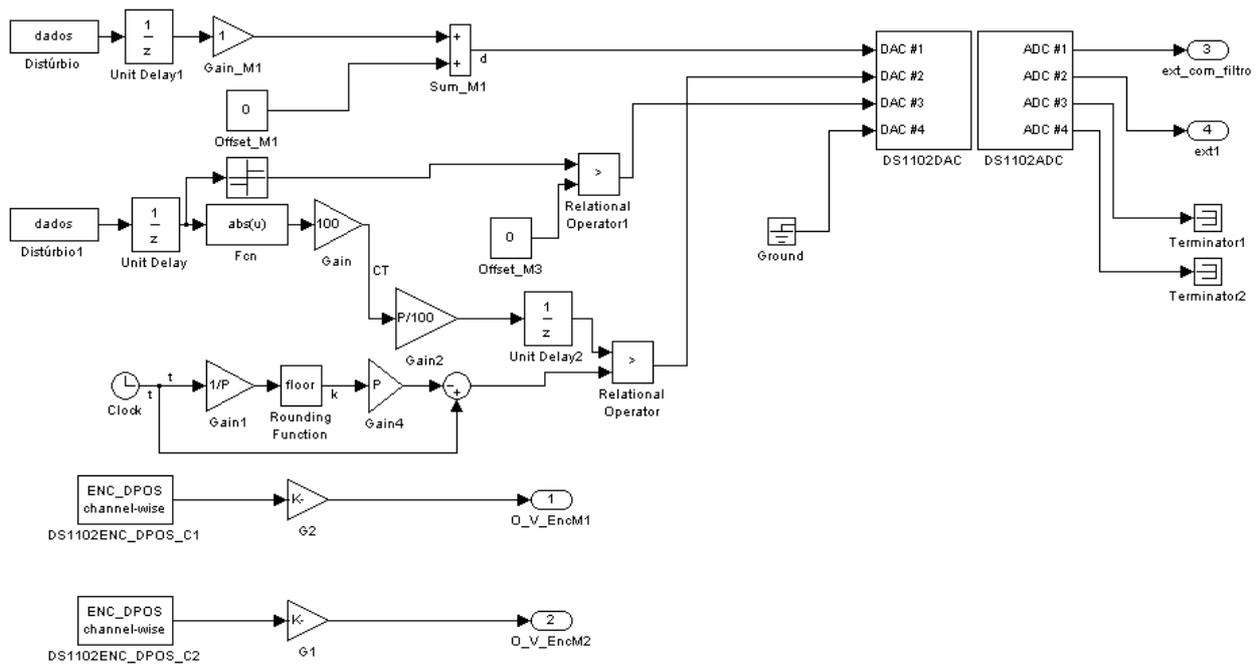


Figura 5.11: Diagrama de Simulink para identificação com o sinal em PWM

100, que corresponde ao maior ciclo de trabalho possível. Esse valor é dividido por 100 (agora apenas para obter uma relação de 0 a 1, foi multiplicado por 100 antes para ter-se disponível um sinal no diagrama que fosse o próprio ciclo de trabalho). O sinal de distúrbio é produzido a uma taxa de amostragem menor do que o tempo fixo de execução do diagrama de blocos mostrado. Por isso, são colocados blocos de atraso que mantêm o valor do sinal enquanto a parcela mais rápida do diagrama é executada. O valor do tempo de execução é comparado com o tempo de duração associado ao ciclo de trabalho e se for menor, coloca-se a saída em nível alto e em nível baixo, caso contrário.

No terceiro canal de saída, coloca-se simplesmente o sinal (positivo ou negativo) associado ao segundo canal na forma de nível lógico alto ou baixo.

Para a identificação, o sinal de distúrbio na forma de PWM foi produzido no Simulink para emular o comportamento do sistema embarcado, pois de outra forma seria necessário produzir o sinal de distúrbio dentro do próprio microcontrolador ou enviá-lo a partir do Simulink usando a porta serial. Ou seja, permanece-se usando a placa de conversão D/A para produzir o distúrbio. Os dados são obtidos usando a placa A/D. Para excitar o motor superior, utiliza-se o sinal de PWM produzido na placa D/A para ativar o circuito de acionamento.

Numa configuração final de controle sem o sistema DSP, o monitoramento do controle pelo computador precisa ser feito de forma indireta, utilizando a porta serial da placa MSC1210 para comunicação com o computador. Para detalhes da transmissão serial, consultar o Apêndice C.

A resposta para essa configuração é mostrada na Figura 5.12 e o módulo de sua transformada de Fourier, na Figura 5.13.

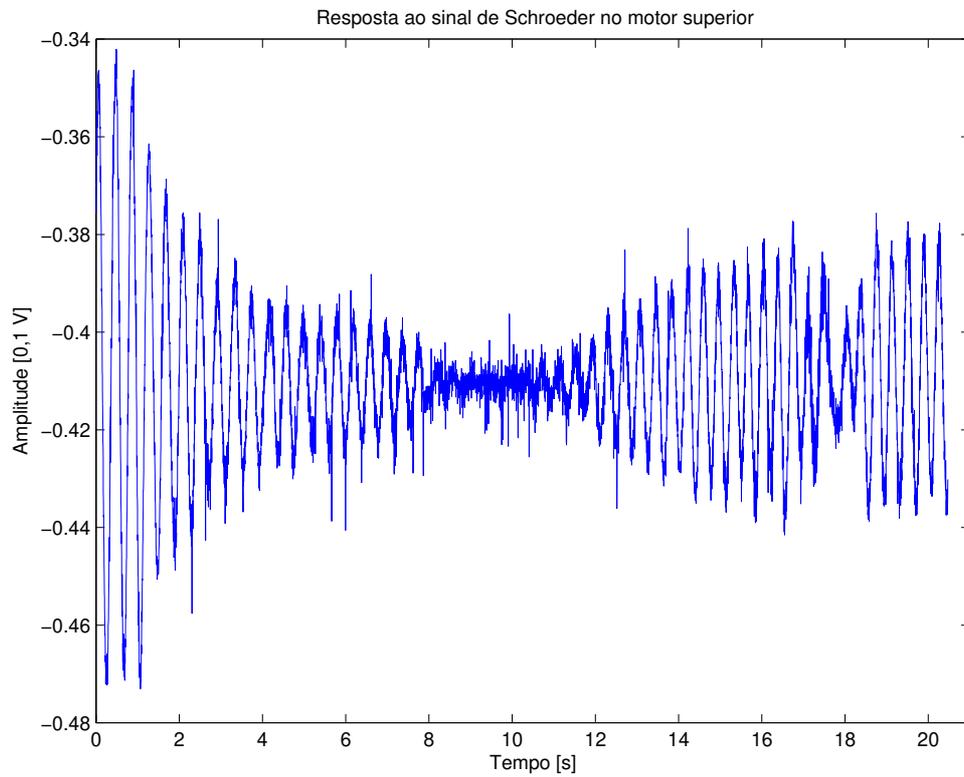


Figura 5.12: Resposta do sistema ao sinal de Schroeder como distúrbio no motor superior usando as características do sistema embarcado

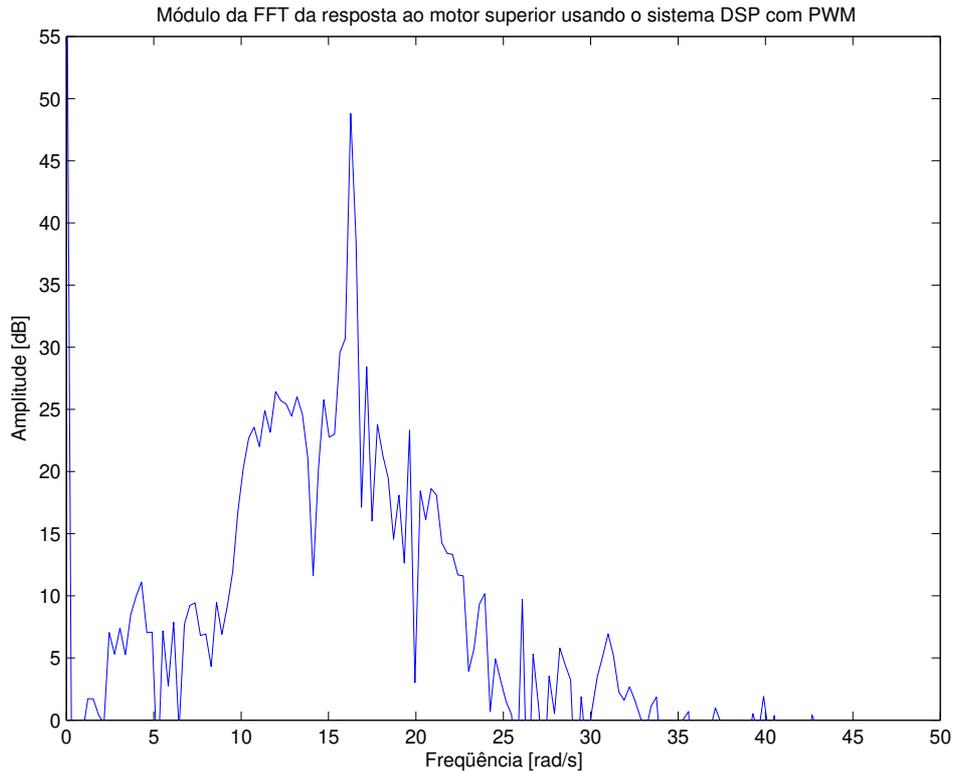


Figura 5.13: Módulo da FFT da resposta do sistema ao sinal de Schroeder como distúrbio no motor superior usando as características do sistema embarcado

A excitação do motor superior utilizando um sinal em PWM envolve a dinâmica do motor no seguinte sentido: o sinal em PWM representa variação do tempo de energização do motor. O pico de ressonância também é visível na Figura 5.12, em torno de 0,4 segundos.

