

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CURSO DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

**DESENVOLVIMENTO DE UM AMBIENTE DE SIMULAÇÃO PARA
ANÁLISE DE DESEMPENHO DE CONTROLADORES ADAPTATIVOS**

Dissertação submetida à Universidade Federal de Santa Catarina para obtenção do
grau de Mestre em Engenharia Elétrica

Leonardo César Kammer

Florianópolis, 22 de Dezembro de 1992.

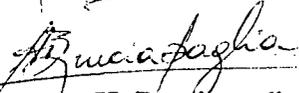
DESENVOLVIMENTO DE UM AMBIENTE DE SIMULAÇÃO PARA ANÁLISE DE DESEMPENHO DE CONTROLADORES ADAPTATIVOS

LEONARDO CÉSAR KAMMER

ESTA DISSERTAÇÃO FOI JULGADA PARA A OBTENÇÃO DO TÍTULO DE

MESTRE EM ENGENHARIA

ESPECIALIDADE ENGENHARIA ELÉTRICA, ÁREA DE CONCENTRAÇÃO SISTEMAS DE CONTROLE E AUTOMAÇÃO INDUSTRIAL, E APROVADA EM SUA FORMA FINAL PELO CURSO DE PÓS-GRADUAÇÃO

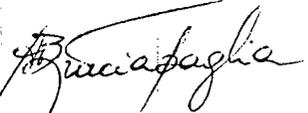


Prof. Augusto H. Bruciapaglia, Dr. Ing.
Orientador

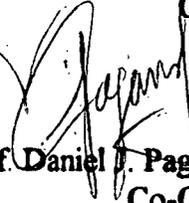


Prof. Roberto de Souza Salgado, Ph. D.
Coordenador do Curso de Pós-Graduação em Engenharia Elétrica

BANCA EXAMINADORA:



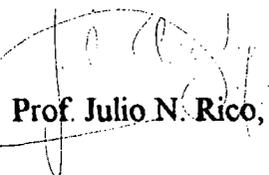
Prof. Augusto H. Bruciapaglia, Dr. Ing.
Orientador



Prof. Daniel J. Pagano, MSc
Co-Orientador



Prof. Liu Hsu, Dr. d'Etat



Prof. Julio N. Rico, MSc

À Claudia Aurich, pelo amor compartilhado.

Ao amigo Saul Caetano, pelos exemplos de honestidade e coerência.

AGRADECIMENTOS

Agradeço a todas as pessoas com as quais dialoguei durante o período de realização do mestrado, pois o viver se estendeu além do trabalhar. Em especial agradeço a meus pais por todo apoio e incentivo. Às pessoas que comigo conviveram neste período, principalmente a família Aurich e os amigos do LCMI. Aos orientadores Augusto Humberto Bruciapaglia e Daniel Juan Pagano, tanto pela ótima orientação quanto pela amizade e liberdade. Ao amigo, e analista do LCMI, Jorge Hermógenes por manter o bom funcionamento da estrutura computacional necessária à realização deste trabalho. Aos brasileiros que, mesmo de uma forma inconsciente, financiaram este trabalho provendo recursos ao CNPq.

RESUMO

A utilização das técnicas de controle adaptativo, no meio industrial, vem ocorrendo lentamente, em parte devido à falta de confiança nos estudos de avaliação de desempenho realizados até o presente momento. O tema "avaliação de desempenho", no contexto do controle adaptativo, não foi ainda bastante explorado, consistindo em uma lacuna aberta a investigações.

O presente trabalho busca oferecer uma contribuição no estudo deste tema, através da implementação de um ambiente de simulação, denominado SADECA, especialmente voltado à avaliação de desempenho de controladores adaptativos monovariáveis. Pretende-se realizar avaliações voltadas à realidade prática, de uma forma sistemática e quantitativa.

O ambiente implementado permite reproduzir com realismo o comportamento de sistemas práticos, podendo ser incluídos os efeitos do ruído, das perturbações, da atuação de conversores A/D e D/A, da variação em parâmetros da planta, etc. As simulações podem ser realizadas, no ambiente SADECA, através de seqüências pré-programadas de eventos, ao final das quais são gerados relatórios contendo índices de desempenho. O ambiente implementado constitui também uma ferramenta útil no desenvolvimento de estratégias de controle adaptativo, permitindo a programação de novos controladores. A forma como foi desenvolvida a interface gráfica torna a interação com o usuário bastante agradável e eficiente.

Os resultados obtidos no aperfeiçoamento e avaliação de desempenho de um controlador adaptativo baseado em reconhecimento de formas, comprovam a importância deste ambiente de simulação.

ABSTRACT

The utilization of adaptive control techniques in industrial systems has grown slowly, partly due to the lack of reliability on performance evaluation studies. The theme "performance evaluation" in adaptive control context is not yet well explored, which represents a gap opened to investigations.

This work attempts to contribute to studying this topic. The implementation of a simulation environment for performance evaluation of SISO adaptive controllers is presented. These evaluations are intended to be accomplished on practical aspects, while in a systematic and quantitative way.

The environment implemented allows the reproduction of real systems behavior. The user can introduce the effects of noise, load disturbances, A/D and D/A converters, changes in plant parameters, etc. The simulations may be done by means of pre-programmed event sequences, with generation of reports containing performance indices. This environment is a useful tool in the development of adaptive control strategies, allowing the programming of new controllers. The way the graphical interface was developed makes man-machine interaction very pleasant and efficient.

Results obtained in the improvement and performance evaluation of an adaptive controller based on pattern recognition show the importance of this simulation environment.

SUMÁRIO

AGRADECIMENTOS.....	iv
RESUMO.....	v
ABSTRACT.....	vi
CAPÍTULO 1 : INTRODUÇÃO.....	1
CAPÍTULO 2 : ASPECTOS DE AVALIAÇÃO DE CONTROLADORES ADAPTATIVOS.....	3
2.1. INTRODUÇÃO.....	3
2.2. ABORDAGENS DE CONTROLE ADAPTATIVO.....	4
2.2.1. SISTEMA ADAPTATIVO POR MODELO DE REFERÊNCIA (MRAS).....	4
2.2.2. REGULADOR AUTO SINTONIZÁVEL (STR).....	5
2.2.3. SISTEMAS BASEADOS EM RECONHECIMENTO DE FORMAS.....	6
2.3. CONTROLADORES ADAPTATIVOS DE PROPÓSITO GERAL.....	8
2.4. AVALIAÇÃO DE DESEMPENHO.....	9
2.4.1. OBJETIVOS DA AVALIAÇÃO.....	10
2.4.2. PARÂMETROS DE DESEMPENHO.....	12
2.4.3. AMBIENTES DE AVALIAÇÃO.....	13
2.5. CONCLUSÃO.....	14
CAPÍTULO 3 : ESPECIFICAÇÃO FUNCIONAL.....	15
3.1. INTRODUÇÃO.....	15
3.2. ESPECIFICAÇÃO INFORMAL.....	15
3.3. ESTRUTURA BÁSICA.....	17
3.3.1. MÓDULO CONTROLADOR.....	17
3.3.2. MÓDULO PROCESSO.....	18
3.3.3. MÓDULO SUPERVISOR.....	19
3.4. ESTRUTURA DE AVALIAÇÃO.....	19

3.4.1. MÓDULO AVALIADOR	20
3.4.2. MÓDULO EXECUTOR	22
3.5. INTERFACE HOMEM-MÁQUINA	22
3.6. MODOS DE OPERAÇÃO	22
3.6.1. DECLARAÇÃO	23
3.6.2. SIMULAÇÃO MANUAL	24
3.6.3. SIMULAÇÃO AUTOMÁTICA	25
3.7. CONCLUSÃO	26
CAPÍTULO 4 : IMPLEMENTAÇÃO	27
4.1. INTRODUÇÃO	27
4.2. MECANISMOS BÁSICOS DE UMA LINGUAGEM ORIENTADA A OBJETOS	28
4.3. MÓDULO CONTROLADOR	29
4.4. MÓDULO PROCESSO	30
4.4.1. BLOCO DLC_FT	32
4.4.2. BLOCO DLC_EE	34
4.4.3. BLOCO DLD	34
4.4.4. BLOCO ATR	34
4.4.5. BLOCO GNL	35
4.4.6. BLOCO PRT	35
4.5. MÓDULO SUPERVISOR	35
4.6. MÓDULO EXECUTOR	36
4.7. MÓDULO AVALIADOR	37
4.8. INTERFACE HOMEM-MÁQUINA	40
4.8.1. DECLARAÇÃO	41
4.8.2. SIMULAÇÃO MANUAL	42
4.8.3. SIMULAÇÃO AUTOMÁTICA	44
4.9. CONCLUSÃO	45

CAPÍTULO 5 : UM EXEMPLO DE UTILIZAÇÃO DO AMBIENTE DE SIMULAÇÃO	46
5.1. INTRODUÇÃO	46
5.2. O CONTROLADOR PID-AA	46
5.3. IMPLEMENTAÇÃO DO PID-AA NO AMBIENTE SADECA.....	48
5.4. ANÁLISE DO MECANISMO DE INICIALIZAÇÃO	50
5.5. ANÁLISE DAS REGRAS DE AJUSTE	53
5.6. UMA AVALIAÇÃO DE DESEMPENHO	57
5.6.1. APRESENTAÇÃO DO "BENCHMARK"	57
5.6.2 AVALIAÇÃO DE DESEMPENHO DA VERSÃO V1 DO PID-AA	61
5.6.3 AVALIAÇÃO DE DESEMPENHO DA VERSÃO V3 DO PID-AA	63
5.7. CONCLUSÃO.....	65
CAPÍTULO 6 : CONCLUSÕES E PERSPECTIVAS	66
ANEXO I : MANUAL DO USUÁRIO DO AMBIENTE SADECA.....	68
ANEXO II : ALTERAÇÃO NA ORDEM DO MODELO DOS BLOCOS DLC_FT	103
ANEXO III : CONJUNTO DE PROCESSOS UTILIZADOS NAS ANÁLISES DO CONTROLADOR PID-AA.....	106
BIBLIOGRAFIA	109

CAPÍTULO 1 : INTRODUÇÃO

É bastante comum serem encontradas plantas industriais operando com baixa eficiência em função da atuação de controladores mal ajustados, aumentando, desta forma, os custos de produção. Considerando-se que a quase totalidade dos controladores utilizados no meio industrial são monovariáveis e a parâmetros fixos, como por exemplo os clássicos controladores PID (proporcional, integral, derivativo), algumas razões podem ser apontadas como as principais motivadoras deste ajuste incorreto:

- inabilidade da pessoa encarregada do ajuste do controlador;
- não linearidades do conjunto atuador-processo-medidor;
- variações na dinâmica do processo.

Excluindo-se os casos em que o próprio esquema de controle adotado não se apresenta adequado à situação, como o caso de processos multivariáveis, com forte acoplamento entre suas variáveis, sendo controlado por estruturas monovariáveis, alguns mecanismos podem ser sugeridos para solucionar o problema de ajustes ineficientes dos controladores.

Para processos lineares invariantes no tempo é possível realizar um bom ajuste através de ensaios de identificação de certas características do processo, com as quais calculam-se os parâmetros do controlador. Estes ensaios têm sido refinados, visando a obtenção de melhores parâmetros, e automatizados, tornando-se um mecanismo extremamente útil e fácil de operar.

As características não lineares do conjunto atuador-processo-medidor podem ser compensadas através de funções inversas, linearizando o produto: ganho do processo \times funções, ou planejamento de ganho ("gain scheduling"), alterando os parâmetros do controlador em função do ponto de operação do sistema.

O problema de controle de processos variantes no tempo motivou o desenvolvimento de técnicas de controle adaptativo, com as quais torna-se possível detectar variações na dinâmica do processo e compensá-las, alterando os parâmetros do regulador. Estas técnicas podem ser utilizadas em conjunto com os mecanismos apresentados acima, melhorando ainda mais o desempenho do sistema.

As técnicas de controle adaptativo vêm sendo desenvolvidas há mais de três décadas e sua implantação no meio industrial começa a ocorrer gradativamente [Hägglund 91]. A lenta implantação destas técnicas em ambientes industriais é devida principalmente a dois fatores:

- falta de confiabilidade nos estudos de avaliação de desempenho e de testes comparativos, realizados via simulação ou sobre plantas piloto de laboratório;
- falta de conhecimento e treinamento no uso destas técnicas por parte de engenheiros e operadores de planta.

A utilização de ambientes de simulação, implementados em computadores, vem contribuindo para contornar estes obstáculos. Estes ambientes podem fornecer potentes recursos aos usuários, a um custo bastante inferior ao de plantas piloto, além de garantir absoluta segurança durante o treinamento de operadores de planta.

Face ao exposto acima, constatou-se que a linha de pesquisa de controle do Laboratório de Controle e Microinformática (LCMI) deparava-se com a necessidade de um ambiente de software voltado à avaliação e desenvolvimento de controladores adaptativos. O principal objetivo da implementação deste ambiente seria permitir uma integração dos diversos mecanismos de análise utilizados até então, mas que, em função de seu caráter desconexo, demandavam quase a totalidade do tempo de avaliação em interações com o usuário.

De forma a tornar este ambiente uma ferramenta de ampla utilização, sua especificação prioriza os aspectos práticos, tanto de simulação do processo e do controlador, quanto da avaliação. Desta maneira é possível observar, e analisar, o comportamento dos controladores adaptativos frente a situações quase idênticas às encontradas em ambientes industriais reais.

Esta dissertação de mestrado compreende o desenvolvimento deste ambiente de simulação de sistemas de controle, voltado à avaliação de desempenho de controladores adaptativos monovariáveis, com o objetivo de contribuir para a solução dos problemas que vêm dificultando a implantação destes controladores no meio industrial.

No capítulo 2 são apresentados alguns aspectos básicos relativos ao tema de controle adaptativo, tais como as possíveis abordagens e os mecanismos auxiliares, além de uma incursão na complexa questão da avaliação de desempenho de controladores adaptativos.

Os capítulos 3 e 4 descrevem, respectivamente, a especificação funcional e detalhes de implementação deste ambiente de simulação, permitindo um conhecimento de sua estrutura interna.

Um exemplo de utilização desta ferramenta computacional é apresentado no capítulo 5, onde é efetuada a avaliação e o aperfeiçoamento de um controlador PID adaptativo baseado em reconhecimento de formas.

CAPÍTULO 2 : ASPECTOS DE AVALIAÇÃO DE CONTROLADORES ADAPTATIVOS

2.1. INTRODUÇÃO

Surgido durante a década de 50, o controle adaptativo ainda não possui uma definição que seja consensual entre os autores, apesar de, em pelo menos duas ocasiões, haver sido realizadas tentativas neste sentido (um simpósio em 1961 e um comitê em 1973). Aparentemente há um consenso de que o objetivo básico do controle adaptativo é prover mecanismos de alteração nos parâmetros do controlador, em função de variações na dinâmica do processo controlado, de forma a manter o desempenho do sistema.

Uma característica comumente aceita pelos principais autores ([Åström 89], [Landau 79], [Ogata 70], etc.) é que esta adaptação dos parâmetros do controlador norteia-se por um conjunto de índices de desempenho pré-definidos, os quais são comparados com índices obtidos através de medição. Isto conduz a uma segunda malha de realimentação, onde a variável controlada é o desempenho do sistema. A figura 2.1 apresenta a estrutura genérica do controle adaptativo, evidenciando a existência de duas malhas de controle.

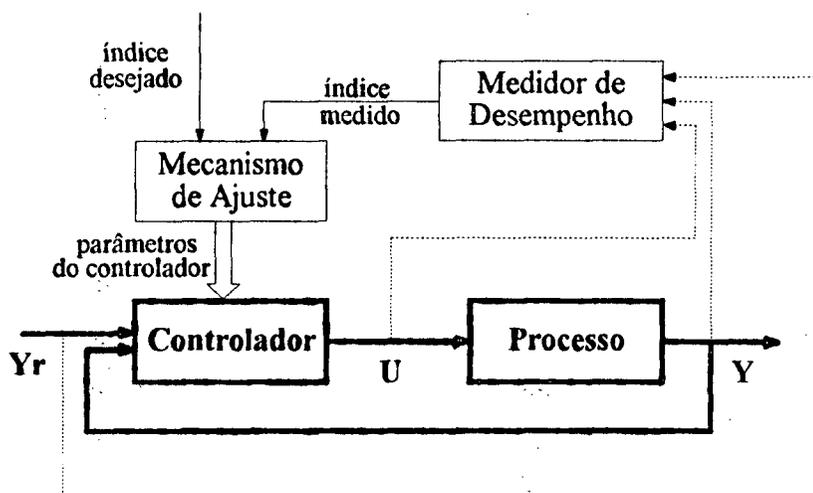


Fig. 2.1 - Estrutura básica de controle adaptativo

Com base nas características apresentadas, podemos encontrar diferentes abordagens para a estrutura dos controladores adaptativos. Na seção 2.2 são apresentadas, de forma sucinta, as abordagens mais importantes.

Inicialmente os controladores adaptativos eram projetados para atender a casos específicos. Com seu aperfeiçoamento e a inclusão de mecanismos apropriados, tais como o pré-

ajuste, estes controladores começam a ser produzidos em larga escala e atender a uma ampla gama de processos. O estado atual dos controladores adaptativos é discutido na seção 2.3 .

Em decorrência do surgimento de diversos controladores adaptativos de propósito geral, abre-se ao usuário um leque de opções à solução de seus problemas de controle. Entretanto a existência de opções suscita a dúvida quanto à escolha mais apropriada para o(s) caso(s) que se pretende resolver. Considerações a respeito deste problema de avaliação de controladores adaptativos são realizadas na seção 2.4 .

Por uma questão de simplicidade, e com vistas a manter uma uniformidade na apresentação desta dissertação, algumas considerações são feitas:

- os controladores adaptativos estarão limitados ao caso monovariável;
- o termo "controle adaptativo" refere-se à sub-área de conhecimento do controle, enquanto que "controlador adaptativo" refere-se a uma implementação de controlador que utilize técnicas de controle adaptativo;

2.2. ABORDAGENS DE CONTROLE ADAPTATIVO

O desenvolvimento dos controladores adaptativos teve seu início através de duas abordagens diferentes: o MRAS ("Model-Reference Adaptive System") e o STR ("Self-Tuning Regulator"). Estas siglas englobam uma grande quantidade de variações quanto aos mecanismos internos, mas mantém sempre seu caráter matemático formal.

A partir do desenvolvimento de um controlador adaptativo de propósito geral, baseado no reconhecimento da forma de onda do sinal controlado, realizado pela companhia Foxboro [Bristol 83], esta nova abordagem toma impulso e assume um papel de importância no contexto do controle adaptativo. Quase todos controladores adaptativos seguem uma destas três principais abordagens, as quais são apresentadas sucintamente nas seções seguintes.

2.2.1. SISTEMA ADAPTATIVO POR MODELO DE REFERÊNCIA (MRAS)

A idéia básica de um MRAS é fazer com que o sinal de saída do sistema tenha um comportamento idêntico ao de um modelo pré-determinado (modelo de referência), realizando alterações em parâmetros do controlador de forma a compensar variações no comportamento do processo. Com isto define-se, a priori, os índices de desempenho do sistema, pois o comportamento deste, em malha fechada, será conhecido para todos os instantes e frente a quaisquer sinais de referência (Y_r).

A figura 2.2 apresenta uma das configurações possíveis à abordagem MRAS. A todo momento a variável controlada (Y) é comparada com a saída especificada pelo modelo de referência (Y_m), sendo a diferença entre estas utilizada no ajuste dos parâmetros do controlador. Uma descrição detalhada do funcionamento deste sistema adaptativo pode ser encontrada em [Åström 89] e demais literaturas referenciadas por esta.

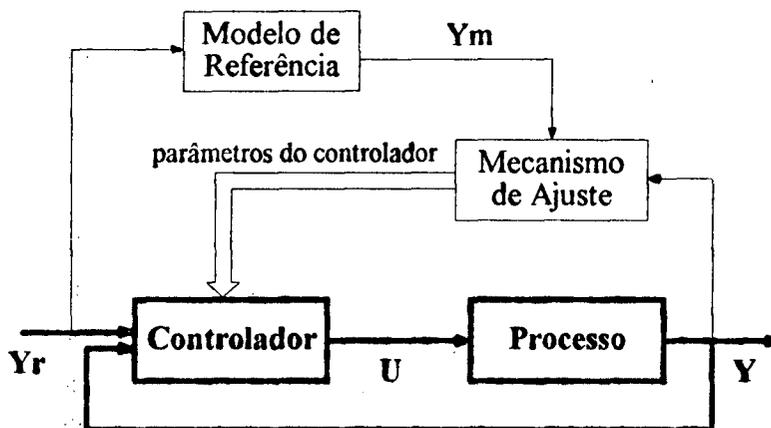


Fig. 2.2 - Diagrama de um MRAS

Uma das principais vantagens desta abordagem é sua alta velocidade de adaptação. Em contrapartida, um certo conhecimento a priori do processo é necessário para a implementação [Landau 79]. Além disto o MRAS analisa o problema de controle adaptativo somente do ponto de vista de mudanças de referência, devendo o caso de perturbações de carga ser tratado de forma especial, como por exemplo na solução apresentada em [Ioannou 86].

2.2.2. REGULADOR AUTO SINTONIZÁVEL (STR)

O STR aborda o problema de controle adaptativo sob a perspectiva de um engenheiro de controle, que identifica os parâmetros de um modelo do processo e, em função destes, projeta o controlador que satisfaça as especificações do sistema em malha fechada. Estas etapas, bastante distintas no funcionamento do STR, são apresentadas na figura 2.3.

A estimação dos parâmetros do processo é realizada a todo instante, fornecendo continuamente os valores relativos a um modelo matemático aproximado, com o qual se deseja caracterizar este processo. A etapa de projeto do controlador é responsável pela atualização dos parâmetros do controlador, utilizando para isto as informações à respeito do processo, obtidas na etapa de estimação. A questão do conjunto de índices de desempenho do sistema é considerada, implicitamente, na fase de projeto do controlador.

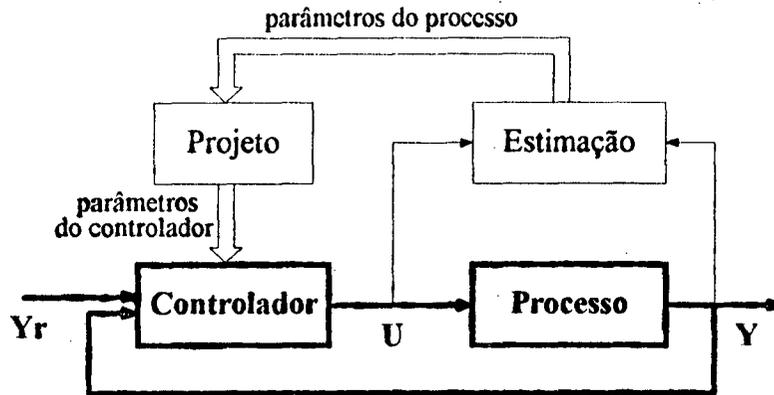


Fig. 2.3 - Diagrama de um STR

Existe uma variedade de métodos possíveis de serem utilizados em cada etapa desta abordagem de controle adaptativo. Para a etapa de estimação podem ser citados os métodos de aproximação estocástica, mínimos quadrados e variáveis instrumentais, entre outros. Alguns exemplos de métodos a serem utilizados na etapa de projeto são: mínima variância, alocação de pólos, predição, etc [Wellstead 86].

Algumas implementações de STR unem as etapas de estimação e projeto, fazendo com que os valores obtidos pela identificação do processo sejam os próprios parâmetros do controlador. Isto resulta em uma significativa simplificação do algoritmo devido à eliminação dos cálculos relativos ao projeto [Åström 89].

Dentre os STR podemos destacar, devido a seu bom desempenho, os controladores preditivos, e em especial o GPC ("Generalized Predictive Control") [Clarke 85]. Em [De Keyser 88] podemos encontrar um estudo a respeito de diversos métodos de controle preditivo, incluindo uma análise de comportamento frente a perturbações de carga e outras situações.

De forma semelhante ao MRAS, esta abordagem apresenta uma alta velocidade de adaptação, mas necessita de um certo conjunto de informações a priori a respeito do processo.

2.2.3. SISTEMAS BASEADOS EM RECONHECIMENTO DE FORMAS

A abordagem de controle adaptativo baseada em reconhecimento de formas realiza ajustes no controlador de maneira semelhante a um operador de planta, o qual identifica o comportamento do sinal controlado, frente a excitações conhecidas, e altera os parâmetros do controlador de acordo com sua experiência na operação desta planta.

A figura 2.4 apresenta o diagrama em blocos desta abordagem. O mecanismo de reconhecimento da forma de onda colhe informações da curva de resposta do sinal de saída (Y),

frente a mudanças de referência (Y_r) ou perturbações de carga (W) de tipos conhecidos, geralmente de graus. Estas informações são analisadas pelo mecanismo de ajuste que as compara com um padrão pré-estabelecido, alterando, se necessário, os parâmetros do controlador [Bristol 83].

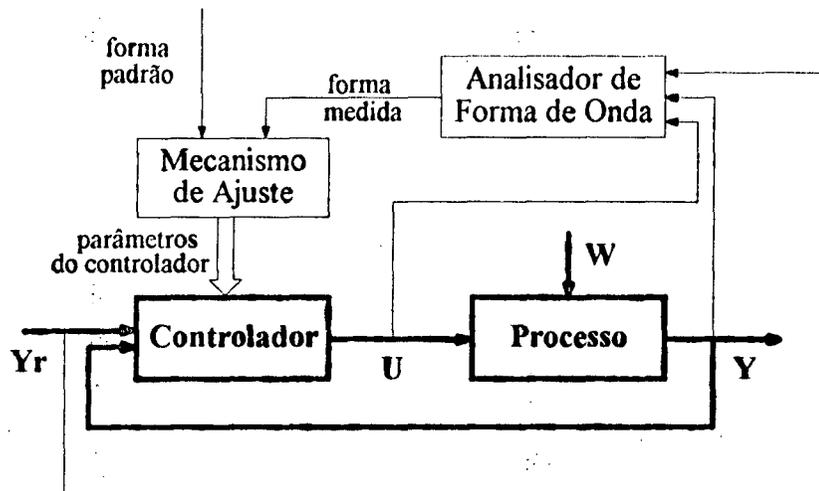


Fig. 2.4 - Diagrama de um Sistema Adaptativo Baseado em Reconhecimento de Formas

O conjunto de índices de desempenho pretendido para o sistema está inserido no padrão de forma de onda especificado. As informações que compõem este padrão dependem da implementação realizada, mas o sobressaio e a razão de amortecimento são exemplos bastante típicos de dados utilizados nesta abordagem [Gomide 91], [Pagano 89], [Bristol 83].

Geralmente o mecanismo de ajuste apresenta-se na forma de um sistema especialista, onde o conhecimento é descrito por regras que comandam o ajuste do controlador. Estas regras podem ser desenvolvidas a partir de estudos teóricos e/ou obtidas através de entrevistas com especialistas nesta área de conhecimento [Porter 87].

Quase a totalidade dos controladores adaptativos implementados sob esta abordagem utilizam como controlador uma estrutura do tipo PID, amplamente utilizada no meio industrial. Esta estrutura apresenta facilidade de operação, requerendo o ajuste de apenas três parâmetros, aliada a um bom desempenho apresentado no controle de inúmeras plantas industriais. A confiança existente no desempenho dos controladores PID auxilia à aceitação no meio industrial dos novos controladores adaptativos que se propõem a ser um aprimoramento desta estrutura, apesar de inibir a utilização de estruturas mais apropriadas.

A velocidade de adaptação desta abordagem é, teoricamente, mais lenta que a das abordagens MRAS e STR, em função destas ajustarem o controlador em todos os instantes, ao contrário da primeira, que avalia toda a forma de onda da saída para, só então, alterar os

parâmetros do controlador. Com relação às informações a priori do processo, a abordagem baseada em reconhecimento de formas apresenta vantagens consideráveis em relação às demais, em função da pequena quantidade e facilidade de obtenção destas informações [Versteeg 86], [Gawthrop 90], [Gomide 91].

2.3. CONTROLADORES ADAPTATIVOS DE PROPÓSITO GERAL

Em uma primeira etapa na história do desenvolvimento de controladores adaptativos (principalmente MRAS e STR), os pesquisadores eram chamados a resolver determinados problemas encontrados em sistemas de controle, apresentando soluções específicas para estes casos. As implementações eram realizadas com dispositivos analógicos, e os resultados nem sempre conseguiam ser satisfatórios [Åström 89].

Com o surgimento dos microprocessadores, a área de controle adaptativo recebeu um impulso bastante grande, possibilitando a implementação de algoritmos relativamente complexos, e a comprovação prática de seu funcionamento. A crescente confiabilidade nestas técnicas de controle incentivaram sua utilização em uma ampla quantidade de processos, o que proporcionou o desenvolvimento de controladores adaptativos de propósito geral.

A partir do desenvolvimento dos primeiros controladores adaptativos de propósito geral, surge a abordagem baseada em reconhecimento de formas, especialmente voltada a este propósito. Sob esta abordagem podem ser encontrados diversos controladores em fase de desenvolvimento, tais como [Litt 91] e [Ruano 92], e outros já implementados, como [Abreu 92], [Porter 87] e [Kraus 84] sendo este último disponível comercialmente.

O desenvolvimento de um controlador adaptativo com vistas a atender a faixa mais ampla possível de casos, deve levar em consideração algumas características apresentadas por certos conjuntos atuador-processo-medidor, tais como saturação, fase não mínima, atrasos de transporte e não-linearidades.

Estas características podem comprometer o desempenho do controlador adaptativo, e em certos casos tornar o sistema instável. Para contornar tais situações e facilitar a operação destes equipamentos, são incluídos mecanismos de apoio e supervisão.

Um dos principais mecanismos de apoio ao controle adaptativo é o de pré-ajuste, responsável por obter, de forma automatizada, as informações a respeito do processo, necessárias para o controlador em questão. Estas informações podem ser fornecidas pelo usuário, mas de uma maneira geral o grau de conhecimento exigido, a respeito do processo, demanda a realização prévia de ensaios de identificação.

Existem diversos métodos de identificação de processos que podem ser automatizados, implementando desta forma, o mecanismo de pré-ajuste. Provavelmente os métodos mais conhecidos e utilizados sejam os de ajuste inicial de controladores do tipo PID, propostos por Ziegler e Nichols [Ziegler 42], os quais servem de base a aperfeiçoamentos realizados tanto para o caso de resposta ao degrau em malha aberta [Pagano 91], como para o caso de oscilação crítica [Åström 84].

Outro mecanismo bastante importante é o de planejamento de ganho, o qual fornece grande auxílio no controle de processos com características não-lineares. Este planejamento de ganho pode estar incluído no pré-ajuste, de forma a compensar características não-lineares do conjunto atuador-processo-medidor, bem como ser projetado para armazenar informações relativas aos pontos de operação do sistema [Åström 89].

Com o objetivo de aumentar a robustez e melhorar o desempenho do controlador, podem ser incluídos mecanismos de supervisão, na implementação dos controladores adaptativos. Estes mecanismos atuam em condições de alarmes, limites de operação, situações de instabilidade e outras que puderem ser detectadas. Nem sempre as informações provenientes do sistema podem ser utilizadas na adaptação, cabendo ao mecanismo de supervisão uma pré-análise da informação e uma validação dos novos parâmetros do controlador [Hägglund 91].

Geralmente os controladores adaptativos possuem variáveis de definição do desempenho global do sistema, seja na forma de valores absolutos, como por exemplo sobrepasso e tempo de resposta, ou valores difusos, como resposta rápida ou lenta. Estas variáveis, independentes do processo sob controle, iniciam com valores "default", mas podem ser alterados pelo usuário em função da necessidade de um desempenho específico. Em [Åström 89] e [Wellstead 86] podem ser encontradas descrições de alguns controladores adaptativos existentes, onde se observa a existência destas variáveis.

Em conjunto com os mecanismos de apoio e supervisão os controladores adaptativos podem conter outros detalhes de implementação, tais como filtros e níveis de segurança, que os qualifiquem a operar em situações reais, frente a ruídos e adversidades. A integração destes elementos apresentados, com um algoritmo de controle adaptativo, constitui uma implementação de controlador adaptativo de propósito geral, o qual consistirá em mais uma opção ao usuário interessado em controlar seu(s) processo(s) variante(s) no tempo.

2.4. AVALIAÇÃO DE DESEMPENHO

Em função da multiplicidade de opções (abordagens e mecanismos) em termos de controladores adaptativos, surgem certas dúvidas quanto ao desempenho relativo destes

equipamentos. Estas dúvidas podem ser formuladas através de perguntas tais como: Qual a melhor opção ao controle de determinado processo? Qual equipamento fornece melhor desempenho frente a um certo conjunto de processos? Em quais situações fracassa?

Apesar da avaliação de desempenho de controladores adaptativos de propósito geral ser o principal enfoque neste trabalho, a questão da avaliação faz parte dos diversos contextos do controle adaptativo.

Inicialmente pode-se pensar em avaliar as próprias abordagens ao controle adaptativo, o que permite apenas uma análise qualitativa, visando identificar, de uma forma genérica, suas principais qualidades e limitações. Em função desta análise pode-se ter uma previsão quanto ao esforço necessário para contornar suas limitações, através dos mecanismos de apoio e supervisão.

Uma avaliação mais aprofundada pode ser realizada sobre algoritmos de controle adaptativo, possibilitando análises qualitativas e quantitativas. Com base neste estudo já é possível obter as primeiras conclusões quanto à aplicabilidade e desempenho relativo destes algoritmos, frente a um tipo de processo sujeito a um conjunto de situações.

Avançando um pouco, chega-se ao caso da avaliação de uma implementação de controlador adaptativo de caráter específico, ou seja, um produto integrado contendo o algoritmo de controle, um mecanismo de adaptação e, opcionalmente, mecanismos de apoio e supervisão. Neste caso o objeto de estudo é projetado para controlar um processo específico, portanto a avaliação deve ser realizada sobre este processo. Algumas situações críticas não especificadas no projeto podem ser incluídas na avaliação, de forma a torná-la algo mais que um simples teste de funcionamento.

Finalmente retorna-se ao caso principal que é a avaliação de desempenho de controladores adaptativos de propósito geral. Em função de sua complexidade e abrangência, este caso merece especial atenção, sendo abordado nas seções que se seguem. Salienta-se que a discussão a respeito da avaliação limitar-se-á ao desempenho operacional dos controladores, apesar de que questões tais como o custo, a facilidade de integração ao meio industrial, entre outras, sejam bastante relevantes em estudos comparativos.

2.4.1. OBJETIVOS DA AVALIAÇÃO

A realização de uma avaliação está sempre relacionada a uma tomada de decisão, seja esta corretiva, no sentido de aperfeiçoar uma estrutura adaptativa, ou definitiva, implicando na escolha de uma alternativa frente a outra(s). Desta forma podem ser identificados alguns objetivos para a realização da avaliação de desempenho de controladores adaptativos de propósito geral:

- um usuário que deseja adquirir um controlador adaptativo, para atuar sobre determinado processo, estará interessado em uma avaliação relativa destes equipamentos, frente à situação real de operação.
- em uma indústria são diversos os processos a serem controlados, sendo economicamente mais interessante a aquisição de um único modelo de controlador adaptativo. Portanto a avaliação deve indicar a opção mais apropriada a este conjunto de processos.
- um fabricante de controladores adaptativos pode estar interessado em realizar comparações de desempenho de seu equipamento com os dos concorrentes. Neste caso podem ser definidos um "benchmark" (experimento padronizado) ou um amplo conjunto de processos e situações.

Além destes casos citados podemos encontrar diversos outros motivos para avaliar controladores adaptativos de propósito geral, porém a forma de realização desta avaliação será uma das apresentadas abaixo:

- **Um controlador × um processo** : este caso deve ser interpretado como a avaliação de desempenho de diversas versões de um controlador adaptativo em desenvolvimento, atuando sobre um processo. Este processo deve possuir características bastante variadas de forma a abranger um amplo conjunto de processos reais. Experimentos com estas características, denominados "benchmarks", costumam ser bastante detalhados de forma a explorar diversas situações de operação.
- **Um controlador × diversos processos** : novamente realiza-se a avaliação de um controlador em desenvolvimento, mas sobre um amplo conjunto de processos. Este caso aparenta ser mais indicado ao desenvolvimento de um controlador adaptativo de propósito geral do que o anterior, em função de sua abrangência. Os experimentos não são tão elaborados e a avaliação baseia-se principalmente em parâmetros globais, como o sucesso no controle e a velocidade de adaptação.
- **Diversos controladores × um processo** : este tipo de avaliação é utilizado para confrontar diversos controladores adaptativos com um "benchmark" proposto, geralmente com objetivos de "marketing" do produto, ou com o processo para o qual deseja-se adquirir um controlador adaptativo. Um exemplo de "benchmark" voltado à comparação de diversos controladores pode ser encontrado em [M'Saad 91].
- **Diversos controladores × diversos processos** : este tipo de avaliação é utilizado para confrontar diversos controladores adaptativos com um conjunto de processos. Este conjunto de processos pode ser determinado pelo usuário interessado em descobrir qual a melhor opção à sua indústria, ou pelo fabricante interessado em analisar o comportamento relativo de seu produto.

Convém salientar que as duas primeiras formas descritas podem ser vistas como casos particulares das posteriores, mas mereceram destaque em função de estarem voltadas ao desenvolvimento de controladores adaptativos, enquanto as últimas estão voltadas à comparação de estruturas adaptativas.

2.4.2. PARÂMETROS DE DESEMPENHO

No contexto deste trabalho o termo "avaliar" assume o significado de "apreciar o valor" [Ferreira 75], seja este qualitativo ou quantitativo. Desta forma a avaliação de desempenho de um controlador adaptativo pode ser entendida como sua apreciação, de forma a determinar um valor, qualitativo ou quantitativo, relativo a seu desempenho.

Devido à complexidade inerente aos controladores adaptativos, a determinação de um valor que caracterize seu desempenho não é atividade de fácil realização. Diversos parâmetros podem ser considerados nesta apreciação, sendo o valor final determinado a partir de uma ponderação destes.

Alguns parâmetros, considerados como os mais importantes na avaliação de desempenho, são apresentados a seguir:

- a) sucesso na realização do controle;
- b) velocidade de adaptação;
- c) qualidade da resposta;
- d) consumo de energia e matéria-prima;
- e) dificuldade de obtenção das informações a priori;
- f) tempo necessário para o pré-ajuste.

O sucesso na realização do controle é uma condição básica para aceitação do controlador adaptativo, bem como no prosseguimento da avaliação. Para identificar a obtenção ou não deste sucesso, faz-se necessária uma definição clara dos objetivos de controle.

A velocidade de adaptação é uma indicação do tempo necessário para que o controlador ajuste seus parâmetros em função de uma variação nos parâmetros do processo. Pode-se expressar esta velocidade através de alguma medida de tempo (segundos, minutos, horas, etc.), ou em termos da quantidade de excitações exógenas (mudanças de referência ou ocorrências de perturbação) necessárias para que o controlador possa ser considerado como ajustado à atual dinâmica do processo.

A qualidade da resposta é relativa às especificações do sistema. Critérios da integral do erro, tais como ISE (integral do erro quadrático) e ITAE (integral do erro absoluto multiplicado pelo tempo), podem ser utilizados, mas as restrições quanto a sobrepasso máximo, tempo de resposta, etc. devem ser respeitados.

O consumo de energia pode ser indiretamente avaliado através da integral do sinal de controle elevado ao quadrado (ISU), a qual fornece um valor proporcional a este consumo. Outros tipos de matéria-prima podem ter seus consumos indiretamente medidos através da integral do sinal de controle (IU), como por exemplo o consumo de ácido em uma malha de controle de pH.

Certamente a dificuldade de obtenção das informações necessárias à inicialização do controlador é uma grandeza representada de forma difusa e sujeita a critérios vagos. A quantidade destas informações pode servir de apoio nesta questão, apesar de não refletir, obrigatoriamente, a dificuldade em obtê-las.

O mecanismo de pré-ajuste visa facilitar a inicialização do controlador adaptativo, apesar de, para isto, consumir um certo tempo. Em alguns casos, onde é freqüente a inicialização do controlador, este parâmetro ("tempo despendido") influencia bastante no desempenho do controlador adaptativo.

A importância de cada um destes parâmetros, no resultado global, depende do objetivo da avaliação e do contexto no qual se insere o controlador. A velocidade de adaptação, por exemplo, é um parâmetro de importância relativa à freqüência com que o processo sofre variações em sua dinâmica. Outro fator que dificulta a ponderação destes parâmetros, na composição do índice de desempenho, é a integração de dados numéricos com dados difusos.

Com base nestas considerações, pode-se concluir que uma avaliação de desempenho deve ser planejada com bastante atenção devido à subjetividade inerente às ponderações, de forma a que se possa colher os resultados pretendidos. As questões da escolha dos parâmetros a serem considerados e da ponderação destes parâmetros são pontos de extrema importância à uma boa visualização do desempenho do controlador adaptativo.

2.4.3. AMBIENTES DE AVALIAÇÃO

Os controladores adaptativos de propósito geral podem ter seus desempenhos avaliados em função do retorno de informações provenientes dos usuários destes equipamentos. Um exemplo pode ser encontrado em [Åström 89], onde é realizada uma breve avaliação de alguns controladores adaptativos encontrados no meio industrial. Este tipo de avaliação não apresenta

resultados definitivos pois os controladores atuam em processos distintos e sob diferentes condições de operação, não havendo uma metodologia padrão para realização deste estudo.

Ainda no âmbito de processos reais, podem ser realizados estudos comparativos e avaliações de controladores adaptativos atuando sobre plantas piloto, como por exemplo em [Dumont 89]. Em função do custo bastante alto destas plantas, apenas alguns privilegiados centros de pesquisa podem realizar este tipo de estudo, a não ser que alguma indústria permita o acesso à seus processos. Apesar de poder ser definida uma metodologia para a realização da avaliação de desempenho, as plantas piloto apresentam-se de forma pouco flexível e sujeitas à certas condições externas que podem vir a atrapalhar a realização dos experimentos.

Os simuladores implementados em computadores analógicos e, principalmente, digitais, apresentam-se como uma solução bastante interessante no estudo avaliativo de controladores adaptativos. De custo bem mais acessível, estes simuladores são utilizados para treinamento de operadores [Hang 91], desenvolvimento [Caetano 89] e avaliação [Versteeg 86] de controladores adaptativos. A flexibilidade na modelagem da dinâmica dos processos, aliada a uma sistematicidade na realização dos experimentos, conferem aos simuladores um papel de importância no contexto do controle adaptativo.

2.5. CONCLUSÃO

Os avanços nos estudos teóricos de estabilidade dos algoritmos de controle adaptativo vêm ocorrendo de forma lenta, principalmente devido a suas características não lineares. A aceitação dos controladores adaptativos no meio industrial tem sido marcada muito mais pela confiança no funcionamento prático do que em termos de argumentos teóricos. Neste sentido os simuladores de sistemas de controle surgem como uma importante ferramenta no auxílio à validação, permitindo a simulação de processos com características bastante próximas às encontradas em sistemas reais, como por exemplo ruído, não linearidades e atrasos de transporte.

Apesar de todas estas características apresentadas pelos simuladores, os estudos de avaliação encontrados na literatura carecem de alguma(s) das seguintes condições: sistematicidade dos testes, apresentação de resultados quantitativos e testes sobre um amplo universo de processos [Kammer 92b].

Um ambiente de simulação especialmente voltado à avaliação de controladores adaptativos é apresentado nesta dissertação. O principal objetivo deste ambiente é possibilitar a realização de avaliações sistematizadas, quantitativas e sobre amplos conjuntos de processos. As potencialidades e os mecanismos previstos para o ambiente são apresentados, no capítulo seguinte, através de sua especificação funcional.

CAPÍTULO 3 : ESPECIFICAÇÃO FUNCIONAL

3.1. INTRODUÇÃO

A partir deste capítulo começa a ser abordado o desenvolvimento de um ambiente de simulação para sistemas de controle, voltado à avaliação de controladores adaptativos. Este desenvolvimento subdivide-se nas etapas de projeto, execução e testes. No presente capítulo é apresentada a primeira destas etapas sob a forma de uma especificação funcional.

Em função de experiências anteriores no desenvolvimento de pacotes de simulação para sistemas de controle, a nível de LCMI, tais como SAPIC [Savi 87], SDPID [Caetano 89], SIMAP [Da Luz 90] e ISAC [Lima 91], pode-se afirmar que certas características como flexibilidade e modularização do código devem ser priorizadas, inclusive além do fator tempo, durante a confecção do software. Estas características começam a ser delineadas na própria especificação do ambiente, requerendo especial atenção.

A abordagem pretendida para a avaliação de controladores adaptativos é essencialmente prática, portanto não está prevista a implementação de ferramentas de análise e projeto. Esta avaliação será realizada com dados provenientes do controlador e dos sinais de entrada e saída do processo, o que ressalta a importância de serem criadas, através de simulações, situações bastante semelhantes às encontradas em ambientes reais.

Com base nestas linhas mestre é realizada a especificação do ambiente, a qual encontra-se descrita neste capítulo. Inicialmente, na seção 3.2, apresenta-se uma especificação informal do ambiente de forma a definir, de uma forma simples e geral, suas potencialidades. De um ponto de vista estrutural o ambiente será dividido em estrutura básica e estrutura de avaliação, sendo estas apresentadas nas seções 3.3 e 3.4, respectivamente. Algumas características da interface homem-máquina são apresentadas na seção 3.5, cabendo à seção 3.6 uma apresentação dos modos de operação do ambiente.

3.2. ESPECIFICAÇÃO INFORMAL

Enquanto principal objetivo do ambiente, a questão da avaliação de desempenho de controladores adaptativos é quem deve definir a espinha dorsal durante o desenvolvimento deste pacote computacional. Para tanto deve-se ter uma definição bastante clara quanto ao que se pretende em termos de avaliação.

Conforme discutido no capítulo 2, o termo **avaliação**, neste trabalho, assume o significado de "apreciação de valor", objetivando o auxílio à tomada de decisões, que poderão ser

corretivas, quando associadas ao desenvolvimento de uma dada estratégia de controle, ou definitivas, quando da escolha de uma estratégia entre várias. Assim, o problema mais geral a ser tratado consiste na avaliação de várias estratégias de controle aplicadas a um conjunto amplo de processos sujeitos a eventos que justifiquem a necessidade de uso da abordagem adaptativa (não-linearidades, parâmetros variantes no tempo, etc.). É evidente que qualquer um dos casos descritos em 2.4.1 poderá também ser estudado, utilizando-se o ambiente.

A "apreciação de valor" do desempenho do controlador adaptativo será realizada através de índices (informações) obtidos via experiências simuladas. Cada experiência consiste na simulação da atuação do controlador adaptativo sobre um determinado processo monovariável.

Considerando o acima exposto, o ambiente a ser desenvolvido deverá conter:

- a) Uma estrutura básica de simulação dinâmica da realidade prática de processos controlados por estratégias adaptativas, com as seguintes características:
 - a.1) para o processo: possibilidade de inclusão de ruído, perturbações aditivas, não-linearidades e perturbações paramétricas, entre outras.
 - a.2) para o controlador adaptativo: estágios de conversão analógico/digital (A/D) e digital/analógico (D/A) com resolução variável e mecanismos que permitam uma ampla liberdade de implementação de estratégias adaptativas.

A simulação de um dado experimento, que deverá poder ser realizada manual (com a intervenção do operador) ou automaticamente (pré-programada), será gerida por um supervisor responsável por tarefas tais como inicialização e configuração do controlador, mudanças dos sinais de referência e perturbação, mudança nos parâmetros da planta, etc.

- b) Uma estrutura de avaliação responsável por: executar de forma sistemática (automática) um amplo conjunto de experimentos, podendo envolver uma estratégia adaptativa e vários processos; obter as informações relativas às simulações que digam respeito ao processo avaliativo e, finalmente, elaborar relatórios "pormenorizados" de avaliação.
- c) Uma interface homem-máquina adaptada ao problema, com recursos gráficos de qualidade que permitam o acompanhamento das simulações, pelo operador; especialmente importante quando do desenvolvimento de uma estratégia adaptativa.

3.3. ESTRUTURA BÁSICA

Como em qualquer simulador para sistemas de controle, a estrutura básica do ambiente será composta pelos módulos "controlador", "processo" e "supervisor", sendo que nesta primeira versão haverá uma instância de controlador atuando sobre uma instância de processo, ambas monovariáveis, sob comando do módulo "supervisor", conforme apresentado na figura 3.1. As características destes módulos são apresentadas nas seções que se seguem.

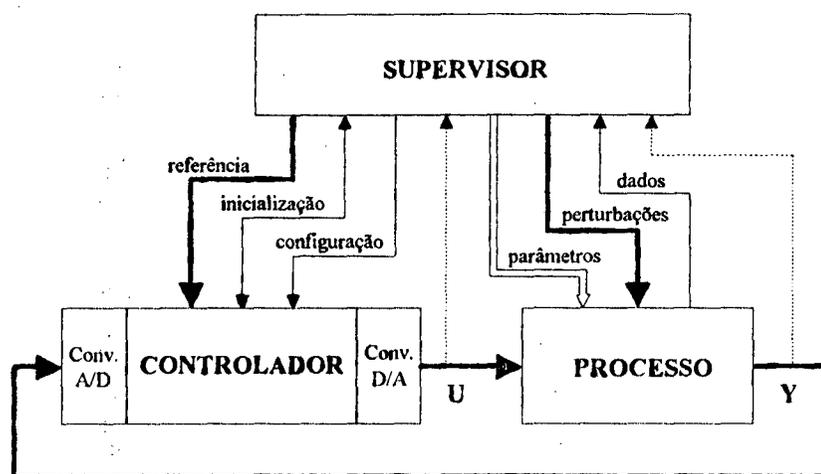


Fig. 3.1 - Estrutura Básica do Ambiente de Simulação

3.3.1. MÓDULO CONTROLADOR

De acordo com a filosofia básica de realizar um ambiente de simulação voltado à realidade prática, o módulo "controlador" deverá simular a atuação de conversores A/D e D/A em sua entrada e saída, respectivamente, uma vez que o computador é a principal ferramenta para implementação de controladores adaptativos [Åström 89].

Considerando-se a ampla gama de abordagens e implementações de controladores adaptativos existentes e que poderão vir a ser desenvolvidos, este módulo deverá prover os mecanismos básicos necessários para que o usuário possa programar sua estratégia adaptativa. De forma a viabilizar esta atividade de programação a um amplo conjunto de usuários, faz-se necessária a modularização do código e definição de interfaces claras e consistentes, principalmente entre o módulo "controlador" e os demais. Estas considerações conduzem à utilização de modernas técnicas de programação orientada a objetos, proporcionando ao sistema poderosos recursos no sentido de atingir características importantes, tais como extensibilidade, reusabilidade e compatibilidade [Meyer 88].

Além da função de controle propriamente dita, duas funções especiais, específicas para cada controlador, farão parte da interface deste módulo: a *configuração* e a *inicialização*. A função *configuração* será utilizada para definir valores às variáveis relativas ao desempenho global do sistema (ver seção 2.3), independentemente ao processo sob atuação. Por outro lado a função *inicialização* encarregar-se-á de obter as informações a respeito do processo, necessárias ao início das atividades de controle, seja através de interação com o usuário ou via mecanismo de pré-ajuste.

3.3.2. MÓDULO PROCESSO

A complexidade da estrutura do processo deve atender a um compromisso entre o potencial de representação e o custo (tempo e esforço) necessário à sua implementação. Desta forma será especificada, como uma versão inicial, uma estrutura para o processo constituída por blocos em cascata, sem realimentações internas, de forma a simplificar os algoritmos de simulação. Serão implementados diversos tipos de blocos, onde cada tipo estará encapsulado por um objeto genérico "bloco", permitindo a realização de um procedimento de utilização comum a todos os blocos. Desta forma a seqüência e a quantidade dos blocos que irão compor o processo serão definidas pelo usuário, durante a fase de operação com o ambiente. A figura 3.2 apresenta esta estrutura do processo composta por blocos em seqüência.

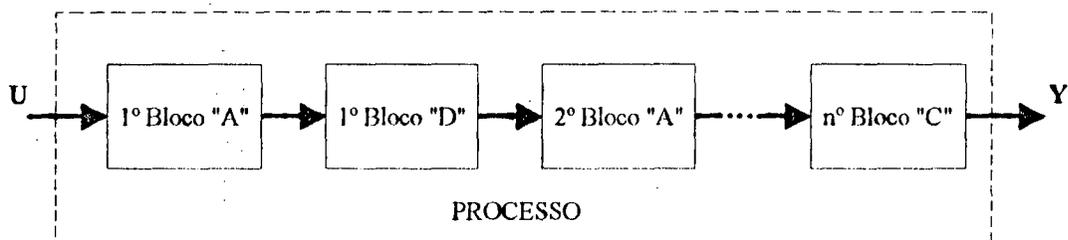


Fig. 3.2 - Estrutura do Processo

A princípio serão implementados os seguintes tipos de blocos: dinâmica linear contínua descrita por função de transferência; dinâmica linear contínua descrita por equações de estado; dinâmica linear discreta descrita por função de transferência; atraso de transporte; curva de ganho não linear e ponto de entrada das perturbações e ruído. Em função da modularização do objeto "bloco", tornar-se-á possível uma futura inclusão de novos tipos de blocos, como por exemplo de dinâmicas não lineares, caso haja necessidade de modelar processos com características diferentes destas.

Por ser este um ambiente de simulação voltado ao controle adaptativo, o módulo "processo" deve fornecer mecanismos para alteração de parâmetros do modelo do processo, durante a realização de experimentos. Assim os blocos de dinâmica linear (contínua e discreta) e

de atraso de transporte conterão parâmetros alteráveis, simulando, desta forma, o comportamento de processos variantes no tempo.

Os tipos de sinais de perturbação que atuarão sobre o processo são: degrau, rampa, pulso e senóide, além do sinal de ruído.

Cabe salientar que a questão da estrutura do processo a ser controlado pode ser analisada sob o mesmo enfoque dado ao módulo "controlador". Com isto será definido um objeto genérico "processo" e sua interface, podendo a estrutura implementada ser substituída por outra que atenda às necessidades de usuários específicos, ou por uma mais complexa que englobe o potencial de representação da estrutura anterior.

3.3.3. MÓDULO SUPERVISOR

A principal função do módulo "supervisor" é controlar o fluxo de informações e comandos, durante a realização de simulações. Estas simulações apresentar-se-ão de duas formas distintas, de acordo com o objetivo pretendido: desenvolvimento ou avaliação.

Em uma simulação voltada ao desenvolvimento de um controlador adaptativo, os comandos de alteração no processo, controlador e/ou excitações serão realizados pelo usuário, via interface homem-máquina, enquanto que em uma simulação voltada à avaliação, os comandos de alteração serão realizados por entidades da estrutura de avaliação, conforme será descrito na seção 3.4.

Apesar destas diferenças, o módulo "supervisor" conterà todas as funções necessárias ao controle de ambas as formas de simulação. Estas funções são: a) transferir os comandos de alteração para os módulos "controlador" e "processo"; b) coletar dados a respeito do processo; c) controlar o fluxo de informações (sinais simulados) entre estes módulos e d) fornecer estas informações à estrutura de avaliação, no caso de uma simulação voltada à avaliação de um controlador adaptativo.

3.4. ESTRUTURA DE AVALIAÇÃO

A estrutura de avaliação será responsável por fornecer, ao usuário, os recursos necessários à realização de avaliações de desempenho de controladores adaptativos, em especial aos de propósito geral. Esta estrutura fará uso dos módulos que compõem a estrutura básica (controlador, processo e supervisor), adicionando a estes, dois novos módulos: o "executor" e o "avaliador", conforme apresentado na figura 3.3.

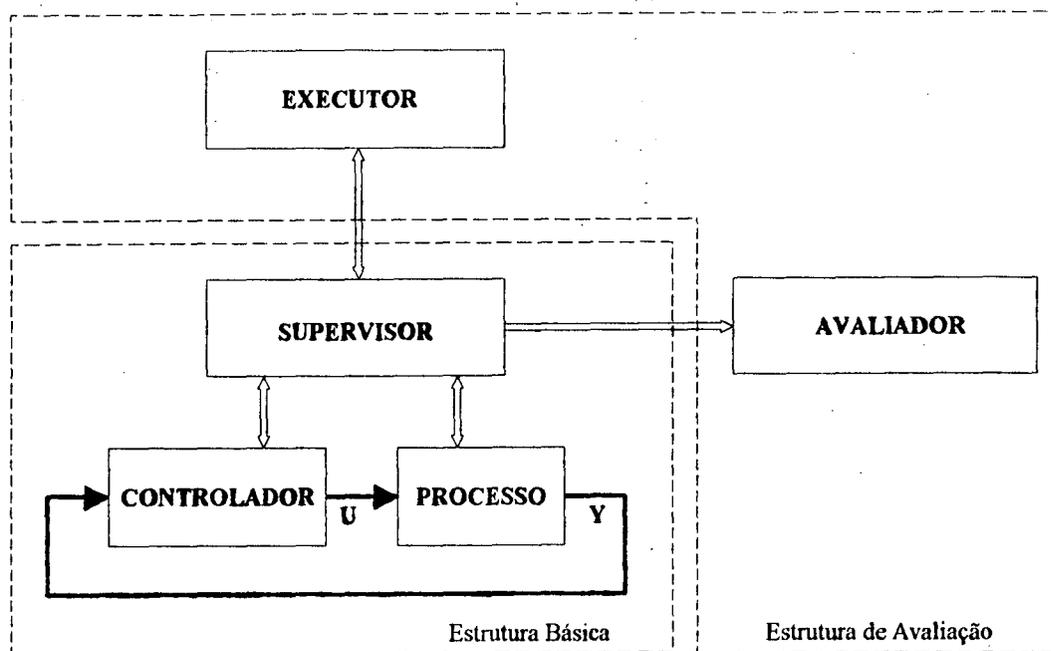


Fig. 3.3 - Estrutura Completa do Ambiente de Simulação

O módulo "avaliador" reunirá todas as informações envolvidas com a simulação, que sejam úteis à avaliação. O módulo "executor" irá fornecer os mecanismos necessários a uma avaliação de desempenho sistemática, quantitativa e sobre um amplo conjunto de processos. Estes módulos são descritos nas seções que se seguem.

3.4.1. MÓDULO AVALIADOR

Entendendo a avaliação de desempenho como a obtenção de um valor, qualitativo ou quantitativo, que caracterize este desempenho, e que este valor é determinado através de uma ponderação de diversos parâmetros a serem considerados, o módulo "avaliador" deverá ser capaz de fornecer, ao usuário, um conjunto de parâmetros, obtidos via simulação, com os quais seja possível realizar avaliações de desempenho de controladores adaptativos.

O ambiente será responsável por fornecer os parâmetros que puderem ser expressos através de valores numéricos, deixando a obtenção dos dados difusos e a definição da ponderação a encargo do usuário. Os parâmetros a serem fornecidos pelo ambiente são: a velocidade de adaptação, integrais do erro e da ação de controle, e o tempo de inicialização.

Nesta primeira versão a velocidade de adaptação será medida em "ciclos de ajuste", sendo que cada ciclo corresponde a uma mudança de referência ou perturbação de carga. Cada controlador adaptativo deve possuir um mecanismo de supervisão capaz de identificar se o controlador encontra-se ajustado, frente ao ciclo ocorrido. Este mecanismo é bastante evidente em controladores baseados em reconhecimento de formas, pois a comparação da forma de onda

obtida com a padrão indica se o controlador já está ajustado. Para as demais abordagens este mecanismo não é evidente, mas também não é impossível de ser implementado, como por exemplo para um controlador MRAS pode-se verificar se a diferença entre a saída do processo e a do modelo não ultrapassa um certo valor limite, até que a resposta atinja o regime permanente, frente a uma mudança de referência na forma de degrau.