Esses dados permitem encontrar o modelo descrito pelas matrizes a seguir:

$$A = \begin{bmatrix} -0.0004669 & 0 & 0 \\ 0 & -0.3206 & 16.44 \\ 0 & -16.44 & -0.3206 \end{bmatrix},$$

$$B = \begin{bmatrix} 9.621e - 005 & 5.488e - 006 \\ -1.191 & -0.5781 \\ 0.1675 & -0.1513 \end{bmatrix},$$

$$C = [34.39 \quad -0.4416 \quad 0.2546] \text{ e}$$

$$D = [0 \quad 0].$$

5.3 Controladores H_2 e H_∞

Um dos objetivos da construção da bancada é o de teste didático de controladores implementados com o modelo da estrutura, anteriormente descrito. Tendo em vista esse fim, foram testados inicialmente

dois tipos de controladores: um deles projetado para minimizar a norma H_2 e o outro para a norma H_∞ .

As normas H_2 e H_∞ são critérios de projeto bastante utilizados atualmente para o projeto de controladores. São mostradas a seguir suas definições.

A norma H_2 de uma matriz de transferência $G(S)$ é dada por

$$\|G(s)\|_2 = \sqrt{\frac{1}{2\pi} \int_{-\infty}^{\infty} \text{tr}(G(j\omega)^*G(j\omega))d\omega}$$

em que $\text{tr}(\cdot)$ é o traço da matriz argumento e $*$ é a transposta conjugada complexa.

A norma H_∞ , por sua vez é:

$$\|G(s)\|_\infty = \max_{\omega} \bar{\sigma}(G(j\omega)),$$

em que $\bar{\sigma}(\cdot)$ é o maior valor singular da matriz argumento.

Essas normas representam requisitos diferentes. Minimizar a norma H_2 significa reduzir toda a curva de resposta em frequência, ao passo que diminuir a norma H_∞ implica redução apenas no valor máximo da citada resposta, que corresponde também ao ganho de energia, como em (Skogestad and Postlethwaite, 1997). Elas são utilizadas com objetivos de engenharia diferentes, não abordados aqui.

A abordagem utilizada aqui e mais comumente na literatura é determinar um controlador dinâmico que, em malha fechada com a planta a ser controlada, reduza uma norma escolhida. O *Matlab* possui funções para projeto de controladores com esses critérios, quais sejam: `h2lqg` e `hinf` (conforme (Packard et al., 2004)). Assim, essas funções produzem as quatro matrizes de estado A , B , C e D que representam o controlador.

Para a implementação, os controladores devem ser discretos. Assim, os controladores obtidos precisam ser discretizados usando as fórmulas conhecidas. Foi adotada neste trabalho a discretização do segurador de ordem zero, implementada na função `c2d`. Para referência sobre discretização, ver (Ogata, 1990).

5.4 Controle usando o sistema DSP

5.4.1 Implementação dos controladores

Fazendo uso do sistema DSP, foram implementados os controladores encontrados com o auxílio do *Matlab*. Isso é feito por meio do *Simulink* (um módulo do *Matlab* utilizado para programar a placa de aquisição) e do *Control Desk* (aplicativo fabricado pela *dSPACE* para operar aquisição e controle).

O programa em blocos mostrado na Figura 5.14 é o desenvolvido para operar os dados da planta em questão.

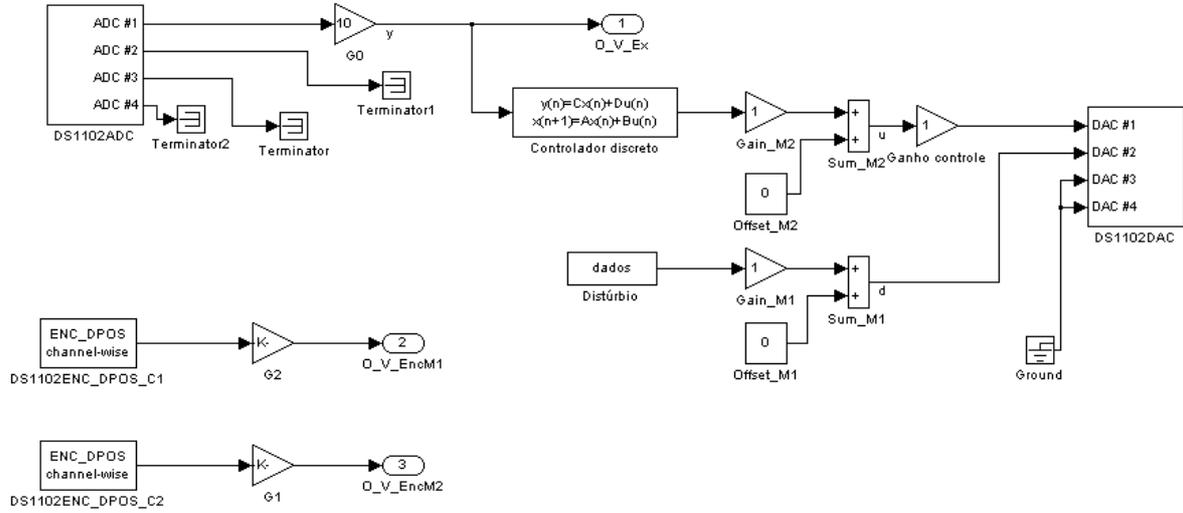


Figura 5.14: Programa utilizando Simulink

O sinal de distúrbio utilizado, sobre o qual se deve realizar o controle, foi o mesmo utilizado para a identificação: o sinal de Schroeder com espectro até 6,25 Hz.

Os códigos de Matlab usados para calcular as matrizes dos controladores são fornecidos no Apêndice B. As matrizes no modo contínuo para o H_2 e o H_∞ , respectivamente, são mostradas a seguir:

Para o controlador H_2 ,

$$A = \begin{bmatrix} -2.1136e3 & 5.1733e2 & -7.2284e1 \\ 1.0480e4 & -2.4819e3 & 3.6583e2 \\ -1.4253e5 & 3.3705e4 & -4.9654e3 \end{bmatrix},$$

$$B = \begin{bmatrix} 6.9691e - 1 \\ -3.2371 \\ 4.3292e1 \end{bmatrix},$$

$$C = [-1.9681e1 \quad 4.6561 \quad -6.6412e - 1] \text{ e}$$

$$D = [0].$$

Para o controlador H_∞ (a função usada para o projeto, `hinf`, simplifica a representação de estados do controlador podendo reduzir a sua ordem conforme critérios próprios, (Mathworks, 2004)),

$$A = \begin{bmatrix} -2.8864e2 & -2.8321e3 \\ -4.8637e1 & -4.4688e3 \end{bmatrix},$$

$$B = \begin{bmatrix} -1.9753e - 12 \\ -3.4177e - 12 \end{bmatrix},$$

$$C = \begin{bmatrix} -1.1208e1 & -1.0995e2 \end{bmatrix} \text{ e}$$

$$D = \begin{bmatrix} -1 \end{bmatrix}.$$

5.4.2 Resultados

As Figuras 5.15 e 5.16 a seguir mostram os resultados em termos da resposta obtida antes e depois do controle orientado respectivamente para as normas H_2 e H_∞ . Assim como para a identificação, o controle foi realizado com o sistema DSP.