Uma avaliação da qualidade do sinal de saída do processo pode ser realizada com auxílio de critérios tais como os da integral do erro. O ambiente de simulação deve fornecer mecanismos para medição dos índices ISE e ITAE. O índice ISE é mais utilizado devido à facilidade de manuseio analítico, enquanto que o índice ITAE fornece uma resposta com melhores características, quando minimizado [Ogata 70]. Estes índices fornecem valores absolutos que individualmente nada contribuem para a avaliação, devendo portanto ser utilizados em análises comparativas, seja entre controladores diferentes ou com um valor padrão.

Além dos índices baseados no erro, o ambiente deve fornecer os índices IU e ISU, baseados no sinal de controle e indicativos do consumo de matéria-prima e/ou energia. De forma semelhante aos índices anteriores, estes devem ser utilizados em análises comparativas.

Para que um controlador adaptativo esteja preparado para atuar sobre determinado processo, faz-se necessária uma etapa de inicialização, na qual são fornecidas certas informações a respeito deste processo. Se este controlador contiver um mecanismo de pré-ajuste, reduz-se a quantidade e/ou a dificuldade de obtenção destas informações, por parte do usuário, apesar de haver um certo consumo de tempo nesta atividade. A nível do ambiente de simulação, o módulo "avaliador" deverá prover mecanismos para medir o tempo consumido durante a inicialização.

Todas estas informações serão apreciadas pelo módulo "avaliador", o qual irá gerar relatórios ao final de cada experimento, inserindo ainda informações provenientes do controlador adaptativo sob teste, tanto para a etapa de inicialização quanto durante a operação normal. Com base nas informações destes relatórios podem ser realizados a avaliação do controlador adaptativo e/ou estudos de suas características com vistas ao aperfeiçoamento.

A forma de avaliação dos controladores adaptativos não é uma questão fechada, razão pela qual o módulo "avaliador" também será definido como um objeto genérico, possuindo uma interface fixa, mas cuja implementação pode ser substituída. Alia-se a isto o fato de que novas estratégias de controle adaptativo, que vierem a ser implementadas, poderão conduzir a formas de avaliação mais apropriadas.

3.4.2. MÓDULO EXECUTOR

De forma a tornar a avaliação de desempenho sistemática, os eventos que irão compor a simulação serão pré-programados em arquivos denominados "**seqüência de eventos**". Estes eventos englobam os comandos da estrutura básica, bem como mecanismos para obtenção dos parâmetros de avaliação.

Para permitir a avaliação de um controlador frente a um conjunto de processos, define-se a entidade "**banco de provas**". Um "banco de provas" é composto por um conjunto de processos e um conjunto de "seqüências de eventos" a serem executadas sobre estes processos. Cabe ao módulo "executor" comandar a simulação da atuação de um controlador adaptativo sobre um "banco de provas".

3.5. INTERFACE HOMEM-MÁQUINA

A interface homem-máquina deverá prover meios para que o usuário interaja com o ambiente, de uma forma o mais completa e agradável possível. Esta interface compreende a entrada e saída de dados numéricos, sejam estes declarativos ou informativos, bem como apresentação de gráficos.

A entrada de dados será realizada através de campos que estarão apresentando os valores atuais para os itens em questão, podendo estes serem alterados pelo usuário caso haja necessidade. Os demais detalhes, como a distribuição destes campos na tela e a forma de acesso a estes, serão definidos durante a etapa de implementação do ambiente, quando já estiverem definidos os recursos de hardware e software a serem utilizados.

Devido às características do ambiente, haverá a necessidade de acompanhamento contínuo dos sinais envolvidos com a simulação (saída do processo, referência e saída do controlador), por parte do usuário. Desta forma deverão ser implementados registradores gráficos que permitam uma apresentação contínua dos sinais, com escalas alteráveis. Os gráficos das simulações deverão poder ser armazenados em arquivo, para uma posterior análise, a qual compreende ampliações de regiões e visualização dos valores de amplitude e tempo dos pontos que compõem estes gráficos.

3.6. MODOS DE OPERAÇÃO

Em termos de operação, o ambiente proposto dividir-se-á em três modos distintos: declaração, simulação manual e simulação automática. O primeiro modo conterà mecanismos para realizar a declaração dos componentes do sistema, ou seja, controlador e processo. Durante o

modo de simulação manual será realizada a evolução da dinâmica do processo, em função de atuações do controlador e excitações externas, sob comando do usuário. No último modo de operação a simulação será realizada através da execução de um "banco de provas", visando a avaliação do controlador adaptativo. Uma especificação mais detalhada destes modos de operação é realizada nas seções que se seguem.

3.6.1. DECLARAÇÃO

A utilização do ambiente de simulação deverá iniciar com uma fase na qual serão definidas as estruturas dos componentes do sistema e atribuídos valores a seus elementos internos. Esta etapa, denominada "declaração", abordará os módulos "controlador" e "processo", conforme observado na figura 3.4. O módulo localizado do lado esquerdo, na figura, representa os objetos envolvidos com a declaração do controlador, enquanto que o módulo localizado à direita representa os objetos envolvidos com a declaração do processo.

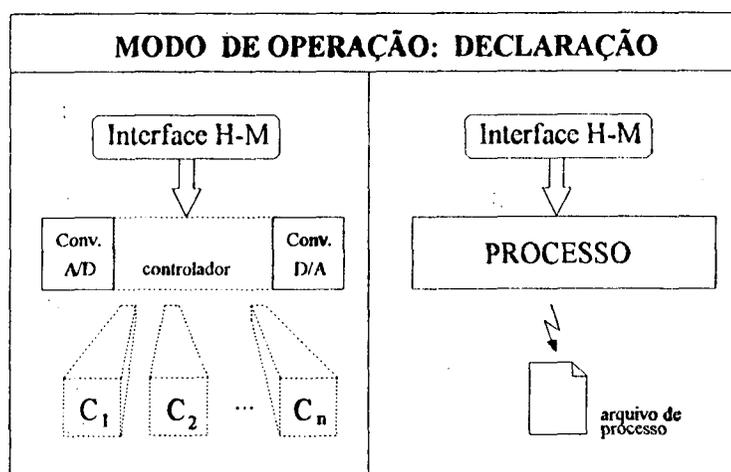


Fig. 3.4 - Diagrama dos Módulos da Declaração

Observa-se a presença dos conversores A/D e D/A, cujas especificações (limites de conversão, resolução e tempo de amostragem) serão determinadas nesta fase do programa. Além disto deverá ser possível selecionar um dentre os controladores disponíveis (C_1, C_2, \dots, C_n).

A declaração do processo envolve a definição da seqüência dos blocos componentes e posterior determinação de valores para os parâmetros de cada bloco. As instâncias de processo, após declaradas, poderão ser gravadas em arquivos, de forma a que se possa compor os conjuntos de processos dos "bancos de provas".

3.6.2. SIMULAÇÃO MANUAL

Após realizadas as declarações do controlador e do processo, procede-se à simulação do sistema. Esta simulação poderá ser feita de forma manual ou automática, sendo a primeira voltada ao desenvolvimento de um controlador adaptativo e a segunda à avaliação deste.

Durante a fase de desenvolvimento de um controlador adaptativo, realiza-se testes de forma individual e sob cuidados do implementador, que busca corrigir eventuais problemas e/ou analisar limites de aplicabilidade para este controlador. A forma de simulação manual será apropriada para este caso, pois nesta o usuário poderá acompanhar, continuamente, a evolução, no tempo, dos sinais do sistema, e atuar sobre ele no momento que desejar.

A figura 3.5 apresenta os objetos envolvidos com a simulação manual.

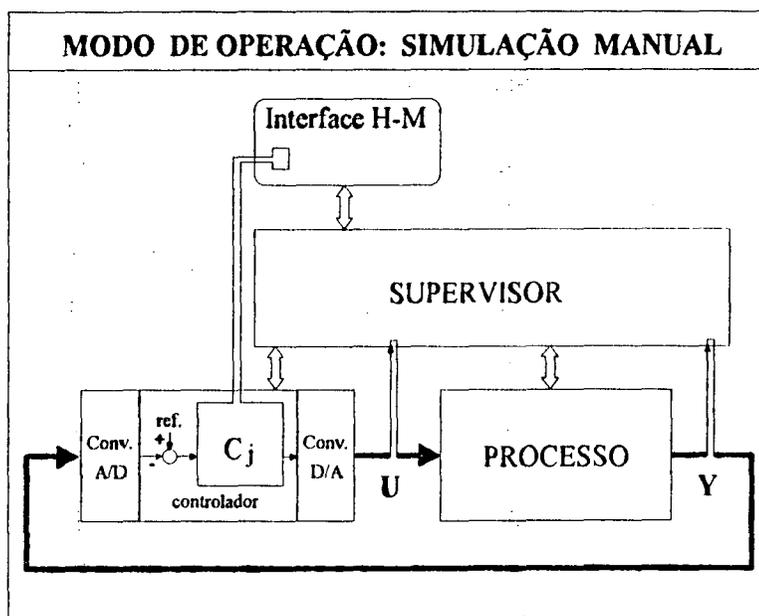


Fig. 3.5 - Diagrama dos Módulos da Simulação Manual

A interface homem-máquina conterá uma área para entrada de dados do controlador, relativos às funções de inicialização e configuração, e que por ser específica de cada instância de controlador adaptativo, deverá ser implementada em conjunto com este.

O acompanhamento da simulação será realizado através de registradores gráficos que continuamente apresentarão os sinais do sistema, ao usuário.

Através da interface homem-máquina o usuário poderá, também, comandar (iniciar, pausar/prosseguir, interromper) a simulação e atuar sobre o sistema, ambos sob coordenação do módulo "supervisor". Esta atuação poderá ocorrer a qualquer momento durante a simulação, e

consistirá de: alterações em parâmetros do processo, chamada às funções de inicialização e configuração do controlador e alteração nas excitações (referência, perturbação e ruído).

3.6.3. SIMULAÇÃO AUTOMÁTICA

A avaliação de um controlador adaptativo será realizada mediante execução de um determinado procedimento pré-estabelecido, verificando-se o comportamento do controlador em relação a um processo ou conjunto de processos. Esta repetição do procedimento de teste pode ser bastante cansativa, além de estar sujeita a falhas, quando realizada por operadores humanos. A forma de simulação automática utilizará os mecanismos da estrutura de avaliação para resolver este problema, automatizando a realização da avaliação e tornando-a bem mais eficiente em termos de tempo despendido.

Através da figura 3.6 é possível observar o diagrama de conexão entre os módulos que compõem a simulação automática. Este diagrama é bastante semelhante ao da fase de simulação manual, e os módulos "supervisor", "controlador" e "processo" permanecem os mesmos. A linha tracejada do módulo "processo" indica que diferentes instâncias de processo serão utilizadas durante a execução do banco de provas.

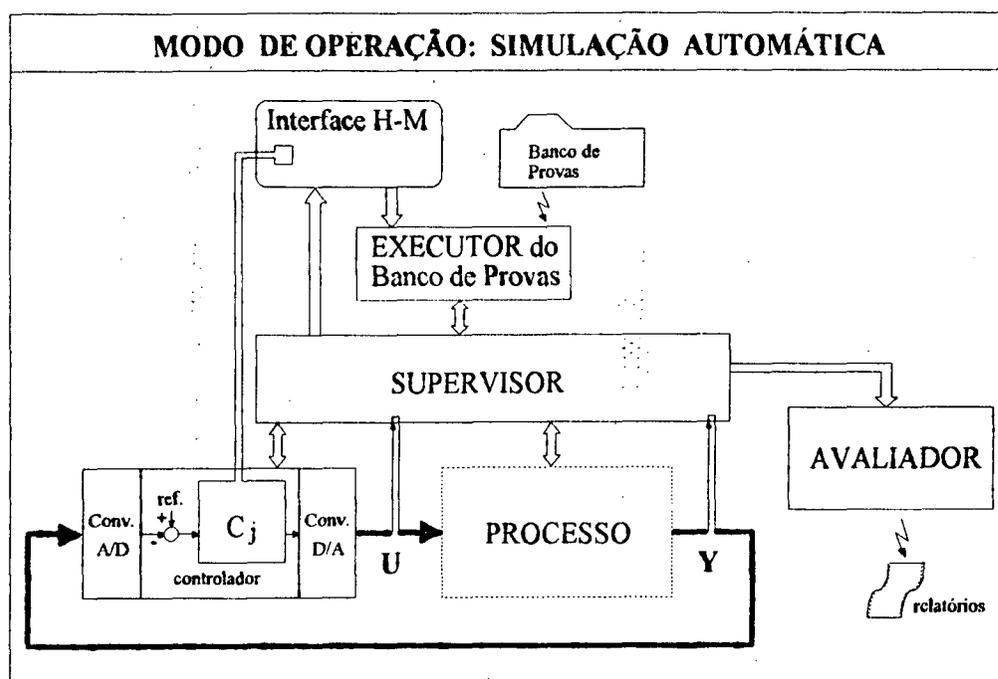


Fig. 3.6 - Diagrama dos Módulos da Simulação Automática

Ao final das experiências serão gerados relatórios contendo os resultados da avaliação. Nestes relatórios poderão estar inseridas mensagens provenientes do controlador, caso este recurso esteja implementado no controlador adaptativo sob avaliação.

3.7. CONCLUSÃO

Em função da especificação funcional, apresentada no presente capítulo, pode-se ressaltar duas características importantes a serem consideradas durante a etapa de execução deste projeto: a) uma modularização do código, com especial atenção aos mecanismos de suporte à implementação de novas estratégias de controle adaptativo, e b) recursos gráficos de qualidade que permitam um eficiente (e agradável) acompanhamento da evolução dos sinais simulados.

Apresentadas as características gerais pretendidas para o ambiente, procede-se à implementação deste produto, onde são tomadas as decisões à respeito dos detalhes internos dos elementos componentes do sistema.

CAPÍTULO 4 : IMPLEMENTAÇÃO

4.1. INTRODUÇÃO

A especificação do ambiente proposto, descrito no capítulo anterior, define o esqueleto da implementação deste pacote computacional, denominado SADECA (Sistema para Avaliação de DEsempenho de Controladores Adaptativos). Em função destas especificações, já podem ser tomadas as primeiras decisões quanto aos recursos computacionais a serem utilizados.

A nível de hardware optou-se pela utilização de estações de trabalho, em função de seus recursos gráficos e velocidade de processamento. Estes equipamentos operam com o sistema operacional UNIX*, no qual a linguagem C apresenta-se como a linguagem de programação "natural". Devido à "necessidade" de utilização de uma linguagem orientada a objetos, optou-se pela C++, a qual foi projetada como uma quase-extensão à linguagem C, mantendo as características de eficiência e portabilidade desta, mas agregando um tratamento mais seguro de tipos e facilidades para encapsulamento de dados [Takahashi 88].

Para implementar as interfaces de entrada e saída do programa utiliza-se o pacote gráfico XView** (padrão OPEN LOOK*), o qual fornece potentes recursos de interface gráfica em um sistema multijanelas (ambiente OpenWindows**). Uma vantagem de se utilizar uma interface padronizada é que os usuários se sentem confortáveis com as similaridades funcionais entre os vários aplicativos, e conseguem operar esta interface com maior tranquilidade [Sun 89].

O presente capítulo descreve a implementação do pacote SADECA, apresentando as principais características dos módulos componentes deste sistema. Devido à utilização de uma linguagem orientada a objetos considerou-se conveniente uma breve descrição de seus mecanismos básicos, de forma a permitir a utilização de uma terminologia mais apropriada. Esta descrição é realizada na seção 4.2.

A seção 4.3 apresenta detalhes de implementação do módulo "controlador", sendo esta apresentação um pouco mais detalhada do que a dos demais módulos, a nível de programação, devido à possibilidade que o usuário tem de implementar novos controladores adaptativos. A seguir apresenta-se, na seção 4.4, a implementação do módulo "processo", descrevendo-se nesta os diversos tipos de blocos que podem compor a estrutura do processo. Concluindo a estrutura básica, descreve-se, na seção 4.5, a implementação do módulo "supervisor".

* UNIX e OPEN LOOK são marcas registradas de AT&T.

** XView e OpenWindows são marcas registradas de SUN Microsystems, Inc.

A implementação da estrutura de avaliação é apresentada nas seções 4.6 e 4.7, através dos módulos "executor" e "avaliador", respectivamente. Nestas seções estão detalhadas as implementações das entidades "seqüência de eventos" e "banco de provas", além dos relatórios de avaliação. Finalmente apresenta-se na seção 4.8 alguns detalhes da poderosa interface homem-máquina presente neste ambiente de simulação.

É de suma importância ressaltar que os detalhes relativos à utilização deste ambiente estão descritos no **manual do usuário**, em [Kammer 92a] e no anexo I desta dissertação, não cabendo neste capítulo repetir estas informações, mas dar uma idéia global das potencialidades implementadas.

4.2. MECANISMOS BÁSICOS DE UMA LINGUAGEM ORIENTADA A OBJETOS

A descrição dos mecanismos e terminologias básicos das linguagens orientadas a objetos será realizada de forma genérica, apesar de em alguns momentos esta apresentação embasar-se em mecanismos da linguagem C++, por motivos óbvios, mas sem chegar ao nível de apresentar a sintaxe de programação.

O primeiro passo na apresentação de uma linguagem orientada a objetos consiste na compreensão do que é uma "classe". Na realidade a programação orientada a objetos (POO) é uma evolução da modularização do código, cujo princípio básico é dividir o programa em módulos o mais independentes possível. Neste sentido a POO visa agrupar estruturas de dados e suas funções correspondentes em unidades denominadas "classes". As funções de uma classe são denominadas "métodos" e instâncias de uma classe são chamadas de objetos [Eckel 91]. Como um exemplo podemos citar a classe retângulo, a qual conterà os dados comprimento e altura, e métodos de cálculo de perímetro, área, etc. Desta forma "retângulo" passa a ser considerado como um novo tipo de dado, e objetos deste tipo (classe) "retângulo" podem ser declarados e utilizados durante a programação.

Relembrando a especificação do ambiente de simulação, observa-se a estruturação deste composta por módulos variados, que interagem entre si. Um procedimento adequado à POO é fazer com que estes módulos sejam implementados através de classes. Desta forma a classe "processo", por exemplo, conterà a estrutura de dados e os métodos relativos ao módulo correspondente.

Outro aspecto importante a ser abordado é a "herança". Em uma linguagem orientada a objetos uma classe (classe derivada) pode ser declarada como herdeira de outra (classe base), passando a conter as características desta. Na classe derivada algumas características da classe base podem ser alteradas, assim como novas características podem ser acrescentadas. Pode-se

exemplificar este recurso criando as classes "retângulo" e "circunferência" herdeiras da classe "polígono". Com isto os objetos das classes "retângulo" e "circunferência" podem ser manipulados como sendo do tipo "polígono". A classe "polígono" conterá a declaração dos métodos de cálculo de perímetro e área, mas cujas implementações serão deixadas para as classes derivadas. Quando o método de cálculo de área (ou de perímetro) for chamado em um objeto do tipo "polígono", o sistema irá invocar a implementação do método correspondente na classe derivada a que pertence este objeto. Esta utilização de interfaces idênticas, com diferentes implementações, é denominada polimorfismo [Eckel 91].

Estas são os mecanismos básicos, necessários à apresentação da implementação do SADECA. Especial atenção deve ser dada à característica de reusabilidade do código a ser implementado, a qual ocupa posição central em Orientação a Objetos [Takahashi 88]. Para maiores detalhes quanto à POO, recorrer a [Meyer 88], e quanto à linguagem C++ cita-se [Stroustrup 86].

4.3. MÓDULO CONTROLADOR

No contexto deste ambiente, o módulo "controlador" é um exemplo típico de classe a ser implementada com os recursos de herança e polimorfismo. Este fato pode ser comprovado através de observação da figura 4.1, na qual apresenta-se a forma genérica de um controlador digital.

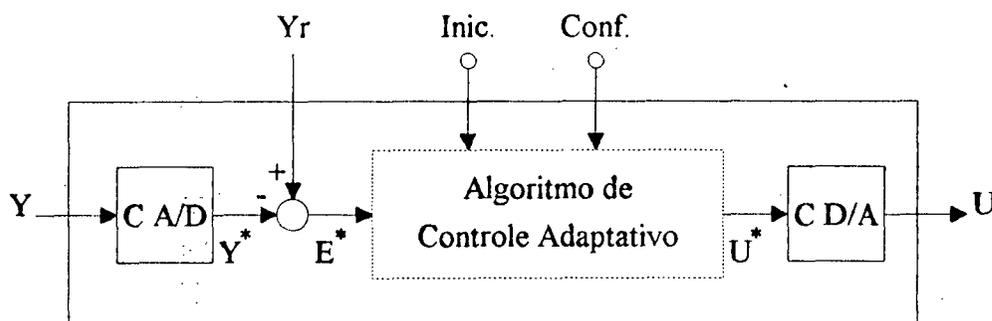


Fig. 4.1 - Controlador Adaptativo Genérico

Os elementos comuns aos controladores adaptativos fazem parte da classe base, como os conversores A/D e D/A. Cada novo controlador será implementado como uma classe derivada, herdando estes elementos comuns, mas redefinindo os métodos de cálculo da ação de controle ($U^*=f(E^*)$), de inicialização e de configuração.

O método de alteração do valor da referência (Y_r) pode, também, ser redefinido. Como exemplo pode-se citar os controladores baseados em reconhecimento de formas, que podem utilizar a informação de alteração de Y_r para iniciar o processo de reconhecimento.

Geralmente a inicialização e a configuração de um controlador adaptativo necessitam de informações a serem fornecidas pelo usuário. Para viabilizar esta entrada de dados, cada nova implementação de controlador adaptativo deverá criar painéis com os campos necessários (com o pacote XView), bem como realizar o tratamento destes dados.

Durante a execução de um banco de provas, o controlador será, provavelmente, inicializado pelo menos uma vez para cada instância de processo. Considerando-se a intenção de automatizar a avaliação do controlador adaptativo, eliminando a interação com o usuário, a inicialização do controlador deve estar preparada para assumir valores "default", o que não se torna tão crítico se forem implementados mecanismos de pré-ajuste (ver seção 2.3). Uma única informação é fornecida pelo módulo "supervisor", a respeito da dinâmica do processo a ser controlado: o tempo (aproximado) de resposta a um degrau em malha aberta, a qual pode ser utilizada, mas com critérios, nesta inicialização com valores "default".

Ainda com relação à inicialização do controlador, esta poderá, além de obter dados via usuário, atuar sobre o processo e observar o comportamento de sua saída, de forma a realizar a identificação deste (pré-ajuste). Informações a respeito da inicialização podem ser enviadas ao módulo "avaliador", passando estas a constar nos relatórios dos experimentos.

Com o objetivo de mensurar a velocidade de adaptação, o algoritmo de controle adaptativo deve poder identificar-se ajustado, ou não, quando questionado pelo módulo "supervisor". Esta comunicação é realizada através de duas funções: uma para início da etapa de ajuste e outra para a verificação do ajuste, ambas comandadas pelo "supervisor". Informações a respeito dos ciclos de ajuste podem, também, ser enviadas ao módulo "avaliador".

Um detalhamento completo dos procedimentos necessários à implementação de um novo controlador adaptativo pode ser encontrado no manual do usuário do ambiente (anexo I).

4.4. MÓDULO PROCESSO

A estrutura do processo é formada por uma seqüência de até dez blocos em cascata, os quais podem ser de qualquer um dos seguintes tipos: dinâmica linear contínua descrita por função de transferência (DLC_FT); dinâmica linear contínua descrita por equações de estado (DLC_EE); dinâmica linear discreta descrita por função de transferência (DLD); atraso de transporte (ATR); curva de ganho não linear (GNL) e ponto de entrada das perturbações e ruído (PRT). O tipo dinâmica linear discreta descrita por equações de estado não foi implementado, até o momento, simplesmente por não haver surgido necessidade. Após concluída a declaração do processo, este pode ser salvo em arquivo.

Os blocos de dinâmica linear (contínua e discreta) e de atraso de transporte poderão associar variáveis auxiliares a seus parâmetros variantes no tempo. Cada um destes parâmetros deverá definir uma faixa de excursão, através de três valores: mínimo, médio (inicial) e máximo. Durante uma simulação as variáveis auxiliares assumirão valores entre -100, 0 e 100, correspondendo a valores para os parâmetros, de forma proporcional, entre o mínimo, o médio e o máximo, conforme gráfico apresentado na fig. 4.2. Este procedimento permite a processos completamente diferentes terem seus parâmetros alterados de uma forma genérica.

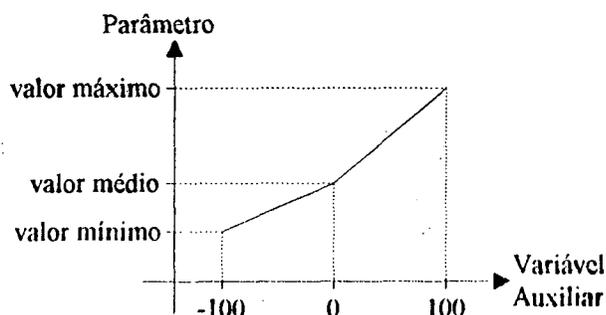


Fig. 4.2 - Gráfico de variação dos parâmetros

Conforme mencionado na seção anterior, o processo deve fornecer uma indicação do tempo de resposta a um degrau em malha aberta, o qual é obtido, de forma muito grosseira, pela seguinte fórmula:

$$t_{ma} = 5 \cdot \sum \tau + \sum L \quad (4.1)$$

onde L representa atrasos de transporte, e τ representa as constantes de tempo, considerando-se todos os blocos do processo.

Além do t_{ma} , o processo deve fornecer um valor que caracterize o tempo de resposta em malha fechada, necessário para a execução de seqüências de eventos. Este valor não necessita, de forma alguma, de precisão, servindo apenas para indicar uma ordem de grandeza na unidade de tempo de simulação. A fórmula utilizada para obter este valor é:

$$t_{mf} = \frac{(\sum \tau)^2}{\tau_{\max}} + 6 \cdot \sum L \quad (4.2)$$

onde τ_{\max} é a maior constante de tempo do processo. Esta fórmula, totalmente empírica, fornece um valor bastante aproximado do tempo de resposta do processo, a um degrau em malha fechada, sob atuação de um controlador PID com ajuste próximo ao de mínimo ITAE (integral do erro absoluto multiplicado pelo tempo).

A descrição dos sinais envolvidos na simulação é realizada através da classe "lista_sim" (lista de simulação), a qual contém uma lista de pares [tempo : valor]. Esta lista permite representar o comportamento de um sinal através de segmentos de reta, conforme apresentado no exemplo da figura 4.3 .

```

lista_sim: [ tempo : valor ]
              [ 0 : v1 ]
              [ t1 : v2 ]
              [ t1 : v3 ]
              [ t2 : v3 ]
              [ t3 : 0 ]
              [ t4 : v1 ]
  
```

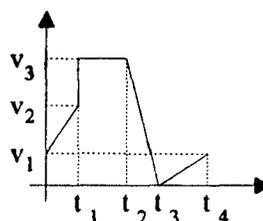


Fig. 4.3 - Exemplo de Representação de um Sinal

Esta forma de representação foi adotada em função dos blocos de dinâmica linear contínua, cujo sinal de saída é muito melhor aproximado por segmentos de reta do que por uma série de patamares de curta duração. Através de um experimento detectou-se que o passo de integração no método proposto pode ser 50 vezes maior que o do método tradicional, mantendo-se aproximadamente a mesma precisão no sinal de saída.

Durante a simulação do sistema, o módulo supervisor envia ao processo uma "lista_sim" contendo a descrição do sinal de saída do controlador (U), com duração de um tempo de amostragem. Ao recebê-la, o processo faz com que este sinal atue sobre o primeiro bloco, o qual retorna uma nova lista descrevendo o sinal em sua saída, no mesmo período de tempo. O sinal retornado pelo primeiro bloco é aplicado no seguinte, e assim sucessivamente até o último bloco, cuja saída corresponde à saída do processo (Y). Este último sinal é, então, retornado ao módulo supervisor que se encarrega de enviá-lo ao registrador gráfico e ao controlador.

Nas seções que se seguem são apresentadas algumas características de cada um dos tipos de blocos implementados para esta estrutura de processo.

4.4.1. BLOCO DLC_FT

A modelagem de uma dinâmica linear, contínua no tempo, pode ser realizada com este tipo de bloco, sendo a representação matemática efetuada por uma função de transferência. Um ponto de destaque para este tipo de bloco é a facilidade com que o usuário declara a função de transferência, sendo esta editada praticamente da mesma forma como é encontrada em textos, evidenciando-se que não há limitação quanto à ordem do sistema, ou seja, não se limita o grau do polinômio do denominador, desde que o sistema seja causal.

No corpo de uma função de transferência podem estar contidas até dez variáveis auxiliares, permitindo simular variações em quaisquer parâmetros: ganho, pólos e/ou zeros. Todos os detalhes quanto à declaração deste e dos demais blocos, além de exemplos ilustrativos, podem ser encontrados no manual do usuário.

A simulação deste bloco será realizada através do método de Runge-Kutta de quarta ordem, devido a um amplo domínio de suas potencialidades, no âmbito do LCMI, e qualidade na precisão dos cálculos, sendo o passo de integração calculado automaticamente, em função dos pólos da função de transferência. Para que este método possa ser aplicado, a dinâmica do bloco deve estar representada por equações de estado, sendo escolhida uma forma bastante parecida com a "forma canônica controlável", em função de certas propriedades.