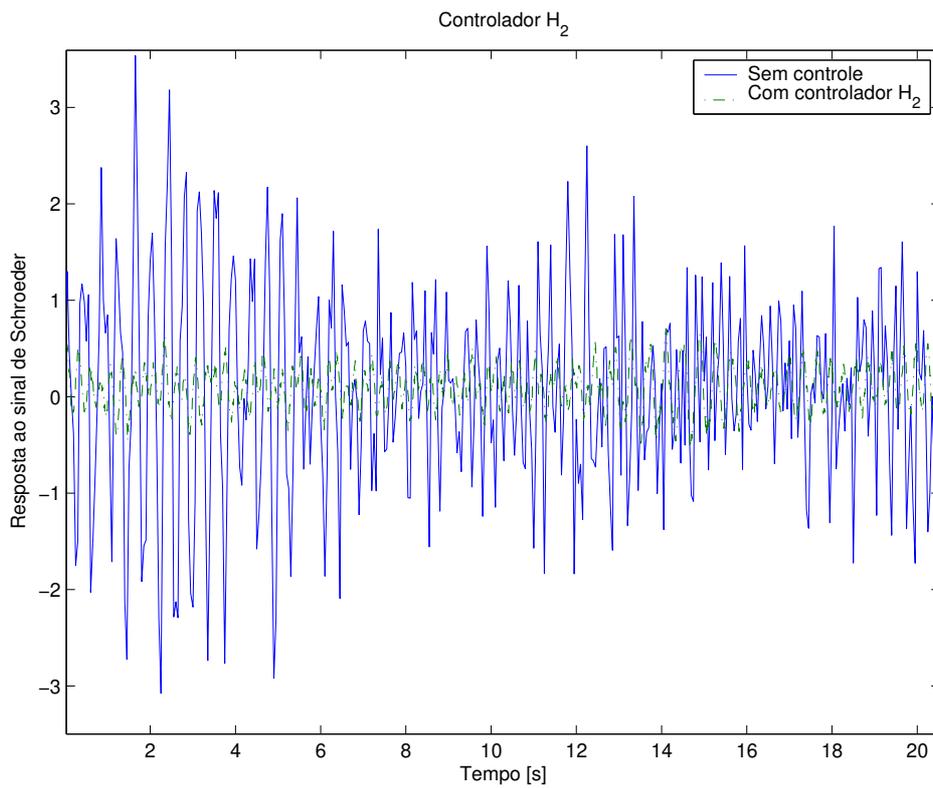


Figura 5.15: Resposta no tempo com controlador H_2

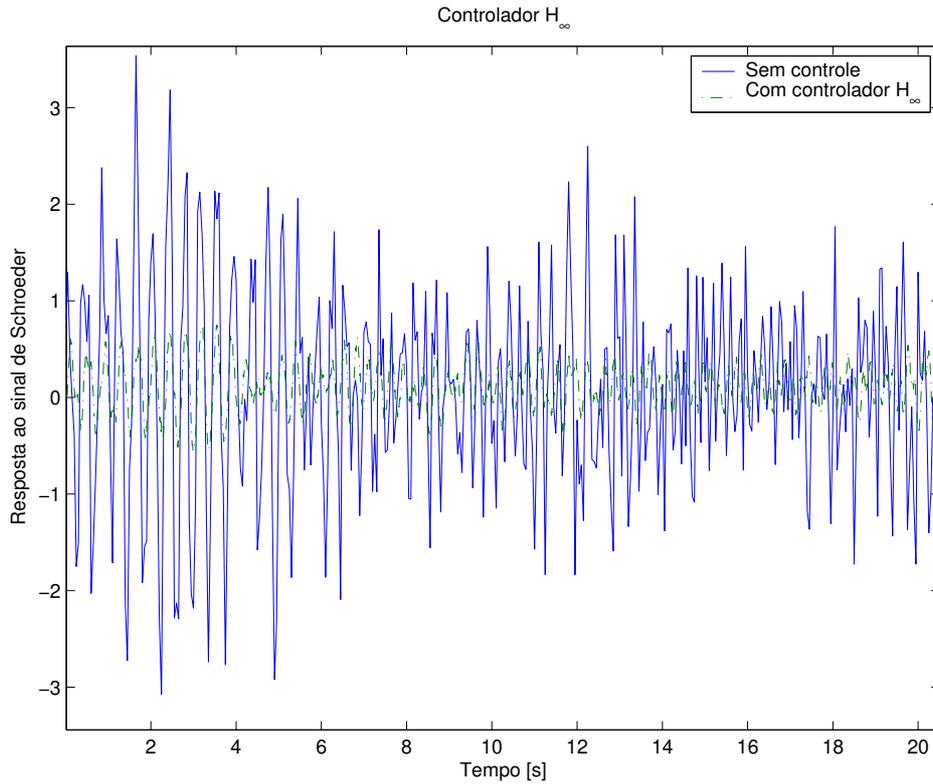


Figura 5.16: Resposta no tempo com controlador H_∞

Foram traçadas também as transformadas de Fourier (FFTs) do sistema original em comparação com as suas versões controladas. É necessário lembrar que todos os testes foram realizados com um sinal de Schroeder de espectro até 6,25 Hz como distúrbio para o sistema. Cada FFT mostrada na Figura 5.17 foi obtida com uma média de três FFTs de sinais experimentais. Esse gráfico mostra uma atenuação de 12 dB no valor da ressonância para o controlador H_2 e de 9 dB para o controlador H_∞ .

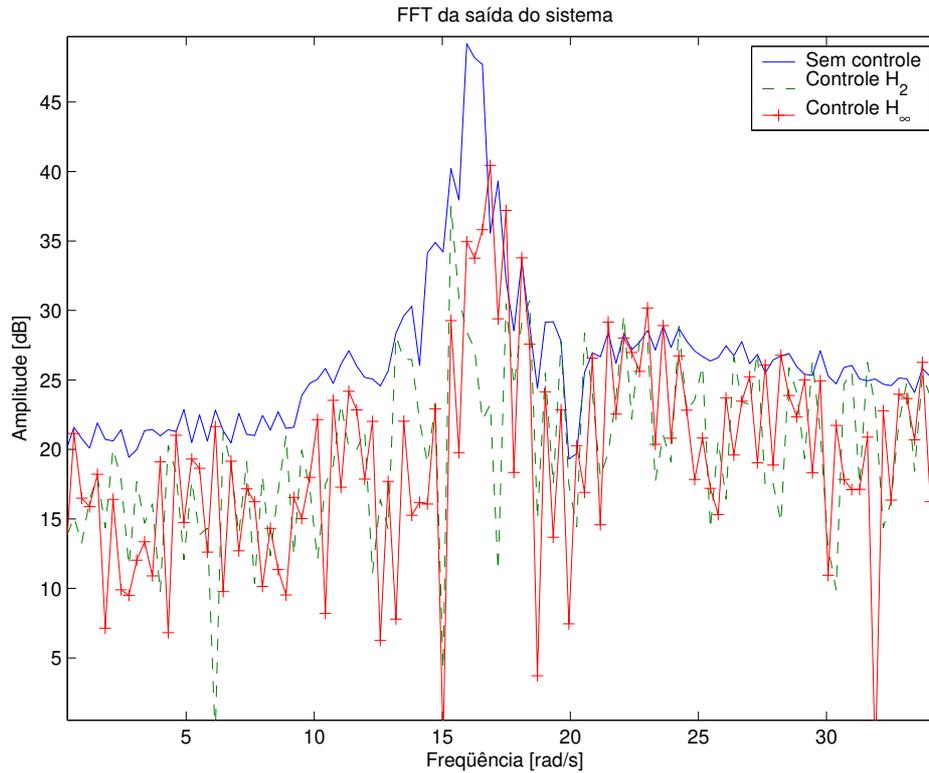


Figura 5.17: FFTs do sistema e das versões com controlador, respectivamente

O fato de o controlador H_{∞} ter promovido uma redução do pico menor do que o H_2 deve-se possivelmente ao fato de o sistema incluir outros modos de vibração não modelados.

5.5 Controle usando o sistema embarcado

5.5.1 Implementação dos controladores

O projeto dos controladores foi realizado usando o Matlab pelos mesmos comandos utilizando, porém, as matrizes do sistema identificado com a segunda configuração indicada nos Capítulos 3 e 5.

Os controladores obtidos pelos mesmos métodos usados para o caso com o sistema DSP têm as matrizes mostradas a seguir:

Para o controlador H_2 ,

$$A = \begin{bmatrix} -0.0040 & 0.0000 & -0.0000 \\ 60.8393 & -1.1020 & 16.8945 \\ -0.5555 & -16.4368 & -0.3247 \end{bmatrix},$$

$$B = \begin{bmatrix} 0.0001 \\ -1.1912 \\ 0.1675 \end{bmatrix},$$

$$C = [-34.3862 \quad 0.4416 \quad -0.2546] \text{ e}$$

$$D = [0].$$

Para o controlador H_∞ ,

$$A = \begin{bmatrix} -0.0040 & 0.0000 & -0.0000 \\ 60.8393 & -1.1020 & 16.8945 \\ -0.5555 & -16.4368 & -0.3247 \end{bmatrix},$$

$$B = \begin{bmatrix} 0.0001 \\ -1.1912 \\ 0.1675 \end{bmatrix},$$

$$C = [-34.3862 \quad 0.4416 \quad -0.2546] \text{ e}$$

$$D = [0].$$

Nota-se que as duas funções do Matlab para projeto dos controladores H_2 e H_∞ produziram os mesmos controladores para esse caso.

5.5.2 Resultados

Na configuração final de controle, o sistema embarcado implementa o mesmo controlador projetado e utilizado na Seção 5.6. O controlador é convertido para código em linguagem C e, conforme o programa mostrado no Apêndice A, o sinal de controle produz os sinais em PWM e de direção utilizados para acionar o circuito com ponte H para mover o motor que realiza o esforço de controle (circuito mostrado na Figura 3.15).

O resultado de controle obtido dessa forma, analogamente à seção anterior, é mostrado na forma do módulo da FFT antes e depois da atuação do controle (Figura 5.18).

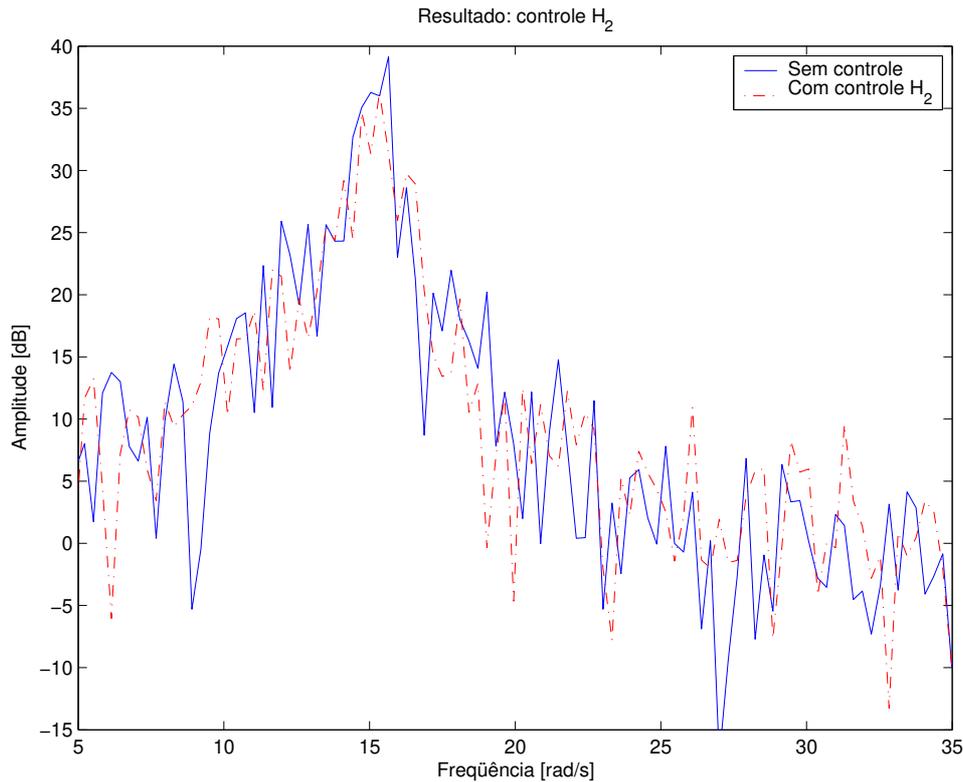


Figura 5.18: FFTs do sistema e da versão com controlador na placa MSC1210, respectivamente

Nota-se que a freqüência de maior oscilação permanece 16 rad/s e a atenuação com controle foi de aproximadamente 3 dB.

5.6 Resultados utilizando o sistema DSP para produzir o sinal em PWM

Para efeito de comparação, foi realizado um novo teste, no qual os controladores com saída modulada em largura de pulso foram implementados com a placa dSPACE, exatamente como deve funcionar no microcontrolador. Para este caso também, os controladores são encontrados usando o programa do Apêndice B. Nessa configuração de controle, foi implementado um dos controladores H_2 ou H_∞ , uma vez que as matrizes de ambos produzidas pelos algoritmos de projeto foram idênticas. O resultado do controle pode ser observado no domínio da freqüência, analogamente ao que foi traçado para o controle usando o DSP sem a conversão do sinal em PWM (seção anterior), na Figura 5.19. O gráfico mostra uma atenuação de 5 dB no valor da ressonância.

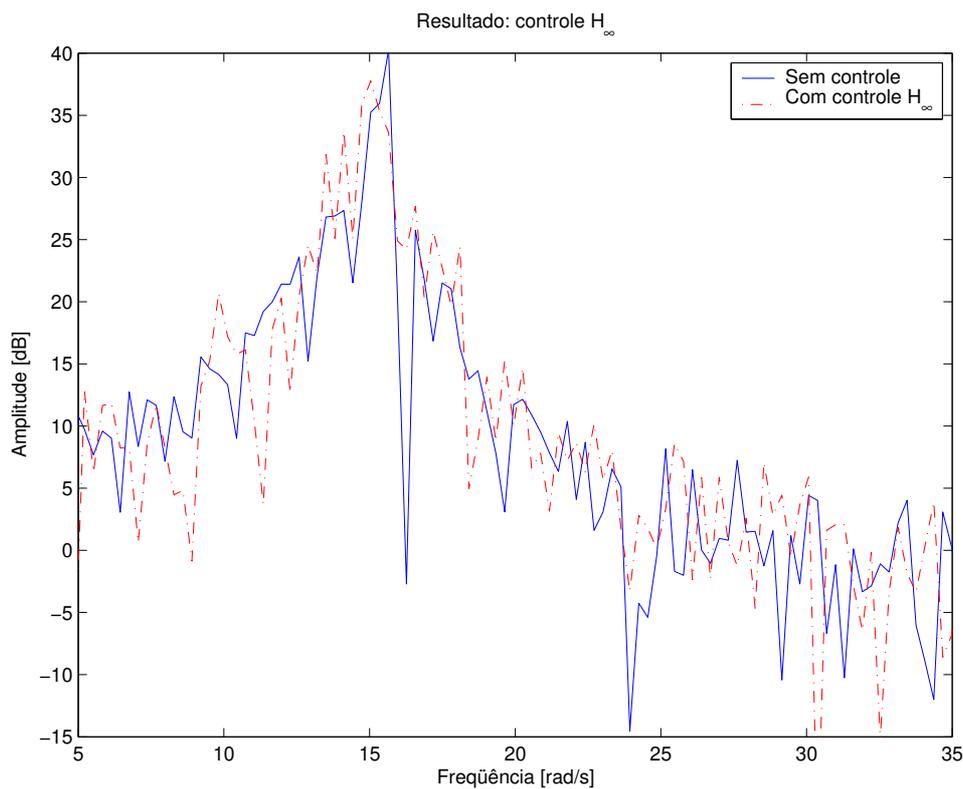


Figura 5.19: FFTs do sistema e da versão com controlador usando o sistema DSP, respectivamente

O arquivo de Simulink para realizar o controle desta maneira, mostrado na Figura 5.20, é bastante similar ao usado para identificação com as características do sistema embarcado (Figura 5.11). A diferença está no fato de que a saída da planta, lida dos extensômetros na placa A/D, é transferida a um controlador. O sinal de controle obtido é substituído como entrada do diagrama que converte um sinal para PWM, usado anteriormente para sinal de Schroeder.

Capítulo 6

Conclusão e propostas para trabalhos futuros

6.1 Conclusão

O objetivo principal deste trabalho foi o desenvolvimento de uma bancada didático-experimental de baixo custo a ser utilizada nas áreas de controle de sistemas mecânicos, processamento de sinais, identificação de sistemas, projeto de sistemas embarcados, entre outras. O trabalho foi concretizado através de alguns objetivos secundários, listados a seguir: desenvolvimento da parte eletrônica que processa analogicamente os sinais de medição e de atuação; a programação dos módulos de processamento digital desses sinais; e o teste dos controladores projetados para a validação da integração do sistema.

A parte eletrônica pode ser subdividida no desenvolvimento de um acionador para o motor de distúrbio, e os acionadores para o motor de controle. Envolve ainda o circuito de amplificação e filtragem para a leitura da deformação da estrutura devido à sua vibração. O acionador para o motor de distúrbio é um circuito comum ao sistema DSP e ao embarcado, e consiste em um amplificador por realimentação de tensão que fornece potência para o sinal gerado como um distúrbio externo. Por outro lado, os acionadores do motor de controle são distintos, tendo sido desenvolvidos um amplificador realimentado em corrente para o sistema DSP e um circuito para modular por largura de pulso o sinal de controle resultante dos algoritmos de controle. Os circuitos desenvolvidos apresentaram bom desempenho, podendo ser considerados adequados para o propósito do trabalho. Ficou evidenciado também que o circuito de potência usando componentes discretos é uma boa opção para reduzir o custo total da montagem. O sinal de deformação, que constituía a saída do sistema, é tratado por um amplificador de instrumentação e um filtro analógico passivo de componentes discretos. Os resultados obtidos mostraram que os sinais assim tratados foram adequados para a aplicação, em termos do nível de amplitude e da relação sinal/ruído.

Quanto ao aspecto das programações desenvolvidas, para o sistema DSP o trabalho foi realizado essencialmente através da construção de diagramas de blocos utilizando o Simulink. Foram implementados os diagramas para a identificação do modelo usando como entrada o sinal de distúrbio e a saída medida, e ainda para a entrada de controle e a mesma saída. Tal modelo foi usado para o projeto dos controladores usando o sistema DSP. Para uso com o sistema embarcado, foi incluído um módulo para geração do sinal PWM a partir do sinal de Schroeder, e realizada uma nova identificação que, utilizando o sistema DSP, foi adotada para o projeto embarcado. Também foi usada a placa da dSPACE para implementar e testar o controle com PWM projetado para o sistema embarcado. Finalmente, foram programados e testados os controladores H_2 e H_∞ para ambas as configurações. Ainda quanto à programação, foi implementado o sistema de controle em tempo real para o sistema embarcado em C, o que exigiu naturalmente muito mais tempo do que a elaboração dos diagramas de bloco para o Simulink, e foram realizados os testes dos controladores.