Dada a função de transferência:

$$G(s) = \frac{b_{n-1}s^{n-1} + b_{n-2}s^{n-2} + \dots + b_1s + b_0}{a_n s^n + a_{n-1}s^{n-1} + \dots + a_1s + a_0} + d \quad (4.3)$$

onde $a_n \neq 0$; representa-se esta através das seguintes equações de estado:

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & 1 \\ \frac{a_0}{a_n} & \frac{a_1}{a_n} & \frac{a_2}{a_n} & \dots & \frac{a_{n-2}}{a_n} & \frac{a_{n-1}}{a_n} \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ \frac{1}{a_n} \end{bmatrix} u$$

$$\mathbf{y} = \begin{bmatrix} b_0 & b_1 & b_2 & \dots & b_{n-2} & b_{n-1} \end{bmatrix} \mathbf{x} + \begin{bmatrix} d \end{bmatrix} u$$

Esta representação permite que seja facilmente alterada a ordem do modelo durante a simulação, ou seja, o número de pólos pode ser modificado [Hsu 92], conforme demonstrado no anexo II desta dissertação. Além disto esta forma de representação não provoca perturbações quando da alteração de valores em constantes de tempo, apesar de alterações no ganho provocarem uma alteração imediata do sinal de saída, de forma proporcional.

4.4.2. BLOCO DLC_EE

Através de blocos do tipo DLC_EE é possível modelar dinâmicas lineares contínuas representadas por equações de estado. Por efeito de simplicidade, a quantidade de variáveis de estado está limitada a três, para cada bloco, permitindo a modelagem de processos de até terceira ordem. Esta limitação de ordem pode ser ampliada, em futuras versões do pacote, caso haja necessidade por parte dos usuários.

A dinâmica deste bloco é descrita pelos elementos das matrizes A, B, C e D que compõem as equações de estado, sendo estes elementos números reais e/ou variáveis auxiliares. Novamente utiliza-se o método de Runge-Kutta de quarta ordem, para a simulação, com o passo de integração calculado automaticamente.

4.4.3. BLOCO DLD

Este tipo de bloco é utilizado para modelar dinâmicas lineares, discretas no tempo, sendo sua representação matemática realizada através de uma função de transferência amostrada. A forma de declaração possui as mesmas propriedades apresentadas para os blocos DLC_FT.

A simulação deste tipo de bloco é realizada através do cálculo das equações de diferenças, obtidas através da conversão da função de transferência amostrada em uma forma semelhante à descrita no bloco DLC_FT.

4.4.4. BLOCO ATR

O bloco de atraso de transporte (ATR) modela um tempo de atraso inserido na estrutura do processo. O valor do tempo de atraso é definido através de uma variável auxiliar, a qual pode inclusive estar associada a parâmetros dos blocos de dinâmica linear.

O procedimento mais simples de realizar a simulação de um tempo de atraso é através de uma lista circular, mas desta forma o valor do tempo de atraso estaria limitado a múltiplos do tempo de amostragem do controlador. Adotou-se, então, uma solução um pouco mais complexa, mediante criação de uma lista de descrição do sinal interno do atraso. Todo sinal que entra neste bloco é adicionado ao final da lista, sendo extraído do início desta uma porção de sinal de tempo equivalente.

Eventuais alterações no valor do tempo de atraso, durante a simulação, farão comprimir ou estender, de forma proporcional, a escala de tempo da lista de descrição do sinal interno.

4.4.5. BLOCO GNL

Os blocos de ganho não linear (GNL) modelam curvas não lineares de ganho, geralmente encontradas em atuadores, sensores, etc. Esta modelagem é realizada através de uma curva segmentada definida por pontos de quebra. Os pontos de quebra são compostos por uma abcissa e uma ordenada.

Este tipo de bloco não utiliza variáveis auxiliares, o que impede que sejam realizadas alterações na curva durante a simulação.

4.4.6. BLOCO PRT

Um bloco do tipo PRT corresponde a um ponto de entrada de sinais de perturbação, definido na estrutura do processo. O sinal de perturbação será adicionado ao do sistema, podendo o primeiro ser uma combinação dos seguintes tipos de sinais: degrau, rampa, pulso, seno e ruído.

Todos estes tipos de sinais de perturbação são bastante simples e de implementação evidente, exceto o sinal de ruído que necessita uma melhor caracterização: geração por intermédio de números randômicos; distribuição normal de amplitude; média nula e desvio padrão definido pelo usuário.

4.5. MÓDULO SUPERVISOR

O módulo "supervisor" atua como uma porta de acesso ao controlador e ao processo, isolando-os dos demais blocos. Como consequência disto, uma grande parte dos métodos desta classe limitam-se à transferência de chamadas à métodos pertencentes às classes "processo" e "controlador".

A principal atividade do supervisor é realizar a transferência de sinais entre o controlador e o processo, promovendo a evolução da simulação. Durante esta simulação os sinais de saída do controlador (U) e do processo (Y), além do sinal da referência (Y_r), são enviados ao módulo "avaliador", para medição dos índices integrativos, e ao registrador gráfico, para permitir que o usuário acompanhe visualmente a evolução na dinâmica do sistema.

A evolução da simulação pode ser efetuada em duas velocidades distintas: em tempo real ou com a máxima capacidade de processamento da CPU. A simulação só ocorrerá verdadeiramente em tempo real se a CPU conseguir processar todas as informações necessárias, no período de tempo de um tempo de amostragem. O chaveamento entre estas velocidades é comandado pelo usuário.

Este módulo realiza, também, a ligação entre os módulos "executor" e "avaliador", durante a execução de bancos de provas.

4.6. MÓDULO EXECUTOR

A realização de simulações sistemáticas, voltadas à avaliação de desempenho de controladores adaptativos, é possível, neste ambiente, graças à execução de seqüências de eventos pré-programadas. Neste sentido, cabe ao módulo "executor" interpretar os arquivos contendo estas seqüências de eventos e executá-los.

Uma das principais características de uma seqüência de eventos é a genericidade, o que permite sua utilização sobre uma ampla variedade de processos. Para que isto seja possível, as medidas de tempo utilizadas para descrever a seqüência devem ser relativas ao tempo de resposta do processo sob simulação. Este fato conduziu à criação de uma unidade de tempo, denominada TMF (tempo de malha fechada), calculada em função das constantes de tempo e atrasos de transporte do processo, conforme equação 4.2 (seção 4.4). Salienta-se que esta unidade de tempo TMF possui uma grandeza variável durante a simulação, devido a mudanças em parâmetros do processo.

Outro fator preponderante para a genericidade das seqüências de eventos é a utilização das variáveis auxiliares para variação dos parâmetros do processo, o que padroniza as alterações à modificação de valores na faixa entre -100 e 100 (ver seção 4.4).

Os tipos de eventos disponíveis, até o momento, para a programação da simulação são apresentados a seguir:

- **Inicialização do controlador** : a inicialização do controlador adaptativo pode ser realizada utilizando-se valores default para as informações a serem prestadas pelo usuário (ver seção 4.3), ou mediante interação com o usuário, o que torna mais lento o processo de avaliação;
- **Configuração do controlador** : é sempre realizada mediante interação com o usuário;
- **Alteração em parâmetros do processo**;
- **Modificação no valor da referência**;
- **Modificação nos sinais de perturbação**;
- **Avanço da simulação, no tempo** : este tipo de evento especifica um intervalo de tempo, em TMF, durante o qual o controlador adaptativo atua sobre a dinâmica do processo;
- **Análise** : com este tipo de evento delimita-se o intervalo de tempo no qual ocorre uma análise dos sinais da simulação, ou seja, são medidos os índices integrativos;

- **Loop** : a utilização de loops permite que se repita a execução de um conjunto de eventos, por uma quantidade especificada de vezes. Um tipo especial de loop, denominado loop sinalizado, repete a execução do conjunto de eventos até que o controlador adaptativo esteja ajustado, ou até que a quantidade de repetições atinja o limite especificado. Este mecanismo é bastante útil na medição da velocidade de adaptação do controlador.

Detalhes quanto à programação das seqüências de eventos podem ser encontrados no manual do usuário, inclusive com um exemplo de utilização.

Programadas as seqüências de eventos, procede-se à construção de um banco de provas, com o qual é possível avaliar um controlador adaptativo frente a um amplo conjunto de processos e situações. Conforme apresentados na figura 4.3, um banco de provas é composto por um conjunto de seqüências de eventos, um conjunto de processos e ligações entre estes conjuntos, determinando os experimentos a serem realizados.

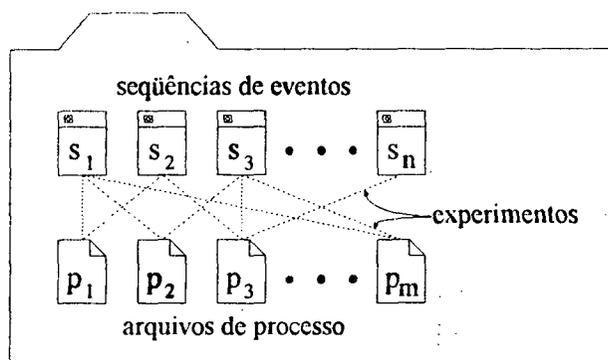


Fig. 4.3 - Banco de Provas

É de responsabilidade do módulo "executor" interpretar os arquivos de descrição do banco de provas, carregando processos e executando seqüências de eventos, conforme uma tabela de experimentos. Observa-se que não há a necessidade de se realizar todas as ligações seqüência-processo, o que permite evitar que seqüências voltadas à análise de perturbações, por exemplo, sejam executadas sobre processos sem entradas de perturbação.

Todos os comandos relativos à realização das simulações e avaliações de desempenho são enviados ao módulo "supervisor", o qual se encarrega de acessar os módulos "processo", "controlador" e "avaliador".

4.7. MÓDULO AVALIADOR

O módulo "avaliador" recebe, e armazena, as informações relativas à avaliação de desempenho do controlador adaptativo, durante a simulação automática de um banco de provas.

Para cada experimento este módulo confecciona um relatório contendo informações a respeito do processo, do controlador, e da seqüência de eventos, o que engloba as inicializações, as análises e os loops sinalizados. Ao final da execução do banco de provas, confecciona-se um relatório com informações estatísticas dos loops sinalizados contidos nos diversos experimentos.

Os controladores adaptativos podem fornecer mensagens relativas à inicialização, às análises, e aos loops sinalizados, as quais serão inseridas nos relatórios de experimentos, entre colchetes ("[" e "]"). Um exemplo destes relatórios de experimento é mostrado na figura 4.4, onde podem ser observadas as seguintes informações:

- **Processo** : nome do arquivo contendo sua declaração; tempo de resposta em malha aberta e malha fechada, conforme equações 4.1 e 4.2, respectivamente, com parâmetros variáveis em seus valores médios (iniciais);
- **Controlador** : nome do controlador adaptativo; limites de conversão e resolução dos conversores A/D e D/A; tempo de amostragem;
- **Seqüência de Eventos** : nome do arquivo que a contém; tempo despendido para cada uma das inicializações, seguido por mensagens provenientes do controlador; valores de ISE, ITAE, ISU, IU e tempo de análise, para cada uma das análises, seguido por mensagens provenientes do controlador; número de ciclos realizados para cada loop sinalizado, seguido por mensagens provenientes do controlador.

```

PROCESSO : CLAA
  Tempo de malha aberta (tma) = 107.2s
  Tempo de malha fechada (tmf) = 135.6s

CONTROLADOR : PID-AA
  Conversor A/D : ( -10 / 10 ) 12 bits
  Conversor D/A : ( -10 / 10 ) 12 bits
  Tempo de amostragem = 0.1s

SEQUENCIA : set1_-100
  Inicializacao 1 : 270s
    [K=4.004 L=4.002 T=21.2 pi=29.93% L/T=0.1888]
    [(1.177 13.46 3.027) (1.182 10.97 2.921)]
  Analise 1 : (ISE ITAE ISU IU tf)
    3.17251 37.2341 107.013 93.7773 129.7s
    [(1.177 13.46 3.027) (1.182 10.97 2.921)]
    [So=0.1696 Su=0.01576 2So=0 2Su=0 (G5) Ind:0.8692]
    [(1.177 18.19 3.027) (1.182 10.97 2.921)]
  Analise 2 : (ISE ITAE ISU IU tf)
    3.13535 30.5658 171.077 133.231 129.7s
    [(1.177 18.19 3.027) (1.182 10.97 2.921)]
    [So=0.1224 Su=0.0412 2So=0 2Su=0 (G3) Ind:0.923]
    [(0.8628 18.19 3.027) (1.182 10.97 2.921)]
  Analise 3 : (ISE ITAE ISU IU tf)

```

```

3.65042 44.0248 95.7563 94.3677 129.7s
{(0.8628 18.19 3.027) (1.182 10.97 2.921)}
{So=0.05166 Su=0.03188 2So=0 2Su=0 (G0) Ind:0.9553}
LOOP 1 : 2 ciclos
{( 5 3 0 )}

```

Fig. 4.4 - Relatório de um Experimento

É de incumbência do módulo "avaliador" promover o cálculo dos valores dos índices integrativos, a partir dos sinais da simulação fornecidos pelo módulo "supervisor". O tempo de inicialização do controlador adaptativo é medido pelo "supervisor", e o número de ciclos realizados em um loop sinalizado é fornecido pelo módulo "executor".

Uma análise mais geral é realizada no relatório do banco de provas, o qual contém dados estatísticos a respeito dos loops sinalizados contidos nos diversos experimentos. Estes dados compreendem o cálculo da média de ciclos realizados em: cada experimento, cada seqüência de eventos e todos experimentos, além do cálculo do desvio padrão relativo ao conjunto total de loops sinalizados. Pode-se observar na figura 4.5 um exemplo de relatório de banco de provas contendo quatro experimentos.

```

BANCO DE PROVAS : bc_tese

CONTROLADOR : PID-AA

Sequencia : seq1
  Processo : Proc3
    LOOP 1 : 1 Ciclos
    LOOP 2 : 3 Ciclos
    media = 2
  Processo : Proc6
    LOOP 1 : 6 Ciclos
    LOOP 2 : 15 Ciclos (nao ajustado)
    media = 10.5
    media = 6.25

Sequencia : seq5
  Processo : Proc3
    LOOP 1 : 7 Ciclos
    media = 7
  Processo : Proc6
    LOOP 1 : 4 Ciclos
    media = 4
    media = 5.5

Media Final (LOOPS ajustados) = 4.2 Ciclos (21/5)
Desvio Padrao = 2.135 Ciclos

LOOPS nao ajustados : 1

```

Media com LOOPs nao ajustados = 6 Ciclos (36/6)

Fig. 4.5 - Relatório de um Banco de Provas

Com as informações presentes nos relatórios é possível realizar avaliações de desempenho de quaisquer controladores adaptativos baseados em reconhecimento de formas, sendo que o estudo de outros tipos de controladores adaptativos pode conduzir à inclusão de novas informações [Kammer 92b].

4.8. INTERFACE HOMEM-MÁQUINA

A interface homem-máquina teve sua implementação guiada pelos princípios básicos de projeto de interface com usuários: simplicidade, consistência e eficiência [Sun 89]. A simplicidade se verifica através da manutenção do padrão visual e da clareza na identificação dos itens, fazendo com que o usuário consiga facilmente operar o ambiente, sentindo-se encorajado a descobrir novas potencialidades deste. A consistência é obtida pela manutenção dos padrões de utilização dos demais aplicativos que seguem o padrão OPEN LOOK, ou seja, mantém-se as funções básicas dos botões do mouse e das teclas, bem como o posicionamento e a nomenclatura dos controles usuais, tais como o botão "aplicar" e os painéis de armazenamento de dados. Busca-se eficiência no uso da interface com a técnica de esconder alguns comandos específicos, os quais serão descobertos apenas por usuários experientes, além de uma tentativa de minimizar o esforço na utilização do mouse e do teclado.

Toda a implementação da interface homem-máquina do SADECA é realizada com objetos do pacote de interface gráfica XView, descrito em [Heller 90], sendo estes objetos pertencentes ao padrão OPEN LOOK de interfaces gráficas. Desta forma o aspecto visual do ambiente de simulação está bastante agradável, e harmônico em relação aos demais aplicativos do ambiente multijanelas OpenWindows, conforme pode ser observado na figura 4.6 .

A apresentação da interface homem-máquina está muito ligada aos modos de operação do sistema, principalmente em função dos limites de interação impostos pela divisão dos modos de operação. Esta divisão não deve ser entendida como uma limitação do sistema, mas como uma forma de isolar etapas distintas, cujas ações não devem realmente ser sobrepostas, o que confere maior clareza na operação do ambiente.

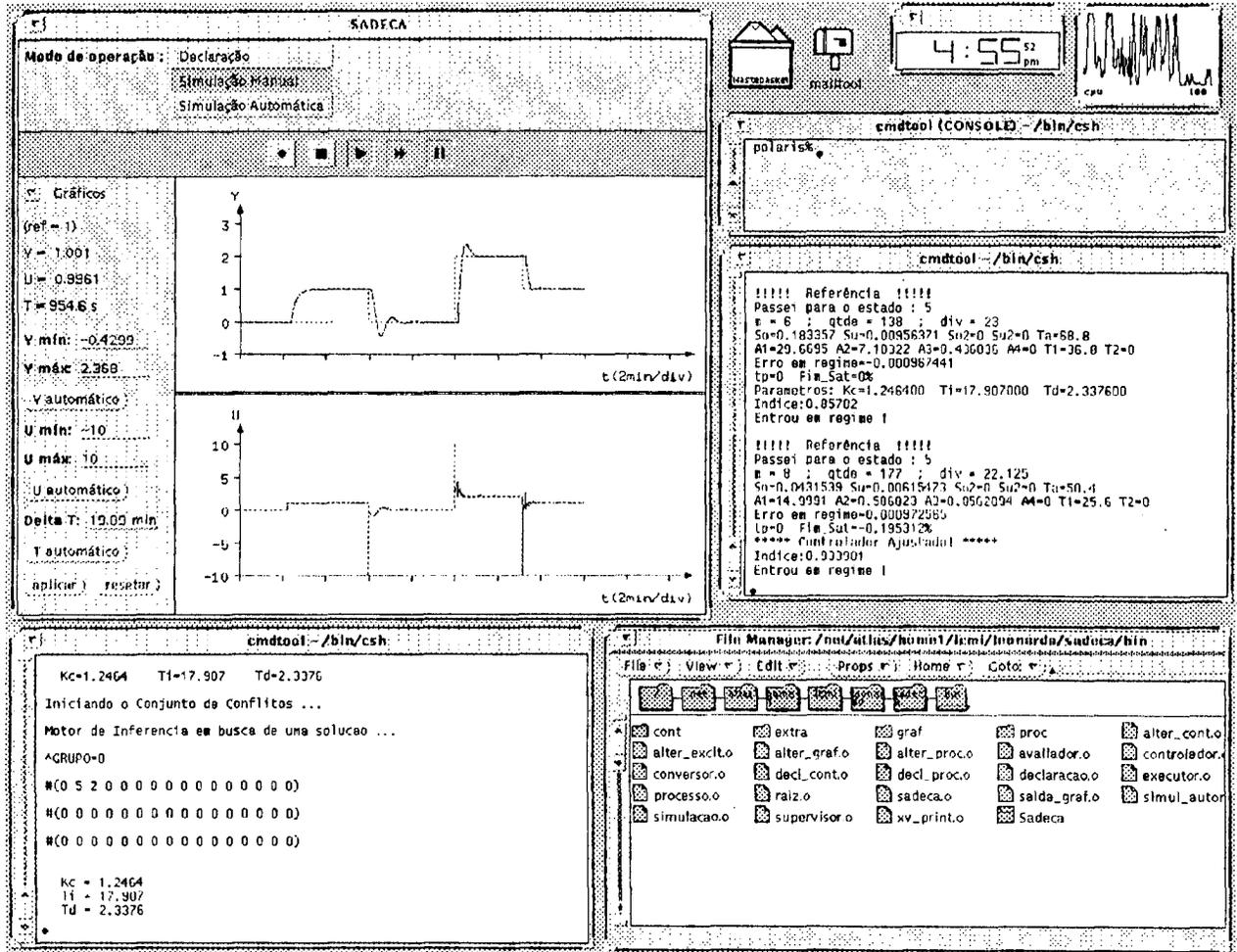


Fig. 4.6 - Ambiente OpenWindows

4.8.1. DECLARAÇÃO

Durante a declaração do sistema, figura 4.7, o pacote SADECA estará apresentando os painéis de declaração do controlador e do processo.

À esquerda da tela de declaração encontram-se os itens relativos à declaração do controlador, iniciando por um conjunto de botões com atuação mutuamente exclusiva, permitindo a seleção de um dentre os diversos tipos de controladores adaptativos disponíveis (implementados). Os demais campos deste painel são de entrada de dados numéricos, relativos aos conversores A/D e D/A, cujos valores podem ser alterados, se necessário. Qualquer alteração no valor de algum dos itens de um painel tornar-se-á vigente somente após ser pressionado o botão "aplicar", o que simplifica o gerenciamento das interações entre o ambiente e o usuário.

A declaração do processo é realizada em duas etapas: a) definição da seqüência de blocos que compõe a estrutura, e b) declaração dos blocos. Estas etapas são cumpridas com a utilização dos painéis existentes na área de declaração do processo, em cuja parte superior encontra-se o

painel relativo à etapa "a)", ao centro há uma tela (de desenho) onde se observa o estado atual da seqüência de blocos, e na parte inferior reserva-se uma área para apresentação dos painéis de declaração dos blocos.

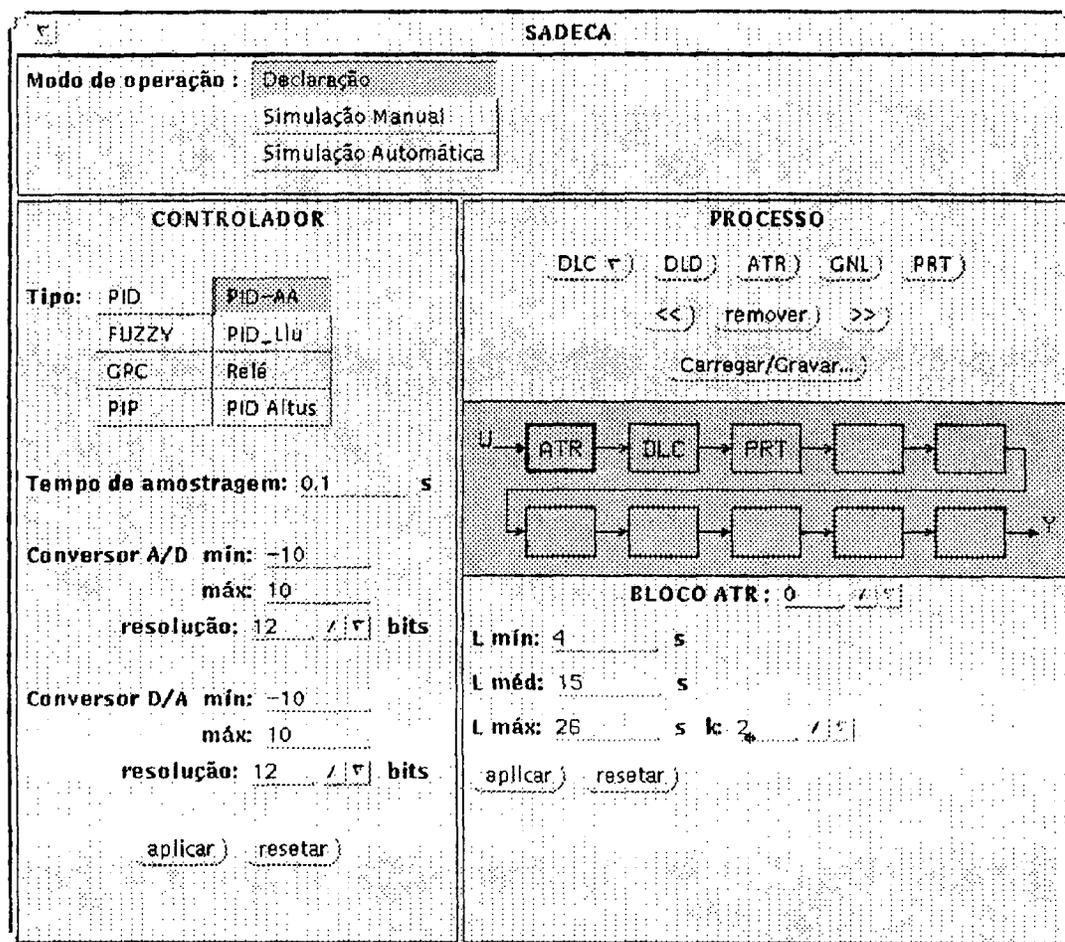


Fig. 4.7 - Tela de Declaração

4.8.2. SIMULAÇÃO MANUAL

Ao trocar o modo de operação do sistema, todos os painéis e telas de desenho são apagados, cedendo sua área aos elementos do novo modo. Desta forma a simulação manual faz uso de painéis e tela de desenho completamente distintos dos da declaração, conforme pode ser observado na figura 4.8 .

As simulações realizadas de forma manual estão, geralmente, voltadas ao desenvolvimento de um controlador adaptativo, em função da possibilidade que o usuário tem de, no transcorrer da simulação, promover alterações no sistema. Para que isto seja possível há a necessidade de acompanhamento contínuo dos sinais simulados.

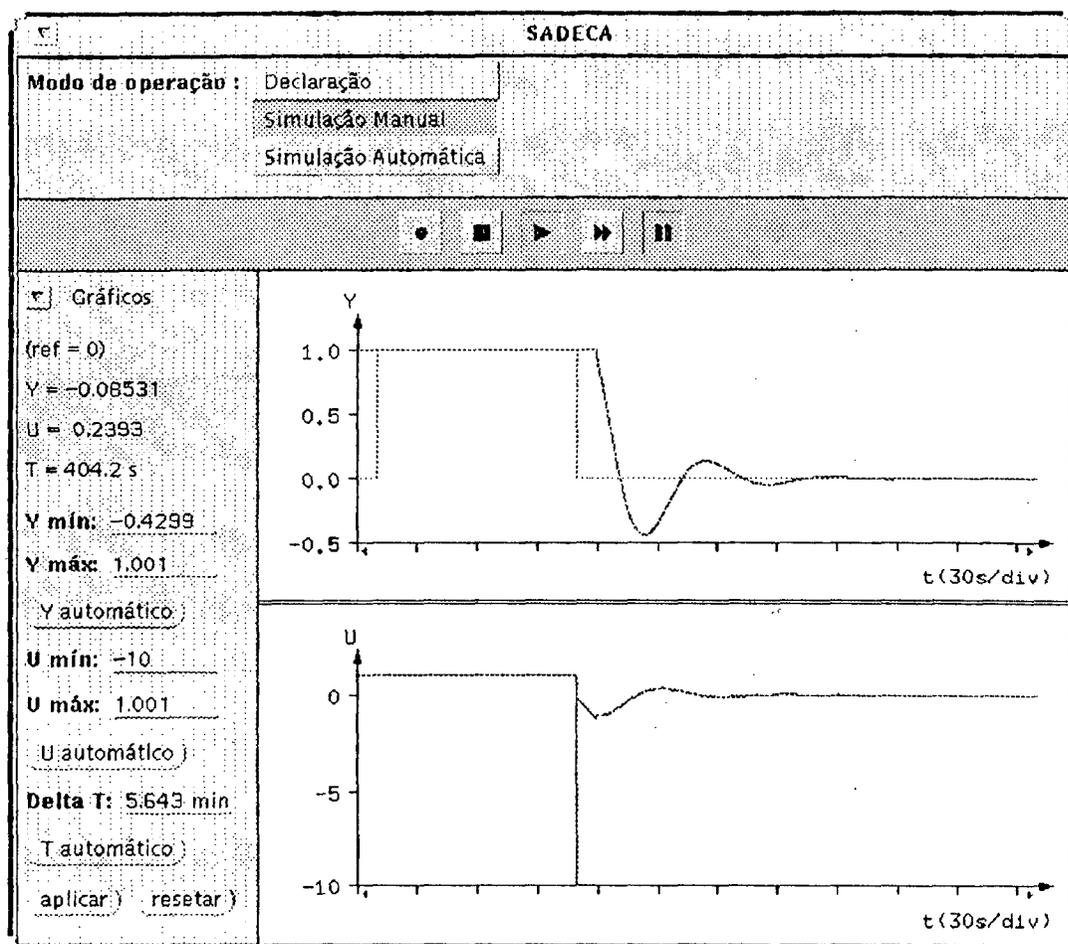


Fig. 4.8 - Tela de Simulação Manual

O acompanhamento da evolução da simulação é realizado através dos registradores gráficos Y e U, nos quais são plotados o sinal de saída do processo (antes do conversor A/D), o sinal de referência (no registrador Y), e o sinal de controle (após o conversor D/A). Estes sinais são plotados da esquerda para a direita dos registradores, e ao atingir o limite da escala de tempo, desloca-se o gráfico do sinal para a esquerda de forma a que se possa desenhar as informações mais recentes, desaparecendo da tela valores antigos.

Os registradores gráficos podem ter suas escalas, de amplitude e tempo, alteradas pelo usuário, conforme necessidade. Para facilitar esta operação foram criadas funções que ajustam, separadamente, as escalas em sua resolução máxima, de forma a conter todo o gráfico do sinal. Outras ações que podem ser realizadas sobre estes gráficos são a leitura dos pontos que os compõem, a ampliação de regiões e o armazenamento em arquivos.

Acima da tela de desenho que contém os registradores gráficos encontra-se o painel de comando da simulação, composto por botões contendo símbolos semelhantes ao de um toca-fitas comum, o que facilita a memorização das operações realizadas por estes.

Do lado esquerdo da tela, onde na figura 4.8 é visto o painel de alteração dos gráficos, na realidade situam-se mais três painéis, além deste, estando apenas um visível a cada momento. Estes painéis destinam-se à alterações nos seguintes elementos: gráficos (escalas), controlador (inicialização e configuração), processo (variáveis auxiliares) e excitações (referência e sinais de perturbação). Para obter uma descrição completa de cada item destes painéis, deve-se recorrer ao manual do usuário, apresentado no anexo I.