A identificação e mesmo o controle utilizando o sistema DSP são mais simples devido ao uso da programação visual permitida pelo Simulink, o que o habilita como uma ferramenta didática mais versátil. No entanto, o seu custo pode limitar a sua utilização.

A programação da placa MSC1210, realizada em linguagem C, apresentou bons resultados apesar da enorme diferença de custo quando comparada à placa da dSPACE. A linguagem de alto nível provou-se uma ferramenta eficaz para o desenvolvimento de aplicações de controle para sistemas embarcados, viabilizando a manipulação de vetores e matrizes, bem como operações com ponto flutuante, o que seria muito trabalhoso ser conseguido apenas com linguagem de montagem. O uso das ferramentas de programação em tempo real, impondo devido a sua sistemática, a elaboração de um projeto consistente, simplificou o trabalho de modo geral.

Os testes dos controladores, os quais apresentaram bom desempenho do sistema controlado com reduções do pico de ressonância da ordem de 10 dB, validam a utilização da montagem proposta como bancada experimental a ser usada em laboratórios de disciplinas de introdução aos métodos de controle, e indicam ainda que pode ser aplicada ao ensino mais avançado do controle de estruturas mecânicas flexíveis.

6.2 Propostas para trabalhos futuros

Como propostas futuras para a melhoria deste trabalho, segue:

- o desenvolvimento de uma interface que permita inferir e alterar, em tempo real, parâmetros do sistema;

- o aperfeiçoamento dos circuitos de proteção para evitar acidentes;
- o projeto de uma placa de circuito impresso para acondicionar todo aparato eletrônico;
- a integração de todos os módulos de modo a facilitar o manuseio e melhorar a aparência do equipamento.

Referências

- Almeida, P. A. O. (1996). Extensometria elétrica iii. Technical report, Escola Politécnica da Universidade de São Paulo. Notas de Aula.
- Amorim, C. R. (2005). Contrôle actif d'un édifice soumis à un tremblement de terre. Trabalho de Graduação.
- Bannatyne, R. and Viot, G. (1998). Introduction to microcontrollers - part 2. In *Proceedings of the IEEE Conference Northcon*, pages 250 – 254.
- Beetner, D., Pottinger, H., and kyle Mitchell (2000). Laboratories teaching concepts in microcontrollers and hardware-software co-design. In *30th ASEE/IEEE Frontiers in Education Conference*, volume 2, pages S1C/1 – S1C/5.
- Belina, F., Hogrefe, D., and Sarma, A. (1991). *SDL with applications from protocol specification*. Prentice-Hall.
- Brglez, F., Bryan, D., and Koiminski, K. (1989). Combinational profiles of sequential benchmark circuits. In *IEEE International Symposium on Circuits and Systems*, volume 3, pages 1929 – 1934.
- Caicedo, J. M., Dyke, S. J., Moon, S. J., Bergman, L. A., Turan, G., and Hague, S. (2003). Phase ii benchmark control problem for seismic response of cable-stayed bridges. *Journal of Structural Control*, 10(3-4):137–168.
- Chen, B. M., Saberi, A., Sannuti, P., and Shamash, Y. (1993). Construction and parameterization of all static and dynamic h₂-optimal state feedback solutions, optimal fixed modes, and fixed decoupling zeros. *IEEE Transactions on Automatic Control*, 38(2):248–261.
- Devices, A. (1999). Precision instrumentation amplifier - ad524. DataSheet.

Encarta, M. (2005). English - english dictionary.

Available: <http://encarta.msn.com/encnet/features/dictionary/dictionaryhome.aspx>.

Farines, J.-M., da S. Fraga, J., and de Oliveira, R. S. (2000). *Sistemas de tempo real*. IME - USP.

Ferreira, L. O. S. (2004). Laboratório de sistemas digitais. Notas de Aula.

Gonzalez, F. G. and Davis, W. J. (1997a). A simulation-based controller for a flexible manufacturing cell. In *Proceedings of the 1997 Winter Simulation Conference*, volume 4, pages 3642 – 3647.

Gonzalez, F. G. and Davis, W. J. (1997b). A simulation-based controller for distributed discrete-event systems with application to flexible manufacturing. In *Proceedings of the 1997 Winter Simulation Conference*, pages 845 – 852.

Gonzalez, F. G., Helton, A., Helton, D., Smith, J., Thompson, E., and Walterscheid, G. (2000). The design of a solid-state physical model of an automatied system to be used as a test bed for control applications. In *Proceedings of the 2000 Winter Simulation Conference*, volume 1, pages 900 – 908.

Heck, B. S., Clements, N., and Ferri, A. A. (2004). A lego experiment for embedded control system design. *IEEE Control Systems Magazine*, 24(5):61 – 64.

Klassner, F. and Anderson, S. D. (2003). Lego mindstorms: Not just for k-12 anymore. *IEEE Robotics & Automation Magazine*, 10(2):12–18.

Lee, C., Kirovski, D., Hong, I., and Potkonjak, M. (1997). Dsp quant: Design, validation, and applications of dsp hard real-time benchmark. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 679 – 682.

Ljung, L. (2004). *System Identification Toolbox: User's Guide*. MathWorks, Inc.

Lu, L.-T., Chiang, W.-L., Tang, J.-P., Liu, M.-Y., and Chen, C.-W. (2003). Active control for a benchmark building under wind excitations. *Journal of Wind Engineering and Industrial Aerodynamics*, 91(4):469–493.

Macário, C. G. N. (2003). Uso da linguagem de especificação sdl como alternativa ao diagrama de estados proposto pela linguagem uml.

Available: <http://www.cnptia.embrapa.br/modules/tinycontent3/content/2003/comtec55.pdf>.

- Mathworks, I. (2004). *Robust Control Toolbox: User's guide*.
- Ogata, K. (1990). *Discrete-time control systems*. Prentice Hall.
- Packard, A., Balas, G., Safonov, M., and Chiang, R. (2004). *Robust Control Toolbox: User's Guide*.
Mathworks, Inc.
- Paganini, F. (1995). Necessary and sufficient conditions for robust h_2 performance. In *Proceedings of the IEEE Conference on Decision and Control*, volume 2, pages 1970 – 1975.
- Sarracini, F. J. (2004). Controle de estruturas flexíveis do ponto de vista h_∞ utilizando a redução de modelo de guyan. Trabalho de graduação, Faculdade de Engenharia Mecânica - Universidade Estadual de Campinas.
- Schroeder, M. R. (1970). Synthesis of low-peak factor signals and binary sequences with low autocorrelation. *IEEE Transactions on Information Theory*, pages 85–89.
- Skogestad, S. and Postlethwaite, I. (1997). *Multivariable Feedback Control - Analysis and Design*. John Wiley and Sons, Inc.
- Spencer, B. F. J., Dyke, S. J., and Deoskar, H. S. (1998a). Benchmark problems in structural control: Part ii - active tendon system. *Earthquake Engineering & Structural Dynamics*, 27(11):1141–1147.
- Spencer, B. F. J. and Sain, M. K. (1997). Controlling buildings: A new frontier in feedback. *IEEE Control Systems Magazine*, 17(3):19–35.
- Spencer, B. J., Dyke, S., and Deoskar, H. (1998b). Benchmark problems in structural control: Part i - active mass driver system. *Earthquake Engineering & Structural Dynamics*, 27(11):1127 – 1139.
- Whorton, M. S., Calise, A. J., and Hsu, C. C. (1998). Study of fixed-order mixed norm designs for a benchmark problem in structural control. *Earthquake Engineering & Structural Dynamics*, 27(11):1315–1330.
- Zhou, K. and Doyle, J. C. (1998). *Essentials of Robust Control*. Prentice Hall, Inc.

Apêndice A

Programa para teste do sistema embarcado

Todos os recursos do sistema embarcado envolvidos na implementação da bancada para controle ativo em tempo real proposta neste trabalho foram testados através de uma simulação utilizando a placa de interface com computador (MSC1210EVM). Essas ferramentas são: leitura de um sinal analógico (a temperatura captada de um sensor presente na placa) e conversão para formato digital, tratamento numérico (cálculo) com o valor desse sinal, transmissão do valor através da porta serial para ser monitorado possivelmente por um computador e acionamento do módulo regulador de largura de pulso (PWM) com ciclo de trabalho proporcional ao sinal obtido do processamento para fazer funcionar uma pequena buzina presente na placa de interface utilizada.

O programa resultante é mostrado a seguir:

```
/* Definições da interface do microcontrolador com a placa.*/  
#include <reg1210.h>  
/* Biblioteca de entrada e saída para usar a função printf. */  
#include <stdio.h>  
#include <math.h>  
  
/* Constantes para o tempo de interrupção.*/  
#define TEMPOH 0x9E;//ED;//DB;  
#define TEMPOL 0x57;//FF;  
  
/* Relação entre o maior valor de tensão lido na entrada DA (5 V)  
e o maior número representado pelos seus 24 bits.
```

```

5/2^24 = 298.0232e-9. */
#define VBIT 298.0232e-9

/* Definição de nomes para pinos da placa. */
sbit p33 = P3^3; // buzina
sbit p35 = P3^5; // LED amarelo
sbit led1 = P3^4; // LED vermelho

/* Variável para a transmissão serial. */
unsigned char mensagem;

/* Variáveis auxiliares para teste. */
bit alterna = 1;
int conta = 0;

/* Variáveis auxiliares para cálculos. */
float r,c;

/* Variáveis para o controlador. */
float x[4];
float xa[4];
float u;
float y;
int controle;

/* União para conter os 24 bits da conversão analógico-digital. */
static union {
    unsigned char c[4]; // bytes
    signed int i[2]; // signed words
    signed long l; // signed long
} res;

```

```

static union {
    unsigned char c[4]; // bytes
    signed int i[2]; // signed words
    signed long l; // signed long
} sinal;

/* Produz um tempo de atraso ajustado para a transmissão serial. */
void espera(void) {
    int i;
    for(i=0;i<240;i++);
}

/* Converte a variável 'mensagem' de binário para hexadecimal
codificado em ASCII. */
void converte_mensagem(void) {
    if((mensagem<=9)&&(mensagem>=0))
        mensagem = mensagem + '0';
    else if((mensagem>=10)&&(mensagem<=15))
        mensagem = mensagem + 'A' - 10;
}

/* Transmissão de um sinal pela porta serial. */
void transmite_sinal(void) {

    mensagem = sinal.c[0] >> 4;
    converte_mensagem();
    SBUF = mensagem;
    espera();
}

```

```
TI = 0;

mensagem = sinal.c[0] << 4;
mensagem = mensagem >> 4;
converte_mensagem();
SBUF = mensagem;
espera();
TI = 0;

mensagem = sinal.c[1] >> 4;
converte_mensagem();
SBUF = mensagem;
espera();
TI = 0;

mensagem = sinal.c[1] << 4;
mensagem = mensagem >> 4;
converte_mensagem();
SBUF = mensagem;
espera();
TI = 0;

mensagem = sinal.c[2] >> 4;
converte_mensagem();
SBUF = mensagem;
espera();
TI = 0;

mensagem = sinal.c[2] << 4;
mensagem = mensagem >> 4;
converte_mensagem();
SBUF = mensagem;
```

```

espera();
TI = 0;

mensagem = sinal.c[3] >> 4;
converte_mensagem();
SBUF = mensagem;
espera();
TI = 0;

mensagem = sinal.c[3] << 4;
mensagem = mensagem >> 4;
converte_mensagem();
SBUF = mensagem;
espera();
TI = 0;

SBUF = '\n'; // Fim de linha para separar os dados.
espera();
TI = 0;

SBUF = '\r'; // Fim de linha para separar os dados.
espera();
TI = 0;
}

```

```

/* Leitura AD, calculo do sinal de controle e impressao
de resultados na porta serial. */
void le_AD_calcula_controle(void) {
    int i;

    while(!(AIE&0x20)) {} // Espera conversao AD.
}

```

```

/* Leitura dos três registradores de 8 bits cada
preenchidos na conversão AD. */
res.i[0] = (unsigned char) ADRESH;
res.c[2] = ADRESM;
res.c[3] = ADRESL;

/* Ajuste do valor de tensão do sinal lido. */
r = res.l*VBIT;
if(r > 2.5)
    r = r-5;

u = 2e3*(r+0.07); //1e3*r-10; //2e3*(r-0.12); //0.1178);

/* sinal.l = (long int)(floor(4.5e6*u));

transmite_sinal();
*/

/* Cálculo do sinal de controle. */
for(i=0;i<4;i++)
    xa[i] = x[i];

x[0] = (0.898350059)*xa[0] + (0.397398131)*xa[1] + (0.0388638779)*u;
x[1] = (-0.397398131)*xa[0] + (0.898350059)*xa[1] + (-0.015177917)*u;

y = (-0.310283102)*xa[0] + (0.178700494)*xa[1];
y = 0.5*y;

/* Escalamento do sinal de controle para um valor inteiro. */
if(y>=0) {
    p35 = 0;

```

```

        led1 = 1;
        y = y;
    } else {
        p35 = 1;
        led1 = 0;
        y = -y;
    }
    controle = (int)(1e3*y);

```

```

/* Transmissão de resultados pela porta serial. */
/* Envia todo o resultado da conversão AD (24 bits) de quatro em
   quatro bits.*/

```

```

mensagem = ADRESH >> 4;
converte_mensagem();
SBUF = mensagem;
espera();
TI = 0;

```

```

mensagem = ADRESH << 4;
mensagem = mensagem >> 4;
converte_mensagem();
SBUF = mensagem;
espera();
TI = 0;

```

```

mensagem = ADRESM >> 4;
converte_mensagem();
SBUF = mensagem;
espera();
TI = 0;

```

```

mensagem = ADRESM << 4;
mensagem = mensagem >> 4;
converte_mensagem();
SBUF = mensagem;
espera();
TI = 0;

mensagem = ADRESL >> 4;
converte_mensagem();
SBUF = mensagem;
espera();
TI = 0;

mensagem = ADRESL << 4;
mensagem = mensagem >> 4;
converte_mensagem();
SBUF = mensagem;
espera();
TI = 0;

SBUF = '\n'; // Fim de linha para separar os dados.
espera();
TI = 0;
}

/* Atualização do PWM com o valor do sinal de controle
da interrupção atual. */
void ajusta_PWM(void) {
    PWMCON = 0; // selecionar período do PWM
    PWM = 1105; // valor do período (em tics do relógio)
}

```

```

PWMCON = 0x10; // selecionar duração do nível alto
PWM = controle; //552;//1000*c+100;
// valor tempo do nível alto (em tics do relógio)
PWMCON = 0x09; // ativar PWM usando SYSCLK/12 como relógio
}

```

```

/* Tratador da interrupção do TIMER0.*/

```

```

void TIMER0_int(void) interrupt 1 {
    /* Desabilita interrupção de TIMER0 e recarrega
    o valor de tempo.*/
    TRO = 0;
    TH0 = TEMPOH;
    TLO = TEMPOL;

    /* Processamento e comunicação. */
    le_AD_calcula_controle();

    /* Atualiza o valor do ciclo de trabalho do PWM
    com o valor do sinal recém-calculado. */
    ajusta_PWM();

    /* Habilita interrupção de TIMER0. */
    TRO = 1;
}

```

```

/* Configuração dos parâmetro do módulo de PWM.*/

```

```

void configura_PWM(void) {
    PDCON &= 0xED; // ativa SYSCLK e circuito de PWM.
    p33 = 1; // mantém o sinal da buzina em nível alto
    PWMCON = 0; // selecionar período do PWM
}

```

```

PWM = 1105; // valor do período (em tics do relógio)
PWMCON = 0x10; // selecionar duração do nível alto
PWM = 552; // valor tempo do nível alto (em tics do relógio)
PWMCON = 0x09; // ativar PWM usando SYSCLK/12 como relógio
}

```

/* Configuração dos parâmetros do conversor AD.

Parâmetros importantes:

A frequência de conversão dos bits do AD é $f_{con} = \text{SYSCLK}/(\text{ACLK}+1)$.

A frequência de modulação é $f_{mod} = f_{con}/64$.

A taxa de conversão de dados é $f_{dados} = f_{mod}/\text{DECIMATION}$.

Aqui são usados: $\text{ACLK} = 1$ e $\text{DECIMATION} = 864$,

resultando $f_{dados} = 100$ Hz.