4.8.3. SIMULAÇÃO AUTOMÁTICA

Através da figura 4.9 pode ser observada a tela correspondente ao modo de operação: simulação automática, contendo na parte superior o painel de comando e abaixo desta a tela de desenho contendo registradores gráficos.

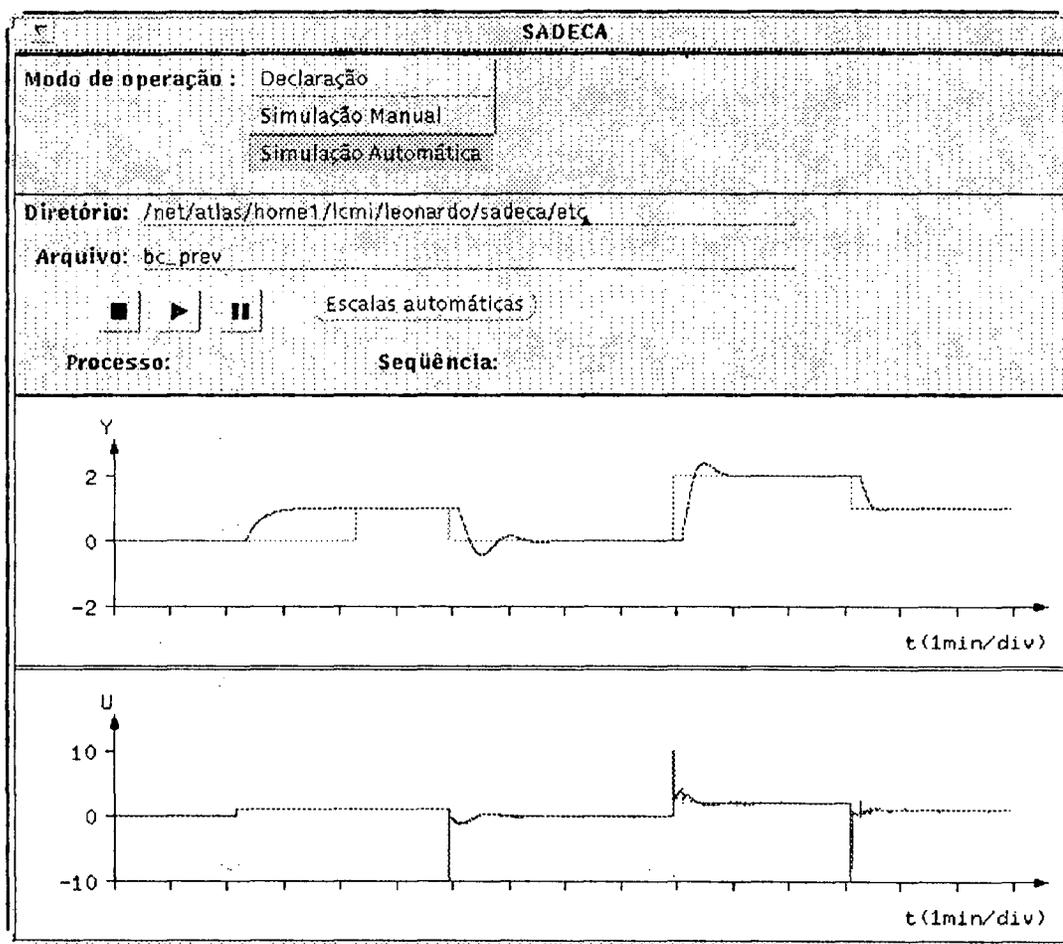


Fig. 4.9 - Tela de Simulação Automática

O painel de comando contém os campos relativos à: determinação do banco de provas a ser executado; controle de início, pausa e fim da simulação; e ajuste automático das escalas de amplitude e tempo dos registradores gráficos. Sobre estes registradores serão efetuadas somente

as ações de ajuste de escalas e gravação, em arquivo, dos gráficos de cada experimento. Estes arquivos podem ser posteriormente carregados, no modo de simulação manual, e os gráficos analisados com os recursos implementados.

4.9. CONCLUSÃO

Em sua forma atual, o pacote SADECA apresenta-se com tamanho aproximado de 540Kbytes, incluindo a implementação de alguns controladores adaptativos, o que pode não parecer muito, mas deve-se considerar que todo o suporte da biblioteca XView não é inserida no código final, apesar de ser utilizado. Para o ambiente SADECA foram criadas 46 classes, num total de quase 13000 linhas de código fonte.

Do total de classes criadas para este pacote, aproximadamente metade destina-se à interface homem-máquina. Este percentual é justificado pelas qualidades obtidas em termos de interação com o usuário, apresentando-se como um ponto de destaque deste ambiente. O primeiro contato com o SADECA já revela seu visual bastante agradável e harmonioso em relação ao ambiente multi-janelas no qual está inserido. Sua interface amigável estimula a utilização dos recursos disponíveis e descoberta de novas funcionalidades, especialmente pela clareza na apresentação dos painéis e comandos. A facilidade de interação com o ambiente não implica na redução de suas potencialidades, principalmente devido à utilização de mecanismos automáticos.

Finda a implementação deste ambiente de simulação, conclui-se seu desenvolvimento com uma etapa de testes, a qual na realidade acompanhou o avançar da implementação, sendo cada nova classe testada e corrigida. Mesmo com estes testes, faz-se necessária a manutenção do pacote, pois diversos defeitos passaram despercebidos, sendo posteriormente detectados por usuários e prontamente corrigidos.

Mesmo considerando que o pacote SADECA não contenha mais defeitos de funcionamento, resta ainda realizar o teste mais importante que é a verificação da utilidade deste ambiente tanto no desenvolvimento quanto na avaliação de controladores adaptativos. Um passo neste sentido será apresentado no capítulo seguinte, quando do aperfeiçoamento de um controlador adaptativo baseado em reconhecimento de formas.

CAPÍTULO 5 : UM EXEMPLO DE UTILIZAÇÃO DO AMBIENTE DE SIMULAÇÃO

5.1. INTRODUÇÃO

No presente capítulo são apresentadas a implementação, análises e avaliação de desempenho de um controlador adaptativo, no ambiente SADECA. Estes estudos visam verificar a utilidade deste ambiente de simulação, no desenvolvimento e na avaliação de desempenho de controladores adaptativos.

O controlador utilizado, denominado PID-AA (PID Auto-Ajustável), é baseado no reconhecimento de formas, sendo seu desenvolvimento descrito em [Pagano 89], [Caetano 92] e [Abreu 92]. Este controlador adaptativo é apresentado de uma forma sucinta na seção 5.2, sendo sua implementação no ambiente SADECA descrita na seção 5.3.

Uma análise da eficiência do pré-ajuste é realizada na seção 5.4, cabendo à seção 5.5 uma análise do desempenho das regras de ajuste dos parâmetros do controlador, em termos da velocidade de adaptação. Conclui-se este capítulo com a avaliação de desempenho de duas versões distintas deste controlador, obtidas em função das análises e dos aperfeiçoamentos realizados nas seções 5.4 e 5.5.

5.2. O CONTROLADOR PID-AA

O controlador PID-AA, como o próprio nome sugere, possui como algoritmo de controle uma estrutura do tipo PID, cujos parâmetros são ajustados automaticamente, de forma a prover uma adaptação à dinâmica do processo. O mecanismo de adaptação deste controlador baseia-se no reconhecimento das formas de onda das respostas do sistema.

A forma de onda desejada para a resposta é especificada pelos seguintes parâmetros: sobrepasso máximo (So_{max}) e subpasso máximo (Su_{max}), ambos em termos percentuais. Se algum destes parâmetros não for satisfeito, classifica-se a forma de onda obtida, em um dentre dez padrões pré-definidos, e realiza-se ajustes nos parâmetros do controlador. Para a quase totalidade dos estudos já realizados sobre o PID-AA, utilizou-se os seguintes parâmetros: $So_{max} = 10\%$ e $Su_{max} = 5\%$, sendo estes os valores especificados para as respostas, nas análises apresentadas neste capítulo. Há uma infinidade de curvas de resposta do processo que atendem a estas especificações, sendo que no desenvolvimento do PID-AA buscou-se fazer com que o ajuste tendesse a uma minimização na integral da área do erro (IAE).

O sinal do erro é continuamente monitorado na busca de transitórios, produzidos por mudanças de referência ou perturbações de carga. Quando o sinal do erro excede uma faixa de ruído previamente definida, inicia-se um ciclo de auto-ajuste, o qual é composto pelas seguintes etapas:

- detecção de um transitório;
- reconhecimento do tipo de excitação (referência ou perturbação de carga) e classificação da forma de onda num dos padrões pré-definidos;
- cálculo dos novos parâmetros do PID através de um sistema especialista baseado em regras de produção;
- transferência dos novos parâmetros para o algoritmo de controle.

A estrutura do controlador PID-AA, conforme implementação apresentada em [Abreu 92], é mostrada na figura 5.1 .

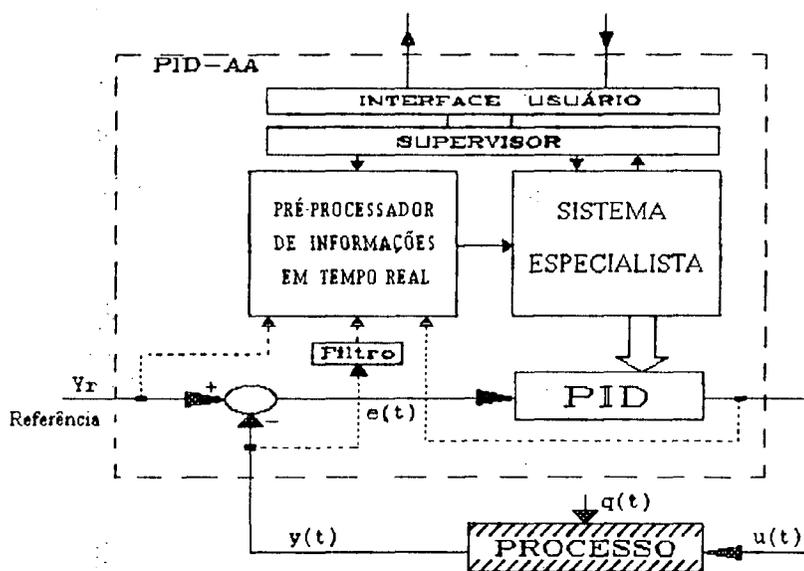


Fig. 5.1 - Estrutura do controlador PID-AA

Os valores iniciais para os parâmetros do controlador PID e para a faixa de ruído são determinados, automaticamente, através de um ensaio em malha aberta. Neste ensaio é aplicado um degrau na entrada do processo e observado o comportamento do sinal de saída, extraindo-se, desta curva de reação, informações a respeito do processo, com as quais calcula-se os parâmetros do controlador PID.

5.3. IMPLEMENTAÇÃO DO PID-AA NO AMBIENTE SADECA

Na sua implementação, no ambiente SADECA, o controlador PID-AA sofreu pequenas modificações, em comparação com a implementação apresentada na figura 5.1. Uma destas modificações é a exclusão do filtro existente na entrada do "pré-processador de informações em tempo real", e outra é que este pré-processador já não opera mais em tempo real. Estas alterações não invalidam, de forma alguma, os resultados obtidos nas análises.

Cabe ressaltar que já no decorrer dessa implementação foram sendo aperfeiçoados alguns mecanismos de obtenção de informações, tais como a medição da faixa de ruído e o reconhecimento da forma de onda em malha fechada.

O conjunto de regras do sistema especialista (SE) foi implementado em "SP0", um sistema de produção de ordem zero, baseado na lógica proposicional. Este sistema de produção, descrito em [De Souza 92], está implementado na linguagem Common Lisp e necessita trocar informações com o pré-processador de informações, implementado no ambiente SADECA, na linguagem C++. A forma encontrada para resolver este problema consiste na utilização de *sockets nomeados*, criados e utilizados por interfaces escritas na linguagem C, as quais facilmente comunicam-se com o Common Lisp e com o C++. A figura 5.2 apresenta este canal de comunicação por sockets, no qual são transferidas, unidirecionalmente, seqüências de caracteres.

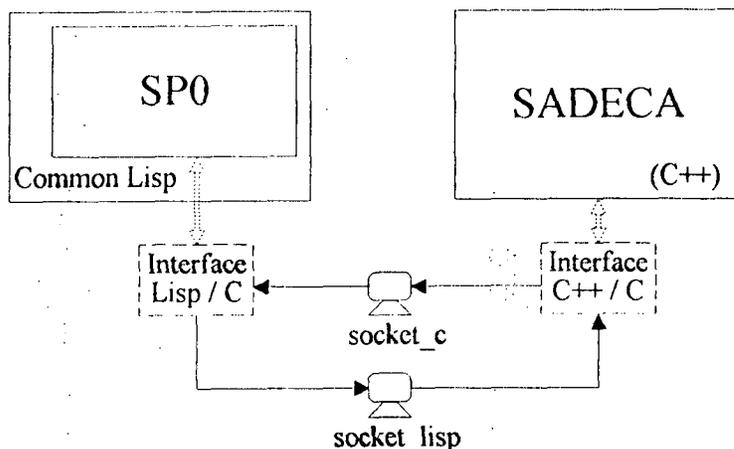


Fig. 5.2 - Mecanismo de Comunicação via Sockets Nomeados

Na configuração do controlador PID-AA são determinados os parâmetros de especificação da forma de onda desejada, um valor máximo para o erro em "regime", e um valor limite para o sobrepasso, em termos percentuais, o qual sendo ultrapassado provoca a atuação do "supervisor" no sentido de baixar o ganho do controlador. Além destes valores, pode-se ativar/desativar um filtro para a ação derivativa. Este filtro é um dos aperfeiçoamentos realizados neste ambiente, sendo seu funcionamento extremamente simples: desliga-se a ação derivativa

quando a resposta do processo atingir o regime permanente, tornando-se a ligá-la na ocorrência de um transitório. Todos estes itens compõem o painel de configuração do controlador PID-AA, apresentado na figura 5.3 .

PID-AA (Configuração)			
So max: 10	% Su max: 5	%	
So lim: 100	% Erro max: 1	%	
Filtro para Td <input type="checkbox"/>			
aplicar			

Fig. 5.3 - Painel de Configuração

A inicialização do controlador PID-AA pode ser realizada via pré-ajuste ou por declarações do usuário, conforme seleção do item "Entrada dos Parâmetros:" do painel de inicialização, visto na figura 5.4 .

PID-AA (Inicialização)			
Entrada dos Parâmetros:		Pré-ajuste	Usuário
Amplitude do Degrau: 1			
Tempo Máximo: 15.6			
PARÂMETROS DE REFERÊNCIA			
Kc: 1.345	TI: 12.65	TD: 2.824	
PARÂMETROS DE PERTURBAÇÃO			
Kc: 1.1E1	TI: 15.67	TD: 1.972	
Faixa de Ruído: 0.00576E			
Aplicar			

Fig. 5.4 - Painel de Inicialização

O mecanismo de pré-ajuste necessita das seguintes informações: amplitude do degrau de malha aberta e tempo máximo de observação da curva de reação. Em inicializações realizadas com valores "default", o degrau terá amplitude de uma unidade e o tempo máximo será de uma vez e meia o tempo de resposta em malha aberta fornecido pelo processo, conforme equação 4.1, ou seja, tempo máximo = $1,5 \cdot t_{ma}$. O pré-ajuste inclui uma análise do ruído, de forma a determinar uma faixa de amplitude que contenha seu sinal.

Uma inicialização realizada pelo usuário prevê a atribuição de valores aos parâmetros do controlador PID e à faixa de ruído. Na realidade o PID-AA contém dois conjuntos de parâmetros para o algoritmo de controle, sendo estes chaveados conforme a ocorrência de uma mudança de referência ou perturbação de carga [Abreu 92]. Como todas as análises realizadas neste capítulo estão voltadas ao caso de mudanças de referência, sempre que se falar em parâmetros do controlador deve-se entender os parâmetros relativos à este tipo de transitório.

Diversas mensagens são enviadas ao módulo "avaliador", passando a constar nos relatórios de experimento. Na fase de inicialização estas mensagens contêm as informações obtidas pela identificação do processo, com as quais calcula-se os parâmetros do controlador PID, bem como os próprios parâmetros calculados. Nas análises informa-se os parâmetros atuais do controlador e, se houver transitórios, valores importantes da forma de onda reconhecida, seguida pelos novos parâmetros inferidos. Ao concluir a realização de um loop sinalizado, o PID-AA informará a seqüência de padrões nos quais foram classificadas as formas de ondas.

Com estas informações torna-se possível analisar a eficiência dos mecanismos de inicialização e inferência, da última versão (V_1) do controlador PID-AA. Serão realizadas alterações criteriosas nestes mecanismos, de forma a que se possa verificar uma eventual possibilidade de aperfeiçoamento da versão sob análise.

5.4. ANÁLISE DO MECANISMO DE INICIALIZAÇÃO

A análise do mecanismo de inicialização visa determinar novas relações de cálculo entre as informações de caracterização do processo, obtidas via ensaio em malha aberta, e os parâmetros do controlador PID. Esta relação de cálculo é montada, de forma empírica, por tabelas, sendo que valores intermediários são obtidos via aproximação por segmentos de reta e formação de planos.

De forma a cobrir toda a extensão destas tabelas utiliza-se um conjunto bastante variado de processos, na formação do banco de provas. Estes processos, num total de dezoito, são apresentados no anexo III desta dissertação, sendo que todos possuem atrasos de transporte variáveis. Utilizando-se cada um destes processos com oito valores distintos para os atrasos de transporte, totaliza-se a quantidade de cento e quarenta e quatro funções de transferência diferentes. Destaca-se que o conjunto de processos escolhido engloba desde um processo de primeira ordem puro, até processos com pólos complexos conjugados e zeros de fase não-mínima, tornando-se significativo para o controlador sob teste.

A forma de realização da análise consiste na inicialização do controlador PID-AA, via mecanismo de pré-ajuste com valores "default", para cada função de transferência, seguido por uma mudança de referência, na qual observa-se o padrão da forma de onda. O banco de provas é

composto pelos dezoito processos mencionados anteriormente, e por oito seqüências de eventos contendo os seguintes eventos:

- atribuição de valor ao atraso de transporte;
- inicialização do controlador com valores "default";
- simulação por 1 (um) *tmf*;
- início de análise;
- mudança de referência;
- simulação por 3 (três) *tmf*;
- fim de análise.

Exemplos típicos de gráfico e relatório de um experimento são apresentados nas figuras 5.5 e 5.6, respectivamente.

Realizado o estudo da versão V_1 do PID-AA, constatou-se que a maior parte das inicializações conduziã a resposta do processo a formas de onda nos padrões G3 e G5, ambos caracterizados por sobrepasso acima do valor máximo e subpasso dentro das especificações, mas cuja diferença é que em G3 há um excesso de ação proporcional, enquanto que em G5 o excesso é de ação integrativa.

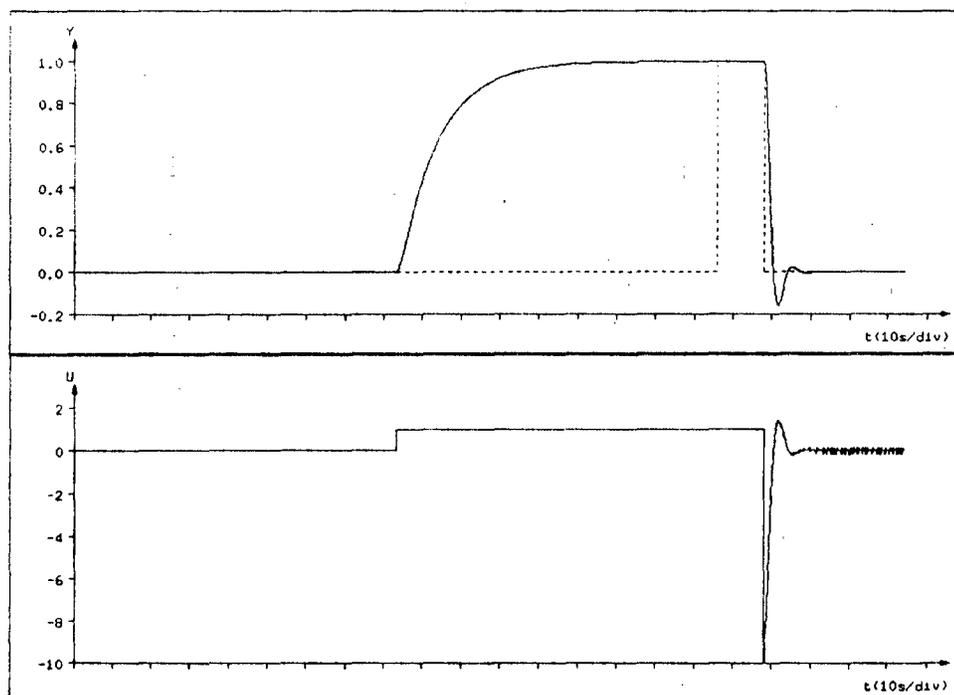


Fig. 5.5 - Exemplo de Gráfico de Experimento

PROCESSO : Proc2	
Tempo de malha aberta (tma) = 67.9s	
Tempo de malha fechada (tmf) = 89.5s	
CONTROLADOR : PID-AA	
Conversor A/D : (-10 / 10) 12 bits	
Conversor D/A : (-10 / 10) 12 bits	
Tempo de amostragem = 0.1s	
SEQUENCIA : seq1_-100	
Inicializacao 1 : 165.5s	
[K=1.001 L=0.473 T=13.02 pi=0.9756% L/T=0.03632]	
[(9.946 9.69 0.06777) (21.07 4.222 0.09679)]	
Analise 1 : (ISE ITAE ISU IU tf)	
0.981452 3.57775 90.9665 -9.94141 36.3s	
[(9.946 9.69 0.06777) (21.07 4.222 0.09679)]	
[So=0.1536 Su=0.0193 2So=0 2Su=0 (G5) Ind:0.8791]	
[(9.946 12.32 0.06777) (21.07 4.222 0.09679)]	

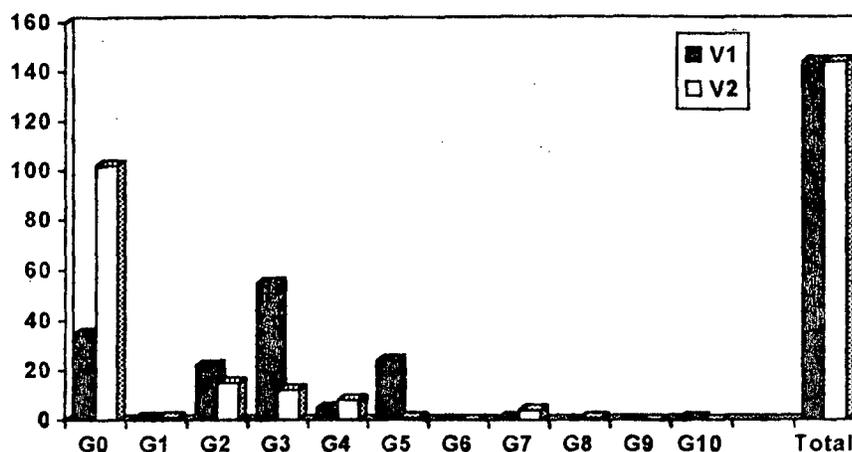
Fig. 5.6 - Exemplo de Relatório de Experimento

De forma a promover alterações nas referidas tabelas, buscou-se descobrir os parâmetros do PID que satisfaziam as especificações da curva de resposta, para cada uma das funções de transferência, obviamente de forma automática. Após um estudo criterioso destes resultados, modificou-se grande parte dos valores das tabelas que relacionam os dados do processo com os parâmetros do controlador PID, resultando em uma versão V₂ do PID-AA.

O mesmo estudo realizado para a versão V₁ foi feito para a versão V₂, sendo obtidos resultados bastante melhores, conforme pode ser observado na tabela 5.1, a qual fornece o número de casos de ocorrência, para cada padrão de forma de onda. A visualização destes dados torna-se mais fácil através do gráfico de barras apresentado no gráfico 5.1. É necessário salientar que o padrão G0 corresponde a uma resposta conforme as especificações, sendo que o número de casos aumentou consideravelmente na versão V₂. Outro ponto de destaque é a ocorrência de um padrão G10 na versão V₁, o que significa uma resposta com características de instabilidade, fortemente indesejável.

Padrão →	G0	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	TOTAL
V ₁	35	1	22	55	5	24	0	1	0	0	1	144
V ₂	102	1	15	12	8	1	0	4	1	0	0	144

Tab. 5.1 - Resultado da Análise de Inicialização



Gráf. 5.1 - Resultados dos Experimentos de Inicialização

As tabelas obtidas para a versão V_2 podem não conter os valores ótimos, mas com certeza contêm valores muito melhores do que as da versão V_1 , o que representa um aperfeiçoamento do controlador PID-AA.

5.5. ANÁLISE DAS REGRAS DE AJUSTE

Partindo-se da versão V_2 do controlador adaptativo sob teste, elaborou-se um conjunto de experimentos de forma a realizar uma análise da eficiência das regras de ajuste, dos parâmetros do PID, contidas no sistema especialista. O objetivo desta análise é tentar aumentar a velocidade de adaptação do controlador PID-AA, reduzindo o número de ciclos de ajuste.

Utiliza-se, novamente, as cento e quarenta e quatro funções de transferência citadas na análise anterior, para as quais o controlador adaptativo deverá ajustar os parâmetros do PID, de modo que a forma de onda, da resposta à mudanças de referência, atenda às especificações.

Os parâmetros do controlador PID deverão iniciar com valores não muito próximos dos ótimos, senão quase não haveriam ajustes a serem realizados, nem com valores absurdamente distantes, pois não se pretende incluir, nesta análise, o efeito dos padrões mais críticos como o G_{10} (instável) e o G_9 (quase instável).

A solução encontrada para o dilema acima consiste na execução do seguinte procedimento:

- a) calcula-se os parâmetros iniciais do PID, com as tabelas da versão V_2 ;
- b) multiplica-se cada um destes parâmetros por números pseudo-aleatórios, com valores na faixa de 0,4 a 1,6.

A utilização de números pseudo-aleatórios justifica-se pela necessidade de repetição do experimento, para diferentes versões do controlador PID-AA. Estes números são obtidos através de seqüências de números randômicos, para os quais é possível determinar um valor inicial, sendo os demais obtidos como uma função do anterior.

Definidos os parâmetros iniciais do controlador PID, procede-se à execução de um loop sinalizado, no qual são realizadas mudanças alternadas (incrementos e decrementos) no valor da referência. Com este procedimento torna-se possível medir a velocidade de ajuste (adaptação) dos parâmetros do controlador. Exemplos típicos de gráfico e relatório de um experimento são apresentados nas figuras 5.7 e 5.8, respectivamente.

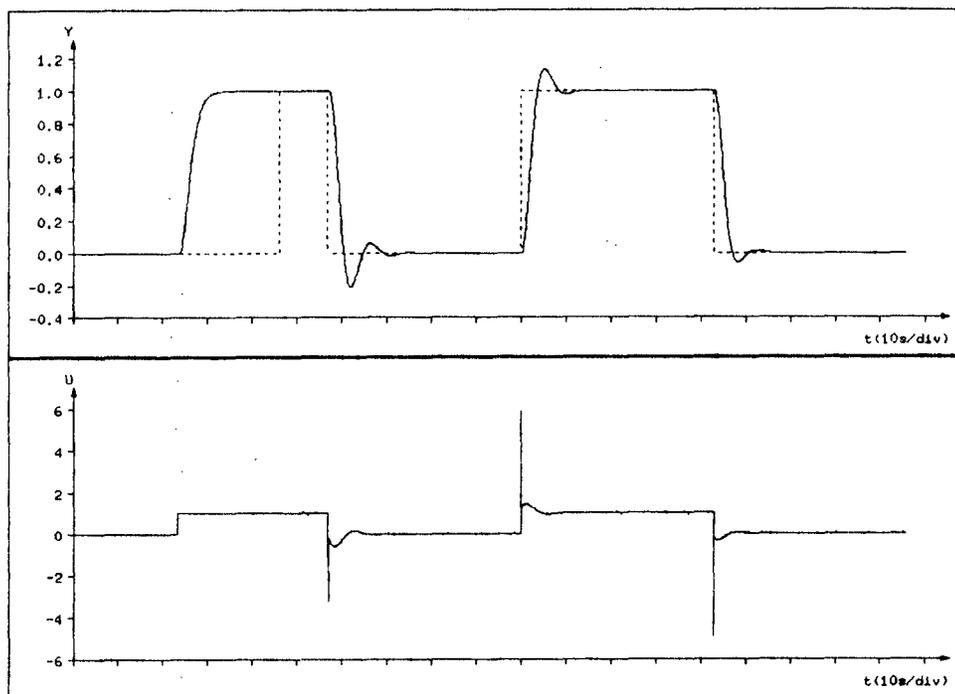


Fig. 5.7 - Exemplo de Gráfico de Experimento

```

PROCESSO : Proc7
  Tempo de malha aberta (tma) = 17.84s
  Tempo de malha fechada (tmf) = 26.0397s

CONTROLADOR : PID-AA
  Conversor A/D : ( -10 / 10 ) 12 bits
  Conversor D/A : ( -10 / 10 ) 12 bits
  Tempo de amostragem = 0.1s

SEQUENCIA : set2_-90
  Inicializacao 1 : 45.9s
  [K=1.001 L=1.112 T=3.64 pi=35.19% L/T=0.3055]
  [(1.143 2.044 0.2661) (2.156 1.932 0.627)]
  Analise 1 : (ISE ITAE ISU IU tf)
  1.80854 8.51815 1.81985 -1.5127 42.9s
  [(1.143 2.044 0.2661) (2.156 1.932 0.627)]
  
```

```

[So=0.2097 Su=0.05463 2So=0.01319 2Su=0 (G2) Ind:0.6923]
[(1.091 2.044 0.4411) (2.156 1.932 0.627)]
Analise 2 : (ISE ITAE ISU IU tf)
-1.70321 6.10202 48.6782 44.3872 42.9s
[(1.091 2.044 0.4411) (2.156 1.932 0.627)]
[So=0.1334 Su=0.01812 2So=0 2Su=0 (G5) Ind:0.8853]
[(1.091 2.379 0.4411) (2.156 1.932 0.627)]
Analise 3 : (ISE ITAE ISU IU tf)
1.69724 4.88612 2.64307 -1.12012 42.9s
[(1.091 2.379 0.4411) (2.156 1.932 0.627)]
[So=0.05819 Su=0.01201 2So=0 2Su=0 (G0) Ind:0.9479]
LOOP 1 : 2 ciclos
[( 2 5 0 )]

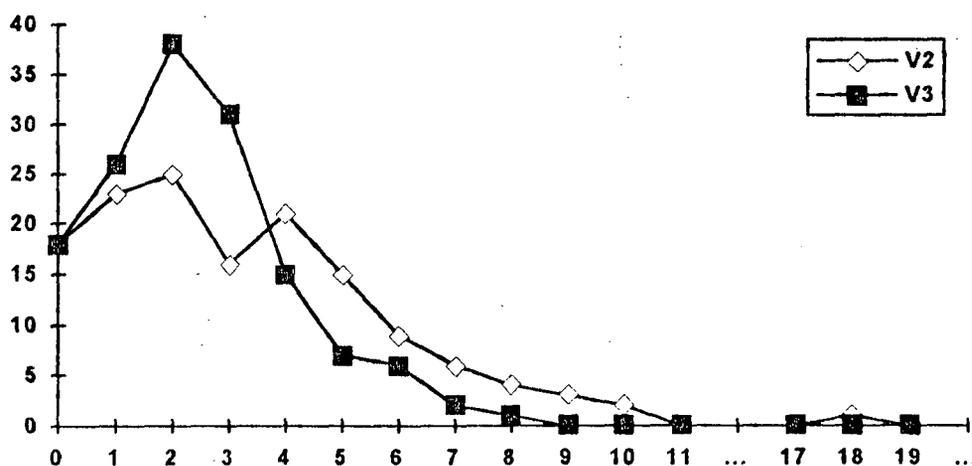
```

Fig. 5.8 - Exemplo de Relatório de Experimento

O estudo realizado sobre a versão V_2 do PID-AA permitiu que fossem alteradas diversas regras do sistema especialista, dando origem à versão V_3 . Os resultados desta análise podem ser vistos na tabela 5.2 e no gráfico 5.3. Na tabela estão registradas as quantidades de ocorrências em função do número de ciclos necessários ao ajuste. Na penúltima coluna desta tabela (∞) encontra-se registrada a quantidade de casos para os quais não foi possível ajustar o controlador, após o número máximo de ciclos definido para o loop sinalizado (30 ciclos para esta análise).

n° de ciclos →	0	1	2	3	4	5	6	7	8	9	10	11	...	17	18	19	...	∞	TOTAL
V_2	18	23	25	16	21	15	9	6	4	3	2	0	...	0	1	0	...	1	144
V_3	18	26	38	31	15	7	6	2	1	0	0	0	...	0	0	0	...	0	144

Tab. 5.2 - Resultado da Análise das Regras de Ajuste



Gráf. 5.2 - Resultados dos Experimentos das Regras de Ajuste

Comparando-se as duas versões, pode-se destacar os seguintes fatos:

- Na versão V₂ houve um caso em que o controlador não conseguiu encontrar parâmetros para o PID de forma a atender às especificações do sistema, o que não ocorreu na versão V₃;
- Desconsiderando-se os casos em que o controlador PID-AA inicia ajustado, e para a versão V₂ o caso em que não conseguiu ajustá-lo, obtém-se os seguintes resultados estatísticos:

	\bar{x}	σ
V ₂	3,784	2,576
V ₃	2,762	1,514

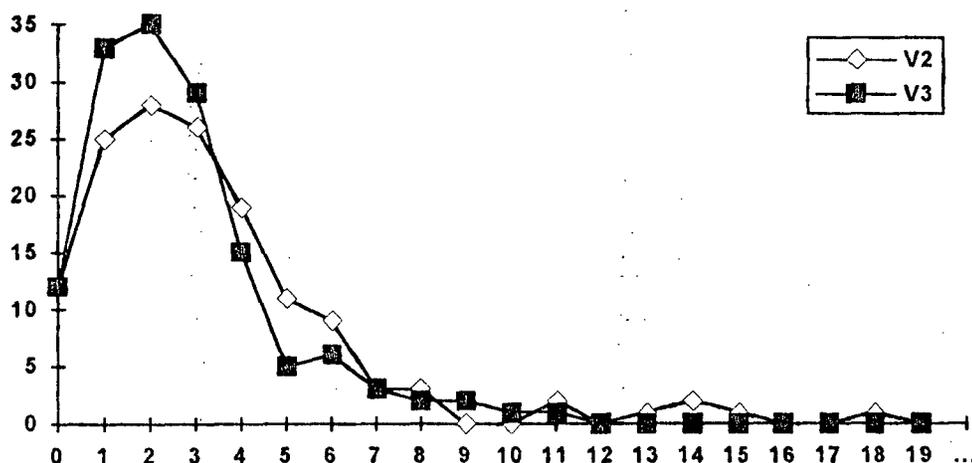
onde \bar{x} corresponde ao valor esperado e σ ao desvio padrão, ambos em número de ciclos de ajuste. Com estes resultados estatísticos conclui-se, com bastante clareza, a superioridade da versão V₃, em termos da velocidade de ajuste dos parâmetros.

De forma a reforçar a conclusão a respeito da melhoria obtida com a versão V₃, repetiu-se esta mesma análise para uma nova seqüência de números aleatórios, o que conduziu a novos parâmetros iniciais para o controlador PID. Estes resultados são mostrados na tabela 5.3, e no gráfico 5.3, além dos dados estatísticos apresentados em seguida.

nº de ciclos →	0	1	2	3	4	5	6	7	8	9	10	11	
V ₂	12	25	28	26	19	11	9	3	3	0	0	2	
V ₃	12	33	35	29	15	5	6	3	2	2	1	1	

nº de ciclos →		12	13	14	15	16	17	18	19	∞	TOTAL
V ₂		0	1	2	1	0	0	1	0	1	144
V ₃		0	0	0	0	0	0	0	0	0	144

Tab. 5.3 - Resultado da Análise das Regras de Ajuste (caso 2)



Gráf. 5.3 - Resultados dos Experimentos das Regras de Ajuste (caso 2)

Resultados estatísticos:

	\bar{x}	σ
V ₂	3,702	2,988
V ₃	2,932	2,020

Estes novos resultados acrescentam argumentos à conclusão de que houve uma real melhoria nas regras de ajuste dos parâmetros do PID, em relação ao número de ajustes necessários até que se atinja as especificações. Certamente as regras vigentes na versão V₃ não são as melhores possíveis, podendo e devendo ser melhor estudadas, mas com o que foi feito até o momento já se pode concluir a respeito da utilidade deste ambiente de simulação.

5.6. UMA AVALIAÇÃO DE DESEMPENHO

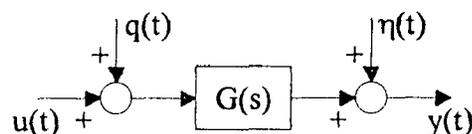
Finalizando esta dissertação, propõe-se um pseudo "benchmark" para avaliação de desempenho de controladores adaptativos, especialmente elaborado de acordo com as diretrizes traçadas no capítulo dois. A estruturação desta avaliação está direcionada ao caso do PID-AA, mas nada impede outros controladores de serem avaliados. Todas as formulações realizadas para esta avaliação possuem um caráter meramente de exemplificação, devendo estas serem modificadas com a realização de estudos detalhados a respeito da importância de cada fator, no desempenho do sistema.

Nas seções que se seguem realiza-se a apresentação do "benchmark" e das avaliações realizadas sobre as versões V₁ e V₃ do controlador PID-AA.

5.6.1. APRESENTAÇÃO DO "BENCHMARK"

O "benchmark" proposto compõe-se de: um processo, definições para o controlador adaptativo, um experimento e a formulação do cálculo da avaliação de desempenho do controlador.

A estrutura do processo possui a seguinte forma:



onde $u(t)$ e $y(t)$ são as entrada e saída do processo, respectivamente, $q(t)$ são sinais de perturbação na forma de degrau, e $\eta(t)$ representa o sinal de ruído. Estes dois últimos sinais

constam na estrutura do processo apenas como complemento, visando uma futura avaliação mais detalhada, pois não serão utilizados no experimento apresentado neste capítulo.

A função de transferência $G(s)$ é dada por:

$$G(s) = \frac{K(1 + \tau_1 s)e^{-Ls}}{(1 + 10s)(1 + \tau_2 s)(1 + 2\tau_1 s)} \quad (5.1)$$

onde os valores dos parâmetros são apresentados na tabela abaixo:

valor	K	τ_1	τ_2	L
mínimo	0,5	0,0	5,0	6,0
nominal	1,0	0,0	10,0	10,0
máximo	2,0	3,0	18,0	16,0

Para o controlador adaptativo são definidos:

- Objetivo de controle : sobressaço máximo de 10% e subpasso máximo de 5%;
- Tempo de amostragem : 100 ms;
- Conversor A/D : +10 / -10, 12 bits;
- Conversor D/A : +10 / -10, 12 bits.

A avaliação de desempenho é realizada com os seguintes parâmetros:

- a) velocidade de adaptação;
- b) qualidade da resposta;
- c) dificuldade de obtenção das informações a priori;
- d) tempo despendido no ajuste inicial.

Para cada um destes parâmetros é calculado um valor entre zero (0,0) e um (1,0), relativo ao desempenho do controlador sob avaliação, sendo o valor final (R_{av}) obtido pela seguinte ponderação:

$$R_{av} = 4 \cdot a + 4 \cdot b + c + d \quad (5.2)$$

De forma a analisar cada um destes parâmetros propõe-se a realização do seguinte experimento:

- atribuir os valores nominais aos parâmetros do processo;
- realizar um ajuste inicial de forma que a resposta atenda às especificações;

- mensurar a qualidade da resposta;
- atribuir o valor 1,25 ao parâmetro K , do processo;
- atribuir o valor máximo ao parâmetro τ_2 , do processo;
- realizar um loop sinalizado com mudanças de referência;
- atribuir o valor 1,5 ao parâmetro τ_1 , do processo;
- atribuir o valor máximo ao parâmetro L , do processo;
- realizar um loop sinalizado com mudanças de referência;
- atribuir os valores nominais aos parâmetros do processo;
- realizar um loop sinalizado com mudanças de referência;
- mensurar a qualidade da resposta;
- atribuir o valor mínimo ao parâmetro K , do processo;
- atribuir o valor 12,0 ao parâmetro τ_2 , do processo;
- atribuir o valor mínimo ao parâmetro L , do processo;
- realizar um loop sinalizado com mudanças de referência;
- atribuir o valor nominal ao parâmetro K , do processo;
- atribuir o valor nominal ao parâmetro τ_2 , do processo;
- realizar um loop sinalizado com mudanças de referência;
- atribuir os valores nominais aos parâmetros do processo;
- realizar um loop sinalizado com mudanças de referência;
- mensurar a qualidade da resposta;

O valor do parâmetro "velocidade de adaptação" (α) é calculado em função do número médio de ciclos (\bar{x}) realizados nos seis loops sinalizados que compõem o experimento. Este valor é dado por:

$$a = 1 - 0,2 \cdot (\bar{x} - 1) \quad (5.3)$$

O valor relativo à "qualidade da resposta" (b) é calculado em função do índice ITAE, medido em uma mudança de referência de zero (0,0) para um (1,0). Durante a realização do experimento este índice é medido três vezes, sempre com os mesmos parâmetros para o processo:

$$G(s) = \frac{1 \cdot e^{-10s}}{(1 + 10s)^2} \quad (5.4)$$

Através de um estudo a parte, desta função de transferência, com as seguintes condições de contorno:

- Controlador PID discreto com tempo de amostragem de 100 ms
- Conversores A/D e D/A com limites +10 / -10, e resolução de 12 bits
- Mudanças de referência de 0,0 para 1,0

encontrou-se como mínimo* valor para ITAE: $ITAE_{\min} = 309,479$, o qual será utilizado no cálculo de b através da seguinte relação:

$$b = \frac{ITAE_{\min}}{ITAE} \quad (5.5)$$

onde \overline{ITAE} corresponde à média das três medições realizadas no experimento. Salienta-se que as respostas que não satisfizerem às especificações receberão valor zero neste item.

À "dificuldade de obtenção das informações a priori" (c) é atribuído o valor calculado pela seguinte expressão:

$$c = 1 - n_f \cdot 0,02 - n_m \cdot 0,08 - n_d \cdot 0,2 \quad (5.6)$$

onde n_f , n_m e n_d são as quantidades de informações consideradas fáceis, médias e difíceis, respectivamente. Face à relatividade de classificação de cada informação, propõe-se a utilização da tabela 5.4, a qual contém algumas características relativas a cada um dos tipos de informação.

* Talvez seja um mínimo local, mas isto não impede que o valor seja utilizado como referência de um bom ajuste.

Tipo de informação	Características
FÁCIL	<ul style="list-style-type: none"> • pode ser fornecida pelo operador de planta, ou obtida pelo simples conhecimento da grandeza controlada; • é limitada em apenas um extremo, como por exemplo maior que certo valor.
MÉDIA	<ul style="list-style-type: none"> • necessita um simples experimento prévio, sobre o processo, a fim de se obter o valor desejado; • é limitada por uma faixa na qual o valor está contido.
DIFÍCIL	<ul style="list-style-type: none"> • é fornecida apenas no manual do processo, ou necessita de um experimento complexo para obtenção; • valor necessita ser obtido com relativa precisão.

Tab. 5.4 - Características das Informações a Priori

O tempo despendido no ajuste inicial engloba a inicialização do controlador adaptativo (pré-ajuste), e os ciclos de mudanças de referência até que os objetivos de controle sejam atingidos. O valor do parâmetro relativo a este tempo despendido (d) é calculado por:

$$d = 1 - \frac{t_{\text{inic}} - 40}{520} - 0,1 \cdot n_{\text{cic}} \quad (5.7)$$

onde t_{inic} é o tempo despendido no pré-ajuste, em segundos, e n_{cic} é o número de ciclos necessários ao ajuste do controlador.

Em algumas das equações apresentadas, os parâmetros a , b , c e d poderiam assumir valores fora da faixa especificada: $[0,0 ; 1,0]$. Quando este fato ocorrer deve-se atribuir, ao parâmetro, o valor limite correspondente, ou seja, atribui-se zero quando o valor calculado for negativo e um (1,0) quando este valor for superior à unidade.

5.6.2 AVALIAÇÃO DE DESEMPENHO DA VERSÃO V₁ DO PID-AA

O experimento apresentado no "benchmark" foi programado em uma sequência de eventos (SeqBench), e em conjunto com o processo apresentado (Bench) compuseram um banco de provas. Este banco de provas, executado frente à versão V₁ do PID-AA, gerou o relatório de experimento apresentado na figura 5.9, com o qual podem ser obtidas quase todas as informações necessárias à avaliação de desempenho proposta. Os dados, deste relatório, a serem utilizados na avaliação estão em destaque para facilitar a leitura.

```

PROCESSO : Bench
  Tempo de malha aberta (tma) = 110s
  Tempo de malha fechada (tmf) = 100s

CONTROLADOR : PID-AA
  Conversor A/D : (-10 / 10 ) 12 bits
  Conversor D/A : (-10 / 10 ) 12 bits
  Tempo de amostragem = 0.1s

SEQUENCIA : SeqBench
  Inicializacao I : 330s
  [K=1.001 L=13.07 T=26.43 pi=29.45% L/T=0.4946]
  [(1.049 16.41 5.128) (1.315 17.57 5.638)]
  Analise 1 : (ISE ITAE ISU IU tf)
  16.7016 356.436 467.782 410.328 400s
  [(1.049 20.47 5.128) (1.315 17.57 5.638)]
  [So=0.04029 Su=0 2So=0 2Su=0 (G0) Ind:0.9282]
  Analise 2 : (ISE ITAE ISU IU tf)
  20.0711 626.621 439.649 407.454 400s
  [(0.7316 16.42 3.825) (1.315 17.57 5.638)]
  [So=0.0848 Su=0 2So=0 2Su=0 (G0) Ind:0.9594]
  Analise 3 : (ISE ITAE ISU IU tf)
  18.3005 450.478 446.688 408.125 400s
  [(0.9207 20.08 3.836) (1.315 17.57 5.638)]
  [So=0.05115 Su=0.0116 2So=0 2Su=0 (G0) Ind:0.9428]
  LOOP 1 : 1 ciclos
  [( 5 0 )]
  LOOP 2 : 8 ciclos
  [( 2 2 2 2 4 2 4 2 0 )]
  LOOP 3 : 3 ciclos
  [( 2 3 7 0 )]
  LOOP 4 : 4 ciclos
  [( 7 7 3 3 0 )]
  LOOP 5 : 2 ciclos
  [( 7 7 0 )]
  LOOP 6 : 3 ciclos
  [( 3 3 5 0 )]
  LOOP 7 : 3 ciclos
  [( 2 3 7 0 )]

```

Fig. 5.9 - Relatório da Avaliação da Versão V₁

O parâmetro a é calculado em função do número médio de ciclos realizados nos loops sinalizados relativos a adaptações, ou seja, LOOP 2 a LOOP 7:

$$\bar{x} = \frac{8+3+4+2+3+3}{6} = 3,8333$$

$$a = 1 - 0,2 \cdot (3,8333 - 1) = 0,4333$$

O parâmetro b considera os valores medidos de ITAE, conforme:

$$\overline{\text{ITAE}} = \frac{356,436 + 626,621 + 450,478}{3} = 477,845$$

$$b = \frac{309,479}{477,845} = 0,6477$$

O mecanismo de inicialização do controlador PID-AA, conforme apresentado na seção 5.3, necessita de duas informações para a realização do pré-ajuste: amplitude do degrau de malha aberta e tempo máximo de observação. Ambas informações são consideradas de dificuldade média, de acordo com os dois critérios de classificação apresentados na tabela 5.4. Isto conduz ao seguinte valor para o parâmetro c :

$$c = 1 - 0 \cdot 0,02 - 2 \cdot 0,08 - 0 \cdot 0,2 = 0,8400$$

O parâmetro d é calculado em função do tempo indicado na "Inicializacao 1" e do número de ciclos realizados no LOOP 1:

$$d = 1 - \frac{330 - 40}{520} - 0,1 \cdot 1 = 0,3423$$

Com estes parâmetros torna-se possível obter o resultado da avaliação da versão V_1 :

$$R_{av}^{V_1} = 4 \cdot 0,4333 + 4 \cdot 0,6477 + 0,8400 + 0,3423 = 5,5063$$

5.6.3 AVALIAÇÃO DE DESEMPENHO DA VERSÃO V_3 DO PID-AA

Utilizando-se o mesmo banco de provas, citado na seção anterior, foi obtido o relatório de experimento relativo à versão V_3 do PID-AA. Com os dados contidos neste relatório, apresentado na figura 5.10, é possível avaliar o desempenho desta versão e compará-lo com a versão anterior.

<p>PROCESSO : Bench Tempo de malha aberta (tma) = 110s Tempo de malha fechada (tmf) = 100s</p>
<p>CONTROLADOR : PID-AA Conversor A/D : (-10 / 10) 12 bits Conversor D/A : (-10 / 10) 12 bits Tempo de amostragem = 0.1s</p>
<p>SEQUENCIA : ScqBench Inicializacao 1 : 330s [K=1.001 L=13.07 T=26.43 pi=29.45% L/T=0.4946] [(1.049 19.7 5.128) (1.315 17.57 5.638)]</p>

Analise 1 : (ISE ITAE ISU IU tf)
16.7009 382.365 469.512 411.059 400s
[(1.049 19.7 5.128) (1.315 17.57 5.638)]
[So=0.05798 Su=0 2So=0 2Su=0 (G0) Ind:0.9406]
Analise 2 : (ISE ITAE ISU IU tf)
16.9931 441.391 460.281 410.014 400s
[(1.126 22.42 3.901) (1.315 17.57 5.638)]
[So=0.08554 Su=0.04223 2So=0 2Su=0 (G0) Ind:0.9852]
Analise 3 : (ISE ITAE ISU IU tf)
16.4851 395.387 467.138 410.031 400s
[(1.153 22.84 4.537) (1.315 17.57 5.638)]
[So=0.05768 Su=0.0364 2So=0.0009623 2Su=0 (G0) Ind:0.9622]
LOOP 1 : 0 ciclos
[(0)]
LOOP 2 : 1 ciclos
[(2 0)]
LOOP 3 : 3 ciclos
[(2 2 3 0)]
LOOP 4 : 5 ciclos
[(7 7 2 2 5 0)]
LOOP 5 : 2 ciclos
[(7 7 0)]
LOOP 6 : 4 ciclos
[(2 2 2 2 0)]
LOOP 7 : 6 ciclos
[(2 2 2 4 5 5 0)]

Fig. 5.10 - Relatório da Avaliação da Versão V₃

Os valores dos parâmetros de avaliação são obtidos conforme apresentado abaixo:

$$\bar{x} = \frac{1+3+5+2+4+6}{6} = 3,5$$

$$a = 1 - 0,2 \cdot (3,5 - 1) = 0,5$$

$$\overline{\text{ITAE}} = \frac{382,365 + 441,391 + 395,387}{3} = 406,381$$

$$b = \frac{309,479}{406,381} = 0,7615$$

$$c = 1 - 0 \cdot 0,02 - 2 \cdot 0,08 - 0 \cdot 0,2 = 0,8400$$

$$d = 1 - \frac{330 - 40}{520} - 0,1 \cdot 0 = 0,4423$$

Com estes parâmetros torna-se possível obter o resultado da avaliação da versão V₃:

$$R_{av}^{V_3} = 4 \cdot 0,5 + 4 \cdot 0,7615 + 0,8400 + 0,4423 = 6,3283$$

Este valor de 6,3283 é um pouco superior ao obtido com a versão V_1 (5,5063), mas pode ser melhorado bastante ainda, face a novos estudos e aperfeiçoamentos do controlador PID-AA. A reduzida quantidade de dados obtidos, através do experimento proposto, deve ser levada em consideração nas conclusões à respeito do desempenho dos controladores, uma vez que este caso representa apenas um dentre muitos para os quais o controlador adaptativo de propósito geral deve estar preparado.

5.7. CONCLUSÃO

O sucesso obtido nas análises realizadas neste capítulo permitiram não somente aperfeiçoar o controlador adaptativo estudado, como também verificar a importância do ambiente SADECA como uma ferramenta extremamente eficiente no desenvolvimento de controladores adaptativos, especialmente os baseados em reconhecimento de formas. Em termos comparativos, pode-se afirmar que cada análise realizada automaticamente, durante um período aproximado de quatro horas, levaria pelo menos uma semana de trabalho interativo com o usuário, caso fossem realizadas com os recursos disponíveis antes da implementação deste ambiente.

Com relação à avaliação de desempenho foi possível comprovar a utilidade do ambiente de simulação, em função dos recursos existentes e das informações fornecidas nos relatórios de experimentos, com os quais torna-se possível realizar avaliações sistemáticas e quantitativas.

A automação dos ensaios mediante a geração de seqüências de eventos sobre diferentes processos (bancos de provas) e a elaboração de relatórios que apresentem índices de desempenho para comparar as diferentes estruturas estudadas, permite minimizar o tempo de teste e alcançar uma alta confiabilidade. Isto é extremamente importante para o desenvolvimento de controladores "inteligentes" baseados em conhecimento, como o PID-AA, onde a integração de novo conhecimento implica na realização de novos testes.

CAPÍTULO 6 : CONCLUSÕES E PERSPECTIVAS

Frente aos problemas que dificultam a aceitação das técnicas de controle adaptativo no meio industrial, especialmente a falta de confiabilidade nos estudos de avaliação de desempenho, o ambiente de simulação SADECA surge como um instrumento bastante poderoso no auxílio à busca de soluções.

Comprovou-se, no decorrer desta dissertação, que este ambiente possibilita a realização de avaliações sistematizadas, quantitativas e sobre amplos conjuntos de processos, preceitos básicos à confiabilidade de estudos avaliativos.

Uma característica fundamental deste ambiente é a flexibilidade oferecida no suporte ao desenvolvimento de diferentes abordagens ao controle adaptativo. Esta flexibilidade é observada tanto nos mecanismos de simulação e avaliação, quanto na implementação (programação) de novos controladores. Isto fez com que se tornasse realidade a implementação, e desenvolvimento, tanto de controladores clássicos, como o PID discreto, quanto de controladores baseados em novos paradigmas, como o FUZZY (baseado em lógica difusa), além, obviamente, de controladores adaptativos, como o PID-AA, o PIP (Proporcional Integral Preditivo) [Bruciapaglia 92] e o GPC.

Apesar da programação dos novos controladores fornecer a flexibilidade mencionada, esta tarefa pode se tornar bastante árdua ao usuário, especialmente na criação e tratamento de dados dos painéis de inicialização e configuração. Uma das perspectivas futuras para este ambiente é a criação de funções básicas de manipulação dos objetos do pacote XView necessários a estes painéis, o que tornaria muito menos complexa a atividade de implementação de controladores adaptativos.

Ferramentas de análise, tais como o lugar das raízes, podem ser de grande interesse ao usuário que esteja desenvolvendo um novo controlador adaptativo. A implementação de algumas destas ferramentas estão incluídas nas perspectivas do ambiente SADECA.

Uma forma de evitar que os usuários tenham que programar seus controladores na linguagem C++, ou outra qualquer, para avaliá-los, é através da conexão de conversores A/D e D/A reais, cabendo ao ambiente promover a simulação da dinâmica do processo, em tempo real, e a execução das seqüências de eventos. Com estas modificações será possível avaliar controladores adaptativos de propósito geral disponíveis comercialmente, mas cuja implementação não é fornecida.

Outra perspectiva para o simulador é a implementação de uma estrutura para o processo que inclua realimentações internas através de blocos lineares e não-lineares, algo bastante poderoso em termos de representação de processos, mas de complexidade computacional elevada.

Além destas perspectivas maiores, pode-se dizer que algumas melhorias podem ser feitas nos registradores gráficos, em termos de permitir a comparação de regiões de gráficos de sinais, além de uma ampliação da estrutura de avaliação, provendo novos recursos nas seqüências de eventos de forma a aperfeiçoar os estudos estatísticos.

Em relação às avaliações de desempenho, estão previstos estudos de padronização dos testes de avaliação e a definição de novos critérios para análise de: robustez frente a não-linearidades, capacidade de solucionar situações de instabilidade do sistema, etc.

Finalmente, com a experiência adquirida no uso deste ambiente de simulação espera-se o desenvolvimento de melhores sistemas de controle adaptativo para uso industrial, bem como sua utilização no treinamento de engenheiros e operadores de planta no uso das técnicas de controle adaptativo.

ANEXO I : MANUAL DO USUÁRIO DO AMBIENTE SADECA

SISTEMA PARA AVALIAÇÃO
DE DESEMPENHO DE
CONTROLADORES ADAPTATIVOS
(SADECA)

Vol. I : Manual do Usuário

Leonardo César Kammer

Laboratório de Controle e Microinformática (LCMI)
Departamento de Engenharia Elétrica
Universidade Federal de Santa Catarina

SADECA (manual do usuário)

Sadeca (Sistema para Avaliação de DEsempenho de Controladores Adaptativos) é um pacote computacional que, através de simulações, permite realizar avaliações do comportamento de diferentes estruturas de controle adaptativo.

Os controladores adaptativos podem ser testados sobre uma ampla gama de processos monovariáveis cuja dinâmica varia no tempo, inclusive processos com alguns tipos de não linearidades. As simulações podem ser realizadas de forma automática através da execução de seqüências pré-definidas de eventos (mudanças nos parâmetros do processo, mudanças da referência, introdução de perturbações e ruído, etc.). Ao final do teste são gerados relatórios de avaliação contendo índices de desempenho e análises estatísticas.

Este pacote foi desenvolvido utilizando-se uma linguagem orientada a objetos (C++), sobre o ambiente UNIX*, em estações de trabalho. As interfaces de entrada e saída do programa foram implementadas com o pacote gráfico XView** (padrão OPEN-LOOK* de interfaces gráficas). Por este motivo a execução do Sadeca deve ser realizada dentro do ambiente OpenWindows**. Muitas das ações realizadas sobre o programa são definidas pelo próprio padrão OPEN-LOOK, cabendo ao usuário consultar manuais referentes a este tópico caso não esteja familiarizado com o mesmo.

O programa divide-se em três modos de operação: "declaração", "simulação manual" e "simulação automática", sendo que no primeiro são declarados o controlador e o processo, e nos demais opera-se a simulação do sistema.

Este manual do usuário é dividido em cinco capítulos. No primeiro há uma descrição de informações preliminares para utilização do pacote; nos três capítulos seguintes aborda-se cada um dos modos de operação, e no capítulo final descreve-se o procedimento para programação de novos controladores adaptativos.

* UNIX e OPEN LOOK são marcas registradas de AT&T.