A tensão de entrada do sinal analógico deve estar entre

0 e 2,5 V. */

```
void configura_AD(void) {
```

```
    PDCON &= 0x0f7; // aciona o conversor AD
```

```
    ADMUX = 0x16; // seleciona a origem dos dados
```

```
    // (FF: sensor de temperatura)
```

```
    ACLK = 0; // seleciona velocidade de conversão:
```

```
        // ACLK+1 divide o SYSCLK para funcionar como relógio
```

```
        // de conversão
```

```
    /* Parâmetros extras:*/
```

```
    ADCON0 = 0x30; // tensão interna de referência em 2,5 V, buffer
```

```
    // desligado,
```

```
        // sem detecção de burnout, ganho programável igual a 1
```

```
    ADCON1 = 0x11; // seleciona autocalibração de offset e ganho,
```

```
        // filtro de resposta rápida e registrador de soma
```

```
        // bipolar
```

```
    DECIMATION = 1440; // 864;
```

```

}

/* Configuração da transmissão serial de dados.
Importante: a taxa de transmissão (baud rate) é determinado por:
baud-rate = 2^(modo de transmissão)*SYSCLK/(32*12*(256-TH1))
Neste caso, TH1 = 253(FD), logo,
baud-rate = 11059200/(32*12*(256-253)) = 19200. */
void configura_serial(void) {
    SCON = 0x52; // seleciona modo 1 de transmissão (assíncrona de 10 bits),
                // habilita recepção pela porta serial, seleciona bit
                // de recepção igual a 1 e bit de transmissão igual a 0
    TMOD = 0x21; // seleciona TIMER0 operando em modo 1 (16 bits)
                // e TIMER1 operando em modo 2 (8 bits com auto-carga)
    TH1 = 0xFD; // seleciona taxa de transmissão, neste caso 19200 bps
    TR1 = 1;    // liga TIMER1, usado para transmissão/recepção serial
    RI = 0;    // Habilita início da transmissão.
}

/* Programa os parâmetros da interrupção principal do programa,
a de TIMER0.*/
void programa_temporizador(void) {
    TMOD = 0x21; // seleciona TIMER0 operando em modo 1 (16 bits)
                // e TIMER1 operando em modo 2 (8 bits com auto-carga)
                // o primeiro para interrupção principal; o segundo para
                // a transmissão serial
    TH0 = TEMPOH; // carrega o valor de tempo para interrupção
    TLO = TEMPOL; // de TIMER0
    ET0 = 1; // habilita interrupção de TIMER0
}

```

```
TR0 = 1; // liga TIMERO
EA = 1; // habilitador global de interrupções
}
```

```
/* Principal. */
void main(void)
{
    /* Colocar o PORT3 em nível alto.*/
    P3 = 0xFF;

    /* Invoca as funções de configuração. */
    programa_temporizador();
    configura_serial();
    configura_AD();
    configura_PWM();

    /* Faz nada esperando pelas interrupções. */
    while(1) {}
}
```

Apêndice B

Código para projetar os controladores usando Matlab

Conforme já dito, a ferramenta computacional utilizada para identificação e projeto dos controladores foi o Matlab. Esse programa possui pacotes de funções específicos para essas aplicações. A seguir, são mostrados arquivos de código na linguagem do Matlab indicando como utilizar suas funções no projeto dos controladores aqui projetados.

```
% modelo_cont é uma variável de sistema contínuo do matlab que contém o
% sistema completo com duas entradas da bancada de controle ativo de
% vibrações.
% Maurício J. Amorim
%
% O modelo do sistema é suposto na forma contínua e armazenado na variável
% 'modelo_cont'.
```

```
close all
```

```
% Extração das matrizes do sistema na representação no Matlab.
```

```
A = modelo_cont.A;
B1 = modelo_cont.B(:,1);
B2 = modelo_cont.B(:,2);
C1 = modelo_cont.C;
C2 = modelo_cont.C;
D11 = modelo_cont.D(:,1);
```

```

D12 = modelo_cont.D(:,2);
D21 = modelo_cont.D(:,1);
D22 = modelo_cont.D(:,1);

D12 = 1;
D21 = 1;

T = 25e-3; % Tempo de discretização dos controladores.

% Controlador H2
% Montagem de uma variavel de sistema do Matlab em notação compacta.
TSS_ = mksys(A,B1,B2,C1,C2,D11,D12,D21,D22,'tss');
% Projeto do controlador H2.
[SS_CP,SS_CL]=h2lqg(TSS_);
% Separação da notação compacta para as matrizes usuais do controlador.
[Ach2,Bch2,Cch2,Dch2] = branch(SS_CP);
% Separação da notação compacta para as matrizes usuais do sistema de
malha fechada.
[AclH2,BclH2,CclH2,DclH2] = branch(SS_CL);

sistema = ss(A,B1,C1,D11);
controladorH2 = ss(Ach2,Bch2,Cch2,Dch2);
malha_fechadaH2 = ss(AclH2,BclH2,CclH2,DclH2);

% Discretização do controlador e separação das matrizes.
controladorH2d=c2d(controladorH2,T);
AcdH2 = controladorH2d.A;
BcdH2 = controladorH2d.B;
CcdH2 = controladorH2d.C;
DcdH2 = controladorH2d.D;

% Grafico

```

```

figure(1)
sigma(sistema,malha_fechadaH2)
title('Controlador H_2 - Valor singular')
legend('Sem controle','Com controle')

h2sis = normh2(sistema);
h2mf = normh2(malha_fechadaH2);
disp('Norma H2 - planta')
h2sis
disp('Norma H2 - planta com controlador')
h2mf

% Controlador Hinf

% Projeto do controlador.
[SS_CP,SS_CL,HINFO,TSS_K]=hinf(TSS_);
% Separação da notação compacta para as matrizes usuais do controlador.
[AcHinf,BcHinf,CcHinf,DcHinf] = branch(SS_CP);
% Separação da notação compacta para as matrizes usuais do sistema de
malha fechada.
[AclHinf,BclHinf,CclHinf,DclHinf] = branch(SS_CL);

controladorHinf = ss(AcHinf,BcHinf,CcHinf,DcHinf);
malha_fechadaHinf = ss(AclHinf,BclHinf,CclHinf,DclHinf);

% Discretização do controlador e separação das matrizes.
controladorHinfd=c2d(controladorHinf,T);
AcdHinf = controladorHinfd.A;
BcdHinf = controladorHinfd.B;
CcdHinf = controladorHinfd.C;
DcdHinf = controladorHinfd.D;

```

```
% Grafico
figure(2)
sigma(sistema,malha_fechadaHinf)
title('Controlador  $H_{\infty}$  - Valor singular')
legend('Sem controle','Com controle')

hinfsis = normhinf(sistema);
hinfmf = normhinf(malha_fechadaHinf);
disp('Norma Hinf - planta')
hinfsis
disp('Norma Hinf - planta com controlador')
hinfmf
```

Apêndice C

Comunicação serial do sistema embarcado

Para facilitar o trabalho de comunicação serial e utilizar o mínimo de tempo e memória do microcontrolador foi adotado um padrão particular de comunicação serial. Na verdade, a principal limitação no projeto não era a memória do sistema embarcado, mas a quantidade dela que o compilador usado (em sua versão gratuita) permitia utilizar.

Foi enviado para o computador, pela transmissão serial, o valor lido na ponte de extensômetros, de 24 bits de precisão. Esses 24 bits eram interpretados, de quatro em quatro (e portanto, em seis grupos de quatro bits) como algarismos hexadecimais. Cada algarismo hexadecimal foi convertido para o código ASCII e enviado pela porta serial da placa MSC1210. Dessa forma, quando lido pelo Matlab, cada algarismo pôde ser interpretado utilizando uma função padrão de leitura de caracteres ASCII. Cada caracter é convertido para número do Matlab e ponderado de acordo com a seqüência lida (a posição na seqüência de bits) na base hexadecimal (base 16).

O código em C do programa final para controle é mostrado a seguir:

```
    /* Transmissão de resultados pela porta serial. */  
/* Envia todo o resultado da conversão AD (24 bits) de quatro em quatro  
bits.*/  
  
    mensagem = ADRESH >> 4;  
converte_mensagem();  
SBUF = mensagem;  
espera();  
TI = 0;  
  
mensagem = ADRESH << 4;
```

```
mensagem = mensagem >> 4;
converte_mensagem();
SBUF = mensagem;
espera();
TI = 0;

    mensagem = ADRESM >> 4;
converte_mensagem();
SBUF = mensagem;
espera();
TI = 0;

mensagem = ADRESM << 4;
mensagem = mensagem >> 4;
converte_mensagem();
SBUF = mensagem;
espera();
TI = 0;

    mensagem = ADRESL >> 4;
converte_mensagem();
    SBUF = mensagem;
espera();
TI = 0;

mensagem = ADRESL << 4;
mensagem = mensagem >> 4;
converte_mensagem();
SBUF = mensagem;
espera();
TI = 0;
```

```
SBUF = '\n'; // Fim de linha para separar os dados.
espera();
TI = 0;
```

Nota-se que os bits da resposta da conversão AD são separados de quatro em quatro dos três registradores de leitura de oito bits cada. O número lido é somado ao código ASCII dos caracteres 0 e A, conforme seja ou não menor do que 10.

No Matlab, por sua vez, uma pequena função foi elaborada para recepção e conversão dos dados da porta serial.

```
clear all
close all

% objeto de leitura da porta serial
s1 = serial('COM1', 'BAUDRATE', 9600);
fopen(s1);

n = 6000; % numero de pontos a ser lidos da porta serial
z = [];
M = [];
i = 1;

% tabela para interpretar os caracteres em números
tabela = ['0' '1' '2' '3' '4' '5' '6' '7' '8' '9' 'A' 'B' 'C' 'D' 'E' 'F'];
m = [];

while(i<n)
    numero = 0;
    m = fgets(s1);

    k = 1;
    % conversão de cada caracter
    for c=1:length(m)
```

```

% testa se é um caracter válido
    if( ((m(c)>='0') & (m(c)<='9')) | ((m(c)>='A') & (m(c)<='F')) )
        % busca na tabela
        indice = find(tabela==m(c));
% podera o caracter na base hexadecimal
        numero = numero + (indice-1)*16^(6-k);
        k = k + 1;
    end
end
i = i + 1;
% armazena o número
z = [z; numero];

% armazena a leitura na forma de string
M = [M m];
i=i+1;
end

% multiplcação pela constante de solução de cada bit
z = z*(5/(2^24-1));

plot(z(5:end))

% pára de usar a porta serial
fclose(s1);

```

Apêndice D

Linguagem Gráfica SDL

Conforme (Macário, 2003), a Linguagem de Especificação SDL (do inglês *Specification and Description Language*) é uma linguagem formal e orientada a objetos definida pela ITU-T (1999), antiga CCITT, cujo desenvolvimento iniciou-se em 1972 para a especificação e a descrição de sistemas de telecomunicações. Atualmente é usada em sistemas complexos dirigidos por eventos de tempo real e interativos, e que envolvam atividades concorrentes. Sua primeira versão foi lançada em 1976, tendo sido seguida de novas versões lançadas a cada quatro anos. A versão lançada em 1992, denominada SDL-92, representou um marco na sua história por incorporar conceitos de objetos, como tipos, reuso e herança, ampliando a sua abrangência, com a tendência de ser utilizada na especificação dos mais diversos produtos de diferentes áreas, principalmente em sistemas de tempo real, interativos ou distribuídos. Atualmente é uma linguagem de padrão internacional, sendo utilizada em empresas de diversos países e tendo com principais áreas de atuação: comunicação de satélites; aeronáutica; equipamentos médicos; sistemas de controle de estradas de ferro; e, protocolos de comunicação usados nos carros. A linguagem SDL serve tanto para representar o comportamento do sistema, como sua estrutura, usando diferentes níveis de abstração, do mais alto ao mais detalhado. Uma de suas características fortes é que, diferente da maioria das linguagens de especificação formal que apresentam apenas uma representação textual, a SDL possui também uma representação gráfica, denominada SDL-GR, baseada em símbolos, o que provê clareza e facilidade de uso, facilitando bastante o seu entendimento. Além disso, por ser formal, possui sintaxe e semântica bem definidas, possibilitando a simulação e a validação automática do sistema sendo especificado, mesmo nas fases iniciais do desenvolvimento, através de uma ferramenta CASE. Uma especificação SDL provê diversas visões de um sistema, sendo elas:

Visão estrutural:

- decomposição hierárquica do *software* em sistema, blocos e processos;
- hierarquia de tipos, através de herança e especialização (apenas da versão SDL-92 em diante).

A visão estrutural permite o particionamento do sistema em diferentes níveis hierárquicos e tem como principais objetivos: ocultar informação, movendo detalhes para níveis mais baixos; seguir as subdivisões naturais; criar módulos de tamanhos gerenciáveis; criar correspondência com o *hardware* e o *software* do mundo real; e, reusar especificações prontas. Nesta decomposição, a SDL provê interfaces bem definidas entre as diversas entidades do sistema. A Figura D.1 ilustra a decomposição de um sistema na linguagem SDL.

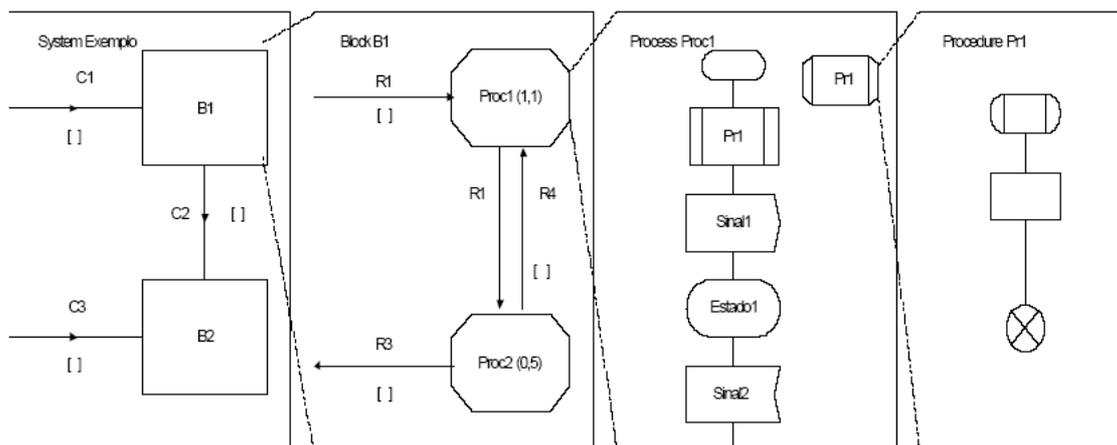


Figura D.1: Decomposição hierárquica de um sistema na linguagem SDL

Visão de comunicação:

- sinais assíncronos e seus parâmetros opcionais;
- chamadas remotas a procedimentos (apenas da versão SDL-92 em diante).

Visão comportamental:

- é descrita através dos processos.

Visão de dados:

- tipos abstratos de dados (sorts) definidos;
- tipos de dados ASN.¹².

Dentre estas visões, a comportamental é a que apresenta-se como alternativa ao diagrama de estados proposto pela UML.

A visão comportamental do sistema descrito em SDL é dada através dos processos. Um processo em SDL é uma máquina de estados finita e pode ser criado no início ou durante a execução do sistema

e podem existir ao mesmo tempo uma ou mais instâncias do processo, tendo cada uma delas um identificador único. A comunicação entre os processos se dá através de sinais, que podem ou não utilizar parâmetros. A troca de sinais é sempre entre dois processos e nunca entre vários deles. A Figura D.2 ilustra a representação de um processo numa especificação SDL.

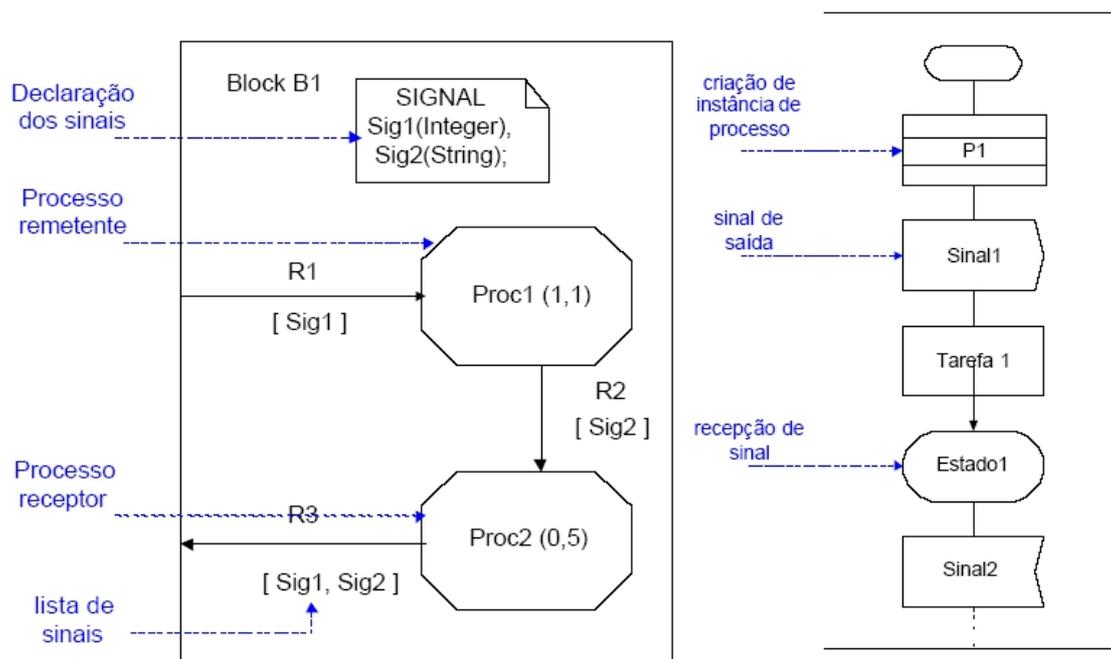


Figura D.2: Representação de um processo numa especificação SDL

Para a descrição do comportamento de um processo é utilizado um conjunto de símbolos que compõem a linguagem gráfica SDL. Os principais deles são apresentados na tabela da Figura D.3.

	Início da execução do processo		término da execução do processo
	representa o estado, cujo nome está escrito no símbolo; em especial o uso de - no seu nome indica o estado não se altera após a execução de alguma ação e o uso de * refere-se a todos os estados, estabelecendo ações comuns a todos eles; a mudança de estado se dá através da recepção de um sinal.		criação de instância de um processo do mesmo bloco
	tarefa, usada para definir uma ação; pode ser um 'texto' ou um comando		decisão; contém uma questão e um determinado número de respostas
	sinal de entrada; vem sempre depois de um estado		sinal de saída, usado para enviar sinais e dados para outros processos
	chamada a procedimento		início da execução do procedimento
	término da execução do procedimento		comentário
	conector, usado para conectar partes do processo		

Figura D.3: Definição dos principais símbolos da linguagem SDL