** Xview e OpenWindows são marcas registradas de SUN Microsystems, Inc.

SUMÁRIO

I.1 : INFORMAÇÕES PRELIMINARES	72
I.2 : DECLARAÇÃO	73
I.2.1. DECLARAÇÃO DO CONTROLADOR	73
I.2.2. DECLARAÇÃO DO PROCESSO	74
I.2.2.1. DECLARAÇÃO DO BLOCO DLC	74
I.2.2.1.1. FUNÇÃO DE TRANSFERÊNCIA	74
I.2.2.1.2. EQUAÇÕES DE ESTADO	76
I.2.2.2. DECLARAÇÃO DO BLOCO DLD	77
I.2.2.3. DECLARAÇÃO DO BLOCO ATR	77
I.2.2.4. DECLARAÇÃO DO BLOCO GNL	78
I.2.2.5. DECLARAÇÃO DO BLOCO PRT	79
I.3 : SIMULAÇÃO MANUAL	80
I.3.1. PAINEL DE COMANDO	80
I.3.2. GRÁFICOS	81
I.3.3. PAINÉIS DE ALTERAÇÃO	82
I.3.3.1. ALTERAÇÃO NOS GRÁFICOS	82
I.3.3.2. ALTERAÇÃO NO CONTROLADOR	82
I.3.3.3. ALTERAÇÃO NO PROCESSO	83
I.3.3.4. ALTERAÇÃO NAS EXCITAÇÕES	83
I.4 : SIMULAÇÃO AUTOMÁTICA	85
I.4.1. SEQÜÊNCIA DE EVENTOS	85
I.4.2. BANCO DE PROVAS	88
I.4.3. PAINEL DE COMANDO	89
I.4.4. RELATÓRIOS DAS SIMULAÇÕES	89
I.5 : PROGRAMAÇÃO DE CONTROLADORES	92
I.5.1. IMPLEMENTAÇÃO MÍNIMA	92
I.5.2. IMPLEMENTAÇÃO DOS RECURSOS BÁSICOS	93
I.5.2.1. INICIALIZAÇÃO E CONFIGURAÇÃO	94
I.5.2.2. CONVERSORES A/D E D/A, E OUTROS RECURSOS	96
I.5.3. IMPLEMENTAÇÃO DOS RECURSOS DE AVALIAÇÃO	99
I.5.3.1. MECANISMOS GERAIS	99
I.5.3.2. MECANISMOS PARA ENVIO DE MENSAGENS	100

I.1 : INFORMAÇÕES PRELIMINARES

Instalação

A instalação do pacote SADECA pode ser realizada em qualquer diretório, suponhamos, por exemplo, em "/x/y/z". O usuário deverá acrescentar a seu arquivo ".cshrc" as seguintes linhas: "setenv SADECAHOME /x/y/z" e "set path = (\$path \$SADECAHOME)". Com isto é possível rodar o programa digitando "Sadeca" após o prompt, a partir de qualquer diretório, desde que se esteja no ambiente OpenWindows. Observe que o nome do programa possui o primeiro caracter maiúsculo ("S") e os demais minúsculos ("adeca").

Arquivos de Dados

Em diversos pontos do programa há a possibilidade de carregar/gravar dados de/em arquivos. Estas operações são controladas por painéis semelhantes ao apresentado na figura 1.1, os quais denominaremos painéis de arquivamento. Através dos campos "Diretório:" e "Arquivo:" são inseridos, respectivamente, o caminho e o nome do arquivo sendo tratado. As operações de carregamento e gravação são ativados através dos botões "Carregar" e "Gravar", respectivamente.

OBJETO : Arquivos	
Diretório:	/net/atlas/home1/lcml/leonardo/sadeca
Arquivo:	
Carregar	Gravar

Fig. 1.1: Painel de Arquivamento

Botões "aplicar" e "resetar"

Em muitos painéis há os botões "aplicar" e "resetar", os quais exercem sempre as mesmas funções. O botão "aplicar" faz vigorar os valores presentes nos itens do painel, portanto a simples alteração de um campo de dados não implica na imediata alteração do objeto relativo ao painel. É importante que o usuário não se esqueça de pressionar o botão "aplicar" após alterados os dados, pois em caso contrário estas alterações serão perdidas assim que houver uma mudança de tela.

O botão "resetar" faz mostrar, nos campos do painel, os valores vigentes do objeto em questão. Este botão é utilizado quando se quer desistir de alterações que haviam sido feitas no painel, para então modificar outro(s) campo(s).

I.2 : DECLARAÇÃO

Neste capítulo aborda-se as declarações do controlador adaptativo e do processo. As operações descritas estarão disponíveis ao usuário quando for selecionado o "Modo de operação : Declaração". A figura 2.1 apresenta o pacote Sadeca neste modo de operação, onde podem ser vistos os campos para declaração do controlador (esquerda) e do processo (direita).

The screenshot shows the SADECA software interface in the 'Declaração' mode. The window is titled 'SADECA' and is divided into two main sections: 'CONTROLADOR' (Controller) on the left and 'PROCESSO' (Process) on the right.

CONTROLADOR (Controller):

- Modo de operação:** Declaração (selected), Simulação Manual, Simulação Automática.
- Tipo:** A table with two columns:

PID	PID-AA
FUZZY	PID_Llu
GPC	Relé
PIP	PID Altus
- Tempo de amostragem:** 0.1 s
- Conversor A/D:** min: -10, max: 10, resolução: 12 bits
- Conversor D/A:** min: -10, max: 10, resolução: 12 bits
- Buttons: aplicar, resetar

PROCESSO (Process):

- Buttons: DLC, DLD, ATR, GNL, PRT
- Buttons: <<, remover, >>
- Button: Carregar/Gravar...
- Block diagram showing a flow from U through ATR, DLC, PRT, and several empty blocks to Y.
- BLOCO ATR:** 0 bits
- L mín:** 4 s
- L méd:** 15 s
- L máx:** 26 s k: 2
- Buttons: aplicar, resetar

Fig. 2.1: Tela de Declaração

I.2.1. DECLARAÇÃO DO CONTROLADOR

Inicia-se a definição do controlador através da escolha do tipo de estrutura a ser utilizada, podendo ser uma das opções existentes no primeiro item do painel: "Tipo:".

Pode ser que nesta lista de controladores haja estruturas não adaptativas. Isto se deve ao fato do sistema, através dos mecanismos existentes, poder realizar comparações e/ou avaliações de controladores clássicos, além dos adaptativos.

Os campos seguintes dizem respeito aos conversores A/D e D/A instalados respectivamente nas entrada e saída do controlador. Define-se, então, o tempo de amostragem destes conversores, a faixa de conversão (valores mínimos e máximos) e a resolução. Uma resolução de zero bits corresponde a uma faixa contínua de valores para o conversor em questão.

I.2.2. DECLARAÇÃO DO PROCESSO

O modelo do processo é dado por uma estrutura de até dez blocos em cascata, sem realimentações internas. Estes blocos podem ser de cinco diferentes tipos: dinâmica linear contínua (DLC), dinâmica linear discreta (DLD), atraso de transporte (ATR), ganho não linear (GNL) e entrada de perturbação (PRT). A função de cada um destes blocos será explicada nas seções seguintes, junto à descrição de seus painéis de declaração.

A tela de declaração do processo está dividida em três partes: painel superior, quadro da estrutura e painel inferior.

Através do painel superior cria-se a estrutura do processo, ou seja, determina-se os tipos dos blocos que constituirão o processo. Esta estrutura é visualizada na segunda parte da tela de declaração, ou seja, no quadro da estrutura. Este quadro possui uma cor de fundo um pouco mais escura que os painéis superior e inferior, e nele podem ser vistos dez blocos (retângulos), em seqüência, que podem estar vazios ou não. Um bloco vazio não contém letras no interior de seu retângulo, enquanto que um bloco não vazio possui três letras que identificam seu tipo.

A cada momento um dos blocos do processo pode ser declarado, ou seja, pode-se alterar o conteúdo do bloco selecionado. Esta declaração é realizada através do painel inferior, que estará apresentando os campos relativos ao bloco sob alteração (ver seções seguintes). O usuário pode identificar este bloco selecionado, através do quadro da estrutura, pois o mesmo apresentar-se-á por um retângulo em linhas mais grossas que os demais. A seleção do bloco é realizada através dos botões "<<" e ">>" existentes no painel superior.

A inserção de novos blocos se dá na posição corrente (posição em que se encontra o bloco selecionado), sendo os blocos seguintes deslocados para a direita. Para realizar esta inserção são utilizados os botões "DLC", "DLD", "ATR", "GNL" e "PRT". É possível remover o bloco selecionado, através do botão "remover", fazendo com que os blocos posteriores recuem para a esquerda.

A seqüência de blocos e seus conteúdos podem ser armazenados em arquivo, agilizando desta forma a declaração do processo. Estas operações com arquivos são realizadas através de um painel de arquivamento ativado pelo botão "Carregar/Gravar...".

I.2.2.1. DECLARAÇÃO DO BLOCO DLC

Um bloco de dinâmica linear contínua (DLC) pode ser descrito tanto por uma função de transferência (DLC_FT) quanto por um conjunto de equações de estado (DLC_EE). Estas opções são selecionadas a partir do botão-menu "DLC", sendo a função de transferência a opção default.

I.2.2.1.1. FUNÇÃO DE TRANSFERÊNCIA

O painel para declaração do bloco de dinâmica linear contínua, através de sua função de transferência, é apresentado na figura 2.2.

A cada bloco DLC_FT associa-se um número entre 0 e 9, o qual será utilizado como referência para alterações em parâmetros do processo. Este número do bloco é declarado no primeiro item do painel ("Bloco DLC_FT :").

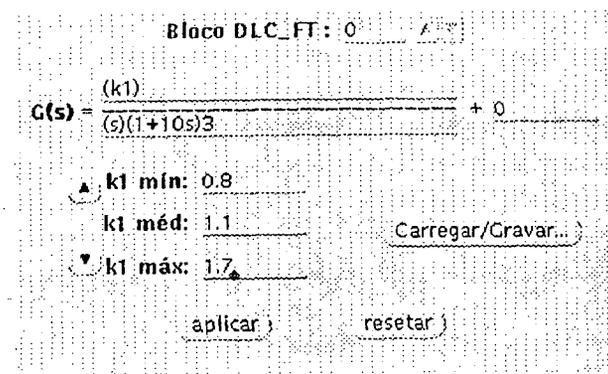


Fig. 2.2: Bloco DLC_FT

A função de transferência $G(s)$, que descreve a dinâmica do bloco DLC, é composta por três campos: numerador ($N[s]$), denominador ($D[s]$) e ganho de transferência direta (d). Estes campos podem ser visualizados no painel:

$$G(s) = \frac{N[s]}{D[s]} + d$$

, onde " $N[s]$ " e " $D[s]$ " são seqüências de caracteres que descrevem polinômios e " d " é um número real. O grau de $N[s]$ deve ser menor ou igual ao de $D[s]$.

As regras sintáticas para " $N[s]$ " e " $D[s]$ ", conforme padrão EBNT, são dadas abaixo:

```

N[s]|D[s] = polinômio [{"+"|-"} polinômio].
polinômio = {" " "(" {" " " [{"+"|-"} termo} ")" [alg_int [algarismo]] {" " "}.
termo = (real [termo_k [termo_s]] | (termo_k [termo_s]) | termo_s.
real = {" " " ("0" "." {algarismo} | "." {algarismo} {algarismo}) [expoente] {" " "}.
expoente = "e" [{"+"|-"} alg_int [algarismo].
termo_k = {" " " "k" algarismo {" " "}.
termo_s = {" " " "s" [alg_int [algarismo]] {" " "}.
algarismo = ("0"|alg_int).
alg_int = ("1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9").

```

Destas regras observa-se: a) $N[s]$ e $D[s]$ são polinômios formados por operações de adição, subtração e multiplicação de polinômios (a multiplicação é realizada quando não há os símbolos "+" ou "-" entre dois polinômios); b) todo polinômio inicia com o caracter "(" e termina com o ")" seguido opcionalmente por um expoente inteiro positivo de até dois algarismos; c) os termos do polinômio são compostos por uma parte real seguida por uma das variáveis "k" e finalmente o termo "s" com sua potência (todas as partes são opcionais).

Ex. 1: A função de transferência $G(s) = \frac{s^2 + 5s + 7}{(s + 2)^2}$ pode ser declarada da seguinte forma :

$N[s] = "(s^2+5s+7)"$, $D[s] = "(s+2)^2"$, $d = "0"$. Uma forma equivalente de declarar esta mesma função de transferência é realizada efetuando-se a divisão dos polinômios:

$$\begin{array}{r} s^2+5s+7 \quad | \quad s^2+4s+4 \\ -s^2-4s-4 \quad | \\ \hline s+3 \end{array}$$

Portanto $G(s)$ pode também ser descrita por: $N[s] = "(s+3)"$, $D[s] = "(s^2+4s+4)"$, $d = "1"$.

A função de transferência do exemplo 1 não contém parâmetros variáveis no tempo. A modelagem de um bloco DLC variável no tempo é realizada com o auxílio de variáveis auxiliares referenciadas por " k_0 ", " k_1 ", ..., " k_9 ", portanto dez no total. Para cada variável são definidos valores mínimo, médio (inicial) e máximo, delimitando sua faixa de excursão.

Ex. 2: Certo processo é descrito pela seguinte função de transferência:

$$G(s) = \frac{K}{(1+Ts)(1+2s)^2}$$

onde os valores de "K" e "T" são dados pela tabela abaixo:

	K	T
mínimo	0.56	4.7
médio	0.70	5.3
máximo	0.74	7.5

A representação deste processo é feita por um bloco DLC_FT: $N[s] = "(k_0)"$, $D[s] = "(1 + k_1 s) (1 + 2 s)^2"$, $d = 0$, $k_{0min} = 0.56$, $k_{0med} = 0.7$, $k_{0max} = 0.74$, $k_{1min} = 4.7$, $k_{1med} = 5.3$ e $k_{1max} = 7.5$. Portanto, neste exemplo, a variação do ganho K é realizada por "k₀" e a da constante de tempo T por "k₁".

A entrada de valores para as variáveis "k" é realizada através dos campos "Kn mín:", "Kn méd:" e "Kn máx:", sendo que os botões com as setas para cima e para baixo alteram o índice "n" da variável.

Os dados referentes a um bloco DLC_FT podem ser armazenados em arquivo e lidos posteriormente. Estas operações são realizadas através do painel de arquivamento ativado pelo botão "Carregar/Gravar...".

1.2.2.1.2. EQUAÇÕES DE ESTADO

Um bloco DLC descrito por equações de estado (DLC_EE) permite a declaração de uma dinâmica linear contínua de até terceira ordem, devido à existência de apenas três variáveis de estado em seu painel de declaração, conforme se observa na figura 2.3.

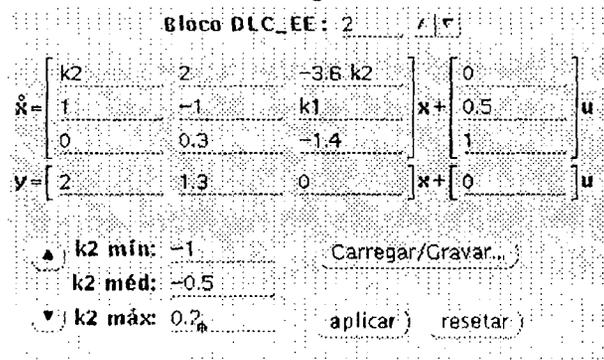


Fig. 2.3: Bloco DLC_EE :

De forma idêntica ao bloco DLC_FT, há um número entre 0 e 9 associado a este bloco, e declarado através do campo "Bloco DLC_EE:".

As matrizes A, B, C e D que compõem as equações de estado são declaradas nos campos seguintes. A entrada dos elementos de cada matriz se dá através de números reais e/ou variáveis auxiliares. As regras sintáticas para entrada destes dados, conforme padrão EBNT, são dadas por:

```

elemento = (real [termo_k]) | termo_k.
real = {" "} ("0" "." {algarismo} | "." algarismo {algarismo} | alg_int {algarismo} [{"." {algarismo}}] [expoente] {" "}.
expoente = "e" [{"+"|"-} alg_int {algarismo}].
termo_k = {" "} "k" algarismo {" "}.
algarismo = ("0"|alg_int).
alg_int = ("1"|2|3|4|5|6|7|8|9").
  
```

Conforme descrito na seção anterior, as variáveis auxiliares são utilizadas para realizar alterações em parâmetros do processo. A faixa de excursão destas variáveis é declarada através dos campos "**Kn mín:**", "**Kn méd:**" e "**Kn máx:**", sendo que os botões com as setas para cima e para baixo alteram o índice "n" da variável.

Os dados referentes a um bloco **DLC_EE** podem ser armazenados em arquivo e lidos posteriormente. Estas operações são realizadas através do painel de arquivamento ativado pelo botão "**Carregar/Gravar...**".

1.2.2.2. DECLARAÇÃO DO BLOCO DLD

O bloco de dinâmica linear discreta (DLD) modela um sistema descrito por uma função de transferência amostrada no tempo. Seu painel de declaração é apresentado na figura 2.4.

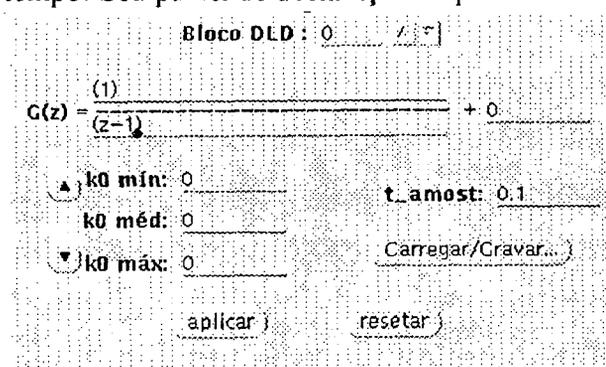


Fig. 2.4: Bloco DLD

Com relação ao bloco **DLC_FT** as únicas diferenças são:

- A função de transferência é em "z" ($G(z)$) e não em "s" ($G(s)$);
- Há um tempo de amostragem associado ao bloco, declarado através do campo "**t_amost:**".

1.2.2.3. DECLARAÇÃO DO BLOCO ATR

O bloco de atraso de transporte (ATR) modela um tempo de atraso inserido na estrutura do processo. A figura 2.5 apresenta seu painel de declaração.

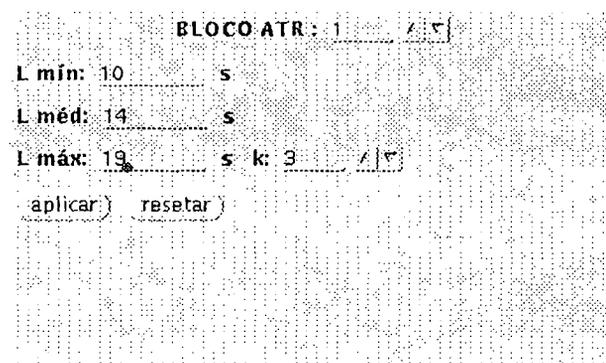


Fig. 2.5: Bloco ATR

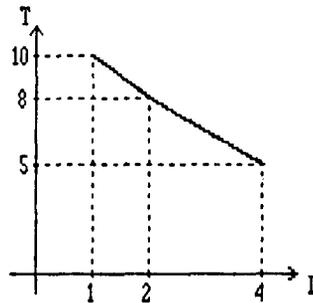
Cada bloco **ATR** possui um número entre 0 e 9 associado a si, de forma equivalente aos blocos de dinâmica linear (**DLC** e **DLD**). Isto é realizado através do campo "**Bloco ATR :**". O atraso é determinado por uma faixa de valores definida por $L_{mín}$, $L_{méd}$ (inicial) e $L_{máx}$, todos em segundos.

É possível associar alterações no tempo de atraso com alterações em parâmetros de um bloco de dinâmica linear. Para isto é necessário que os números dos blocos sejam iguais e que o atraso seja referenciado pela mesma variável "k" que o parâmetro desejado. O campo "k:" define o índice (entre 0 e 9) da variável "k" associada ao atraso.

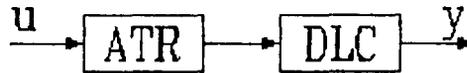
Ex. 3: A função de transferência de um dado processo é:

$$G(s) = \frac{(1+2s)e^{-Ls}}{s(1+Ts)}$$

,onde os valores de "L" e "T" estão relacionados conforme o gráfico abaixo:



Este processo é formado pela seguinte estrutura:



,devendo o número de ambos os blocos ser igual, por exemplo 3, e a variável "k" ser a mesma, por exemplo k1.

Com isto teremos:

Bloco ATR : 3

Bloco DLC : 3

L mín: 1 s

L méd: 2 s

L máx: 4 s k: 1

$$G(s) = \frac{(1+2s)}{(s)(1+10s-k1s)} + 0$$

k1 mín: 0

k1 méd: 2

k1 máx: 5

1.2.2.4. DECLARAÇÃO DO BLOCO GNL

Os blocos de ganho não linear (GNL) modelam curvas não lineares de ganho. Esta modelagem é realizada através de uma curva segmentada definida por pontos de quebra (figura 2.6).

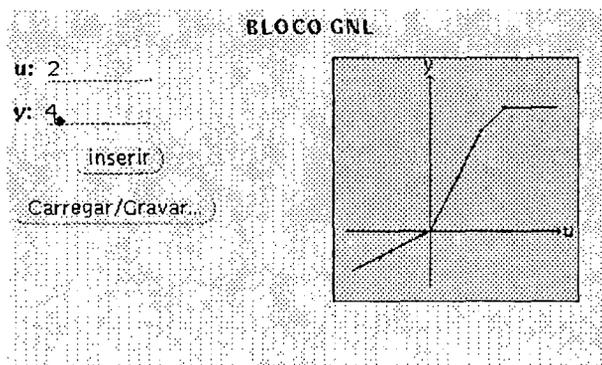
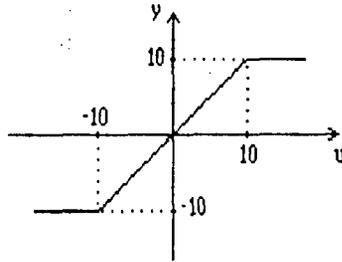


Fig. 2.6: Bloco GNL

Os pontos de quebra são declarados nos campos "u:" e "y:" (abscissa e ordenada, respectivamente) e inseridos através do botão "inserir", até um máximo de trinta pontos.

Ex. 4: Para modelar uma curva de saturações positiva e negativa em 10 unidades, realiza-se o gráfico abaixo:



Isto é feito com a inserção dos seguintes pontos:

u	-11	-10	10	11
y	-10	-10	10	10

A curva segmentada pode ser vista no gráfico existente à direita do painel. Com o cursor sobre este gráfico pode-se pressionar o botão esquerdo do mouse para visualizar um ponto de quebra, com seus valores de abscissa e ordenada aparecendo nos campos "u:" e "y:" respectivamente. Os demais pontos de quebra são vistos utilizando-se as teclas de direção (direita e esquerda) existentes a direita do teclado. Pode-se eliminar um ponto de quebra pressionando-se a tecla <cut> quando este ponto estiver sendo visualizado.

A curva segmentada pode ser armazenada em arquivo, através do painel de arquivamento ativado pelo botão "Carregar/Gravar..."

1.2.2.5. DECLARAÇÃO DO BLOCO PRT

Um bloco PRT corresponde a um ponto de entrada de sinais de perturbação, definido na estrutura do processo. A perturbação será adicionado ao sinal do sistema, podendo o primeiro ser uma combinação dos seguintes tipos de sinais: degrau, rampa, pulso, seno e ruído.

A declaração de um bloco PRT consiste simplesmente em associar um número (entre 0 e 9) ao mesmo, através do campo "Bloco PRT:" (figura 2.7). Convém salientar que este número não tem ligação com os dos blocos DLC, DLD e ATR. As amplitudes das perturbações serão definidas posteriormente, durante a simulação.



Fig. 2.7: Bloco PRT

I.3 : SIMULAÇÃO MANUAL

No capítulo anterior abordou-se as declarações do controlador e do processo. Neste será visto como realizar a simulação do sistema mediante intervenções do usuário. Seleciona-se as operações descritas no presente capítulo através de "**Modo de operação : Simulação Manual**".

Uma simulação consiste na evolução da dinâmica do processo, frente a atuações do controlador, durante um período de tempo. O transcorrer de uma simulação pode ser acompanhado continuamente, pelo usuário, através de dois gráficos (Y e U) presentes na tela do programa Sadeca (figura 3.1). Acima destes gráficos encontra-se o painel de comando, utilizado para realizar certas operações sobre a simulação, tais como iniciar, pausar e interromper. À esquerda dos gráficos estão situados os painéis de alteração, através dos quais o usuário interage com a simulação.

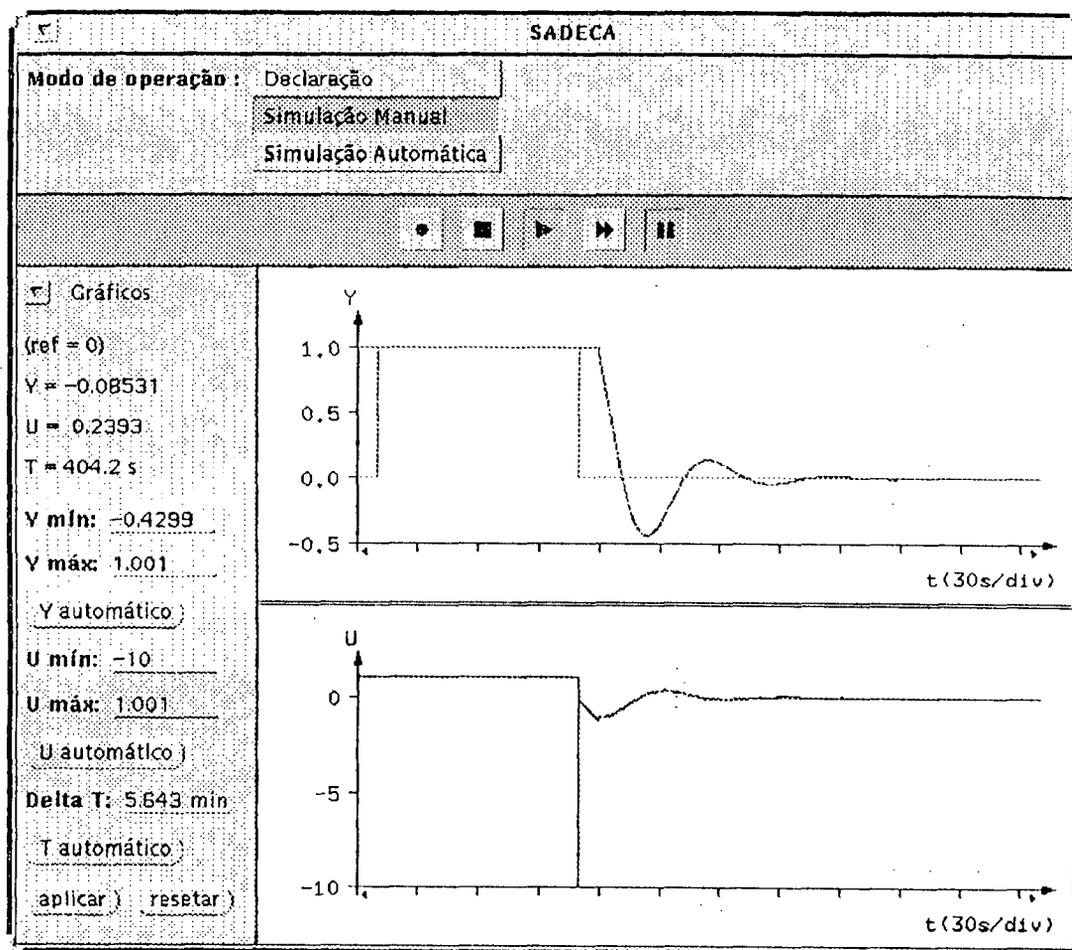


Fig. 3.1: Tela de Simulação Manual

I.3.1. PAINEL DE COMANDO

O painel de comando da simulação é composto por cinco botões contendo símbolos semelhantes ao de um toca-fitas comum, o que facilita a memorização das operações realizadas por estes:

-  aciona o painel de arquivamento dos gráficos. Além das operações padrão, descritas no capítulo 1, este painel contém um botão ("**imprimir**") que aciona a impressão dos gráficos Y e U em um arquivo denominado IMPR, o qual poderá ser enviado a uma impressora do tipo Epson 24 agulhas (Ex.: EPSON LQ 870).
-  finaliza uma simulação. O acionamento deste botão finaliza a simulação em curso, preparando o sistema para o início de uma nova simulação.
-  inicia uma simulação em tempo real. Ao iniciar uma simulação, limpa-se todas as informações anteriores, fazendo com que o sistema parta do repouso. A característica de tempo real não é garantida.
-   acelera a simulação ao máximo da capacidade da CPU. Pode-se retornar à simulação em tempo real pressionando-se, novamente, qualquer um destes dois botões.
-  pausa/prossegue a simulação. Este botão faz interromper, temporariamente, uma simulação, de forma a que se realize alterações e/ou análises.

1.3.2. GRÁFICOS

O acompanhamento da evolução da simulação é realizado através dos gráficos Y e U. No gráfico Y são plotados o sinal de saída do processo (antes do conversor A/D), na cor azul, e o sinal de referência, na cor verde, enquanto que no gráfico U é plotado o sinal de controle (após o conversor D/A), em azul.

As escalas dos gráficos são determinadas por dois fatores: a) valores desejados, definidos pelo usuário e b) tamanho da tela do programa. Sempre que o usuário alterar o tamanho da tela do programa Sadeca, os gráficos acompanharão esta mudança, podendo haver alterações nas escalas de tempo e amplitude vigentes, as quais tendem a se aproximar o máximo possível das escalas desejadas pelo usuário.

Inicialmente os gráficos Y e U possuem escalas de amplitude idênticas às dos conversores A/D e D/A, respectivamente. As escalas de tempo dos gráficos serão sempre idênticas entre si, sendo o valor inicial calculado em função das constantes de tempo e atrasos de transporte (valores médios) do processo declarado.

Os sinais simulados são plotados da esquerda para a direita dos gráficos, e ao atingir o limite da escala de tempo, desloca-se o sinal para a esquerda de forma a que se possa desenhar as informações mais recentes, desaparecendo da tela valores antigos. Este procedimento permite ao usuário um acompanhamento contínuo da simulação.

Os pontos que compõem os sinais apresentados nos gráficos são armazenados em vetores de tamanho limitado. À medida que o tempo de simulação aumenta, reduz-se a resolução destes sinais de forma a poder armazenar os novos pontos.

Com a simulação pausada ou interrompida é possível ampliar uma região do gráfico, bastando para isto: a) conduzir o cursor (com o mouse) até o início da região; b) pressionar o botão central do mouse; c) mantendo este botão pressionado, conduzir o cursor até o final da região; d) soltar o botão. Em uma ampliação, as escalas de amplitude são automaticamente ajustadas. Para visualizar toda a simulação realizada até o momento, deve-se selecionar a região compreendida pelos extremos da escala de tempo.

Outro recurso existente é o de aumento gradativo da escala de tempo, utilizando-se para isto o botão central do mouse, que deve ser pressionado quando o cursor estiver antes ou após a região em que o sinal está plotado.

Pequenas setas vermelhas existentes nos cantos inferiores esquerdo e direito do gráfico indicam que o sinal simulado vai além dos limites da escala de tempo vigente. Não é possível acompanhar o avanço de uma simulação enquanto a seta do canto direito estiver visível.

1.3.3. PAINÉIS DE ALTERAÇÃO

A interação do usuário com a simulação do sistema é realizada através dos painéis de alteração dos gráficos, controlador, processo e excitações, situados a esquerda dos gráficos Y e U. A cada momento apenas um destes painéis estará visível, apresentando os campos relativos à alteração do objeto em questão. Para selecionar o painel desejado utiliza-se o menu situado na parte superior da área reservada a estes painéis.

1.3.3.1. ALTERAÇÃO NOS GRÁFICOS

Os primeiros campos do painel de alteração dos gráficos (figura 3.2) são utilizados para apresentação de valores destes gráficos Y e U. Estes valores são: referência (ref), saída do processo (Y), saída do controlador (U) e tempo (T) em segundos. Durante a simulação estes campos apresentam os valores atuais de cada uma destas variáveis. Com a simulação pausada ou finalizada pode-se visualizar quaisquer valores do gráfico, bastando para isto conduzir o cursor (com o mouse) sobre as curvas. Pode ser observada uma certa "imantação" do cursor em relação às curvas, o que auxilia neste processo de análise. A movimentação ponto a ponto se dá pressionando o botão esquerdo do mouse (para fazer com que o gerenciador de eventos fixe sua atenção na área dos gráficos) e posteriormente as teclas de direção (direita e esquerda) existentes a direita do teclado. As teclas de direção para cima e para baixo fazem o cursor alternar-se entre os gráficos Y e U.

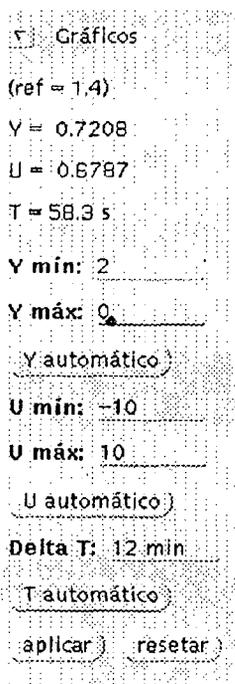


Fig. 3.2: Painel de Alteração dos Gráficos

Os valores desejados para os limites dos gráficos são definidos através dos campos "Y mín:", "Y máx:", "U mín:", "U máx:" e "Delta T:", e vigoram a partir do momento em que o botão aplicar é pressionado.

Os botões "Y automático" e "U automático" fazem com que os gráficos de Y e U tenham limites mais apropriados possíveis para apresentação das curvas atualmente visíveis. O botão "T automático" redefine a escala de tempo para uma boa apresentação de toda a curva de simulação. Estes botões atuam imediatamente.

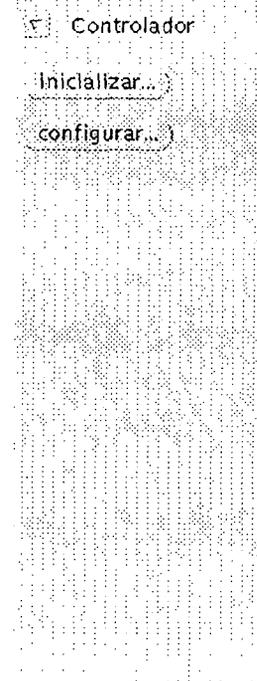


Fig. 3.3: Painel de Alteração do Controlador

1.3.3.2. ALTERAÇÃO NO CONTROLADOR

O painel de alteração do controlador (figura 3.3) é composto por dois botões: "inicializar" e "configurar", os quais acionam os mecanismos de inicialização e configuração do controlador adaptativo sob teste. O primeiro

mecanismo é utilizado para inicializar os parâmetros do controlador adaptativo, e o segundo para definir os objetivos de controle (ex.: máximo sobrepasso e tempo de subida), alarmes, etc.

Cada estrutura adaptativa implementa de forma diferenciada estes mecanismos, e a entrada de dados, por parte do usuário, é realizada através de painéis com moldura independente do programa principal.

1.3.3.3. ALTERAÇÃO NO PROCESSO

Conforme visto no capítulo anterior, os blocos DLC, DLD e ATR podem ter seus parâmetros alterados, desde que associados a variáveis auxiliares. Além disto estes blocos possuem um número para referência.

O primeiro campo do painel de alteração do processo (figura 3.4) seleciona o número do bloco e o segundo seleciona a variável "k" a ser alterada. O terceiro item (slider) é o responsável pela definição de um novo valor para a variável em questão.

Todas as variáveis têm associadas a si três valores: mínimo, médio e máximo, os quais correspondem respectivamente a -100, 0 e 100 no slider. Valores negativos intermediários fazem com que "k" assuma um valor proporcional entre k_{\min} e $k_{\text{méd}}$. De forma semelhante um valor positivo intermediário faz com que "k" assumira um valor proporcional entre $k_{\text{méd}}$ e $k_{\text{máx}}$, conforme figura 3.5.

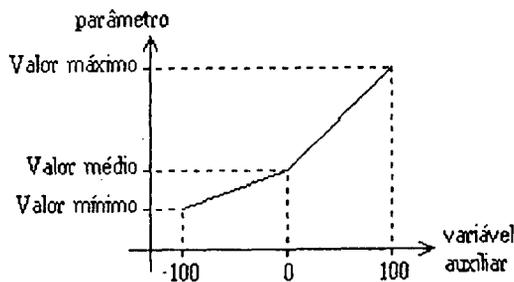


Fig. 3.5: Gráfico de alteração dos parâmetros

Sintetizando:

$$k = \begin{cases} k_{\text{méd}} + (k_{\text{máx}} - k_{\text{méd}}) * \text{slider} / 100, & \text{para } 0 < \text{slider} < 100; \\ k_{\text{méd}} + (k_{\text{méd}} - k_{\text{mín}}) * \text{slider} / 100, & \text{para } -100 < \text{slider} \leq 0. \end{cases}$$

Este procedimento faz com que processos completamente diferentes possam ter seus parâmetros alterados de uma forma genérica.

1.3.3.4. ALTERAÇÃO NAS EXCITAÇÕES

As excitações do sistema são compostas pela referência e pelas entradas de perturbação, podendo estas perturbações serem dos tipos: degrau, rampa, pulso, senóide e ruído. A figura 3.6 apresenta o painel de alteração destas excitações.

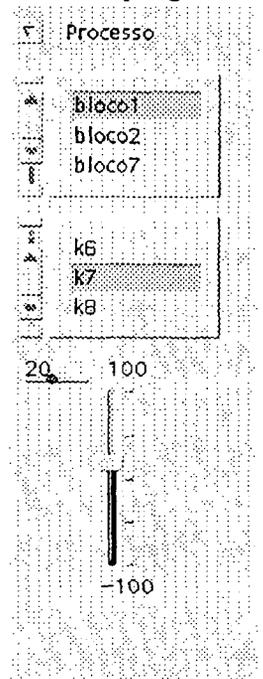


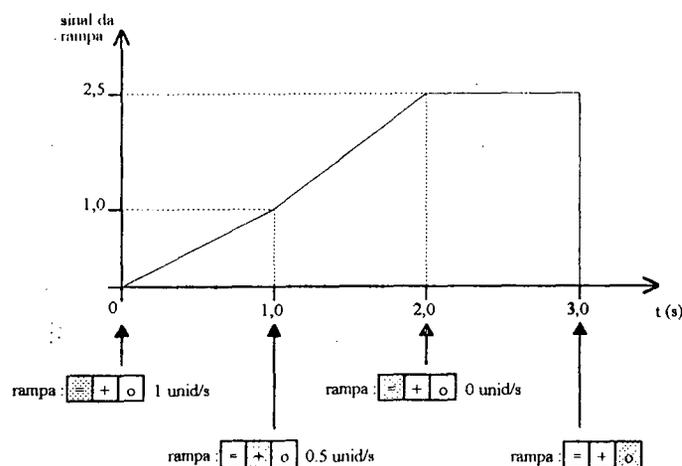
Fig. 3.4: Painel de Alteração do Processo

Ao iniciar nova simulação, todas as excitações têm seus valores zerados. Para fazer com que um ruído, por exemplo, atue desde os primeiros instantes de uma simulação, deve-se: a) pressionar a tecla de pausa e em seguida a de avanço, ambas do painel de comando; b) definir o valor do ruído e finalmente c) soltar a tecla de pausa.

O valor da referência é definido através do primeiro campo do painel: "**Referência:**". O campo seguinte é um menu contendo uma lista das entradas de perturbação declaradas para o processo. Os demais campos contém os dados referentes a entrada de perturbação selecionada através do menu. Estes dados descrevem cada um dos tipos de perturbação:

***TERMO CONSTANTE:** as perturbações em forma de degrau são realizadas através de alterações neste termo constante que é adicionado ao sinal de entrada do bloco PRT. O termo constante pode ser setado em determinado valor, ou sofrer um incremento especificado, conforme seja a posição do seletor " = + ". Ex: suponhamos que o valor atual do termo constante seja 3.4 e que desejamos realizar um degrau unitário negativo nesta entrada de perturbação. Para isto existem duas possibilidades: 1) Setar (=) o valor do termo constante em 2.4 ou 2) Incrementar (+) o valor deste em -1 unidades.

***RAMPA:** uma perturbação em rampa é definida em unidades por segundo, e seu valor, da mesma forma que o termo constante, pode ser setado ou incrementado. O sinal da rampa é acumulado com o passar do tempo, e ao cessar a rampa permanece um valor constante adicionado ao sinal de entrada. Para zerar este valor constante utiliza-se a tecla "o" do seletor " = + o ". Ex.:



***PULSO:** um pulso é definido através de sua amplitude e sua duração (em segundos).

***SENÓIDE:** o sinal senoidal é definido por sua amplitude e período (em segundos).

***RUÍDO:** após adicionados, ao sinal de entrada, todos os sinais de perturbação anteriormente descritos, acrescenta-se o sinal de ruído gerado via números randômicos. Este sinal de ruído possui uma distribuição normal de amplitude, com média nula e desvio padrão definido pelos campos do painel. O primeiro valor corresponde ao termo absoluto e o segundo define o termo percentual, relativo à amplitude do sinal ao qual o ruído será acrescentado. Ex.: suponhamos que o sinal na entrada do bloco PRT seja de 2 unidades. Acrescenta-se a este um termo constante de 0.5. Não há rampa, pulso ou seno, e o ruído é definido como sendo: "**Ruído: 0.05 + 1%**". Com isto será acrescentado um valor aleatório, ao sinal de 2.5 unidades, de desvio padrão: $0.05 + 0.01 * 2.5 = 0.075$.

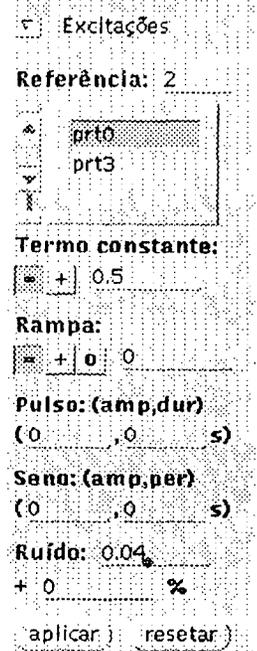


Fig. 3.6: Painel de Alteração das Excitações

I.4 : SIMULAÇÃO AUTOMÁTICA

No capítulo precedente abordou-se a simulação manual, onde as alterações sobre o sistema eram realizadas pelo usuário. Neste capítulo é descrito o mecanismo de simulação automática que, através da execução de seqüências de eventos (alterações no controlador, processo e excitações) definidas em arquivos, permite avaliar o comportamento da estrutura de controle adaptativo atuando em variadas condições, sobre um conjunto de processos.

Na tela de simulação automática apresentada na figura 4.1 é possível identificar um painel de comando na região central e gráficos Y e U na parte inferior. Estes gráficos comportam-se de maneira idêntica aos da simulação manual no que concerne à apresentação do sinal, embora não possuam recursos para análise destas curvas.

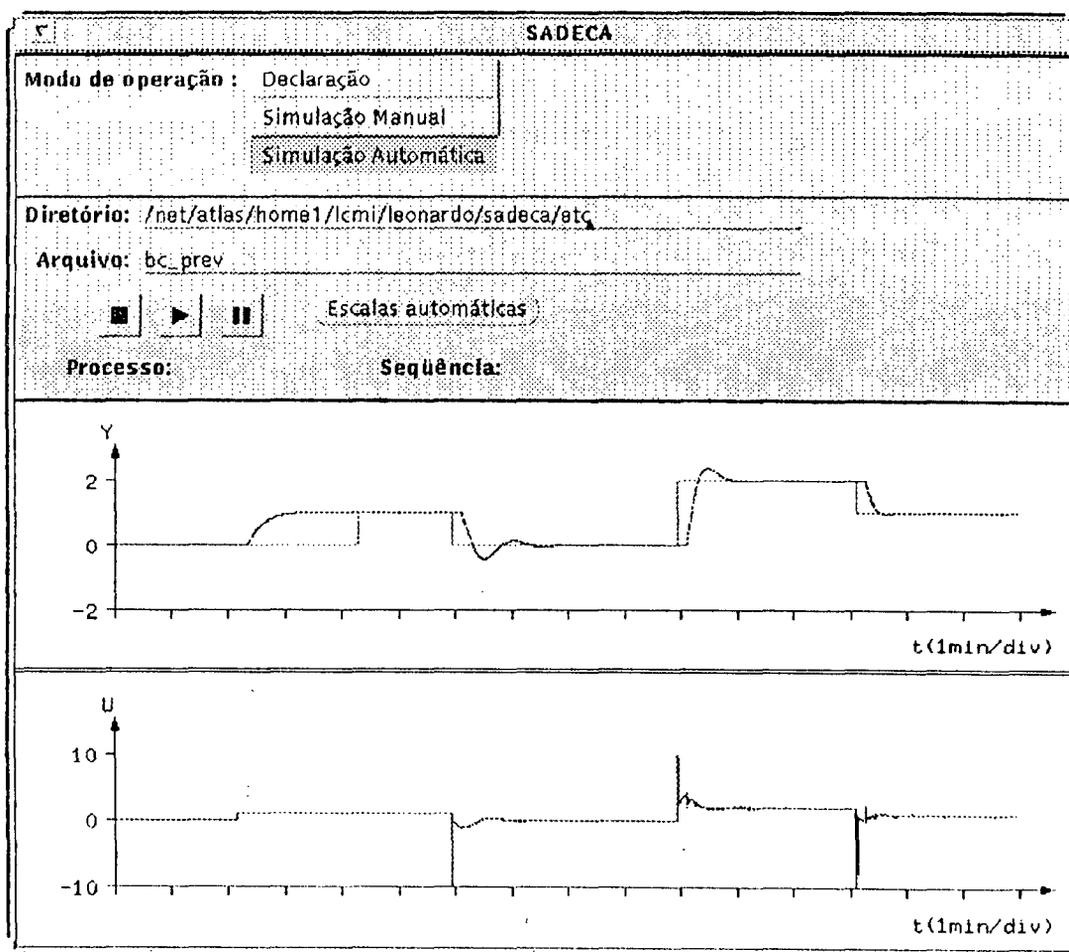


Fig. 4.1: Tela de Simulação Automática

Nas primeiras seções deste capítulo descreve-se a seqüência de eventos (seção 4.1) e o banco de provas (seção 4.2), que são os elementos básicos da simulação automática. Posteriormente apresenta-se o painel de comando (seção 4.3) deste modo de operação e, finalmente, descreve-se os relatórios das simulações (seção 4.4).

I.4.1. SEQÜÊNCIA DE EVENTOS

Uma seqüência de eventos é composta por um conjunto de instruções (eventos) que são executados seqüencialmente. Estes eventos compreendem as possibilidades que o usuário possuía para

atuar sobre o sistema (controlador, processo e excitações) no modo de simulação manual, além de outros mecanismos utilizados para avaliação de desempenho da estrutura adaptativa.

Uma das principais características da seqüência de eventos é a genericidade, o que permite sua utilização sobre um amplo conjunto de processos. Para que isto seja possível, as medidas de tempo utilizadas para descrever a seqüência devem ser relativas ao tempo de resposta do processo sob simulação. Este fato conduziu à criação de uma unidade de tempo, denominada TMF (tempo de malha fechada), calculada em função das constantes de tempo e atrasos de transporte do processo. Salienta-se que esta unidade de tempo TMF possui uma grandeza variável durante a simulação, devido a mudanças de parâmetros do processo.

O tempo de 1 (um) TMF é, aproximadamente, o tempo necessário para que o processo atinja o regime permanente, em malha fechada, frente a uma mudança de referência do tipo degrau, utilizando como controlador um PID com ajuste próximo ao de mínimo ITAE (integral do erro absoluto multiplicado pelo tempo).

As seqüências de eventos são descritas em arquivos editados pelo usuário, utilizando para isto qualquer editor de textos disponível (vi, emacs, etc.). Cada evento é delimitado por parênteses ("(" e ")"), podendo haver qualquer quantidade de espaçadores (espaço, tabulação ou linhas em branco) entre os eventos.

Abaixo é dada a descrição de cada um dos tipos de eventos, onde:

$n, i, j \in \mathbb{N}; i, j \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}; n \geq 1;$

$x, v, f, t, z \in \mathbb{R}; x \in [-100; 100]; f, t, z \in [0; +\infty).$

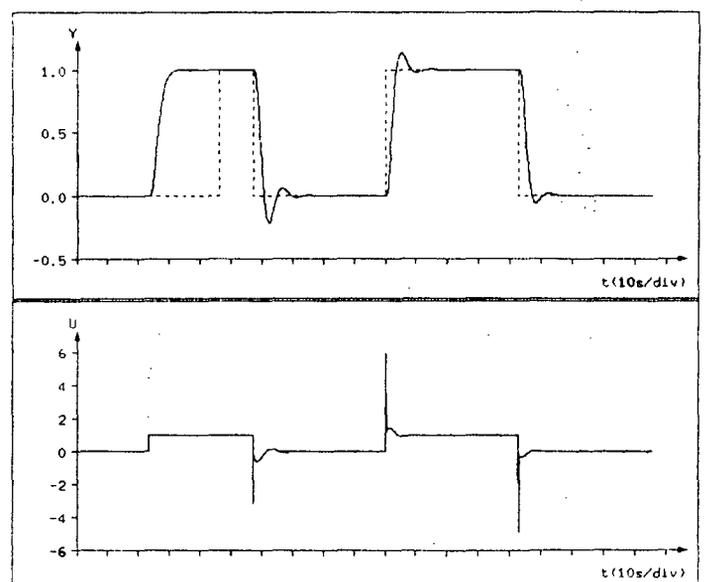
- (I) \Rightarrow realiza a inicialização do controlador. Se houver necessidade de informações preliminares, será apresentado ao usuário um painel independente contendo os campos relativos à entrada destes dados.
- (ID) \Rightarrow realiza a inicialização do controlador. As informações preliminares, necessárias à inicialização, assumirão valores "default".
- (C) \Rightarrow realiza a configuração do controlador. A entrada de dados para configuração do controlador é realizada através de um painel independente.
- (P $_{ij}=x$) \Rightarrow realiza alterações em parâmetros do processo. A variável k_j do bloco i assumirá um valor proporcional a x , conforme descrito na seção 3.3.3.
- (R+v) \Rightarrow acrescenta à referência, o valor v ($ref_atual = ref_antiga + v$).
- (R-v) \Rightarrow diminui da referência, o valor v ($ref_atual = ref_antiga - v$).
- (R;v) \Rightarrow alterna acréscimos e diminuições de valor v , à referência. Este tipo de evento é, geralmente, utilizado dentro de loops.
- (R=v) \Rightarrow estabelece o valor v para a referência ($ref_atual = v$).
- (Q $_i$ C+v) \Rightarrow acrescenta ao termo constante da perturbação i , o valor v .
- (Q $_i$ C-v) \Rightarrow diminui do termo constante da perturbação i , o valor v .
- (Q $_i$ C;v) \Rightarrow alterna acréscimos e diminuições de valor v , ao termo constante da perturbação i . É geralmente utilizado dentro de loops.
- (Q $_i$ C=v) \Rightarrow estabelece o valor v para o termo constante da perturbação i .

- (QiR+v)** => acrescenta o valor v (em unidades por TMF) à taxa de crescimento da rampa da perturbação i.
- (QiR-v)** => diminui em v unidades/TMF a taxa de crescimento da rampa da perturbação i.
- (QiR=v)** => estabelece o valor v (em unidades por TMF) para a taxa de crescimento da rampa da perturbação i.
- (QiRL)** ==> zera o valor constante que se acumula com o passar do tempo, em função da existência de rampas (ver seção 3.3.4).
- (QiP=v;t)** ==> estabelece um pulso na perturbação i, de amplitude v e duração t (em TMF).
- (QiS=v;t)** ==> estabelece para o seno da perturbação i uma amplitude v e período t (em TMF).
- (QiS=v;f)** ==> estabelece para o seno da perturbação i uma amplitude v e frequência f (em Hz).
- (QiN=v+z%)** => estabelece um sinal de ruído na perturbação i, de amplitude média v acrescida de z por cento do sinal afetado.
- (Ln=(evento 1)(evento 2) ...)** = realiza n repetições do conjunto de eventos definidos em seu corpo.
- (Sn=(evento 1)(evento 2) ...)** => realiza até n repetições do conjunto de eventos definidos em seu corpo. Após cada repetição o controlador adaptativo é arguido se já está ajustado, o que em caso afirmativo encerra o loop sinalizado.
- (Tt)** => realiza a simulação durante um tempo t (em TMF).
- (A)** => inicia o processo de análise através da medição dos valores ISE (integral do erro quadrático: $\int e^2(t)dt$), ITAE (integral erro absoluto multiplicado pelo tempo: $\int t|e(t)|dt$), ISU (integral do sinal de controle elevado ao quadrado: $\int u^2(t)dt$) e tempo de análise (tf). A ocorrência de um novo evento (A) finaliza o processo de medição.

Uma típica seqüência de eventos para análise do ajuste inicial de um controlador adaptativo é apresentada abaixo. Inicialmente ajusta-se uma variável auxiliar do processo e realiza-se a inicialização. A seguir efetua-se um loop sinalizado com limite máximo de quinze ciclos, no qual alterna-se variações unitárias no valor da referência.

```
(P01=-80)
(ID)
(T1)
(S15=(R;1)
(T3))
```

Um gráfico resultante desta seqüência de eventos é apresentado ao lado. Inicialmente, o controlador aplica um degrau em malha aberta para reconhecimento do processo e inicialização de seus parâmetros. A seguir são efetuados dois ciclos até que o controlador consiga ajustar-se às especificações de controle, o que é detectado na terceira mudança de referência.



1.4.2. BANCO DE PROVAS

O Banco de Provas é composto por um conjunto de processos, um conjunto de seqüências de eventos e uma tabela que define o conjunto de experimentos a ser realizados, conforme observado na fig. 4.2. Define-se por **experimento** a execução de uma dada seqüência de eventos sobre um dado processo.

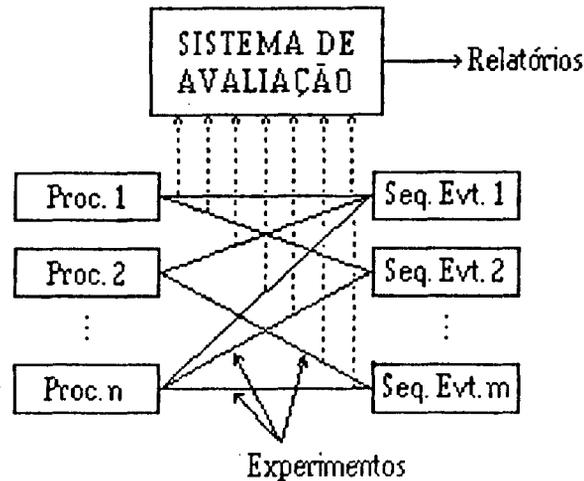


Fig. 4.2: Banco de Provas

Da mesma forma que as seqüências de eventos, os bancos de provas são descritos em arquivos, editados pelo usuário. Seu conteúdo é apresentado abaixo:

```
#SEQUENCIAS
<seqüência 1>
<seqüência 2>
...
<seqüência n>

#PROCESSOS
<processo 1>
<processo 2>
...
<processo m>

#TABELA
XX-X ... X
-XX- ... -
  ⋮
-XX- ... X
} n
m
```

Após a linha "#SEQUENCIAS" são editados os nomes das seqüências de eventos, formando desta forma o conjunto de seqüências. Após estes deve haver uma linha em branco e outra com a seguinte cadeia de caracteres: "#PROCESSOS", seguida dos nomes dos arquivos contendo os processos que formam o conjunto de processos.

Abaixo dos processos edita-se a tabela que irá comandar a execução ou não de combinações seqüência-processo. As posições da tabela com o caracter "X" indicam a execução da combinação

correspondente, enquanto que o caracter "-" faz com que a combinação não seja executada. A tabela terá tantas linhas quantas forem as seqüências e tantas colunas quantos forem os processos.

1.4.3. PAINEL DE COMANDO

Os primeiros campos do painel de comando: "**Diretório:**" e "**Arquivo:**", especificam o banco de provas a ser executado (caminho e nome, respectivamente). A execução do banco de provas é comandada pelos botões semelhantes aos da simulação manual, cujas funções são descritas abaixo.

-  finaliza a simulação (execução do banco de provas).
-  inicia uma simulação com a máxima capacidade da CPU.
-  pausa/prossegue a simulação.

Durante a simulação os gráficos dos sinais de saída do processo e saída do controlador podem ser observados na parte inferior da tela. Para realizar um ajuste nas escalas destes gráficos basta pressionar o botão "Escalas Automáticas", o que permite uma visualização de toda a simulação até o presente momento, com as melhores escalas de amplitude possível. As execuções dos comandos apresentados até o momento não são realizadas de forma imediata, pois a CPU permanece dedicada à simulação durante quase todo o tempo, fazendo com que o tratamento de eventos seja realizado de forma escassa.

O processo e a seqüência que compõem o experimento em curso possuem seus nomes escritos nos campos "**Processo:**" e "**Seqüência:**", permitindo ao usuário um acompanhamento da execução do banco de provas.

Os gráficos de cada experimento realizado são armazenados em arquivos, permitindo uma análise posterior dos mesmos. Os nomes dos arquivos que contém estes gráficos são formados da seguinte forma:

graf_NomeDoProcesso_NomeDoControlador_NomeDaSeqüência

1.4.4. RELATÓRIOS DAS SIMULAÇÕES

Dois tipos de relatórios são gerados pelo pacote Sadeca: relatórios de experimento (RE) e relatório do banco de provas (RBP). Estes relatórios são armazenados em arquivos texto que podem ser diretamente lidos pelo usuário.

Os REs são gerados após a conclusão de cada experimento, sendo seus nomes formados da seguinte maneira:

relat_NomeDoProcesso_NomeDoControlador_NomeDaSeqüência

Um exemplo de relatório de experimento é mostrado na figura 4.3, onde se distinguem três áreas: inicialmente apresenta-se os dados relativos ao processo, em seguida os do controlador e, finalmente, os resultados dos eventos programados.

```

PROCESSO : Proc7
  Tempo de malha aberta (tma) = 17.84s
  Tempo de malha fechada (tmf) = 26.0397s

CONTROLADOR : PID-AA
  Conversor A/D : ( -10 / 10 ) 12 bits
  Conversor D/A : ( -10 / 10 ) 12 bits
  Tempo de amostragem = 0.1s

SEQUENCIA : set2_-90
  Inicializacao 1 : 45.9s
  [K=1.001 L=1.112 T=3.64 pi=35.19% L/T=0.3055]
  [(1.143 2.044 0.2661) (2.156 1.932 0.627)]
  Analise 1 : (ISE ITAE ISU IU tf)
  1.80854 8.51815 1.81985 -1.5127 42.9s
  [(1.143 2.044 0.2661) (2.156 1.932 0.627)]
  [So=0.2097 Su=0.05463 2So=0.01319 2Su=0 (G2) Ind:0.6923]
  [(1.091 2.044 0.4411) (2.156 1.932 0.627)]
  Analise 2 : (ISE ITAE ISU IU tf)
  1.70321 6.10202 48.6782 44.3872 42.9s
  [(1.091 2.044 0.4411) (2.156 1.932 0.627)]
  [So=0.1334 Su=0.01812 2So=0 2Su=0 (G5) Ind:0.8853]
  [(1.091 2.379 0.4411) (2.156 1.932 0.627)]
  Analise 3 : (ISE ITAE ISU IU tf)
  1.69724 4.88612 2.64307 -1.12012 42.9s
  [(1.091 2.379 0.4411) (2.156 1.932 0.627)]
  [So=0.05819 Su=0.01201 2So=0 2Su=0 (G0) Ind:0.9479]
  LOOP 1 : 2 ciclos
  [( 2 5 0 )]

```

Fig. 4.3: Arquivo "relat_Proc7_PID-AA_set2_-90"

No experimento acima foi realizada a execução da seqüência de eventos "set2_-90" sobre o processo "Proc7" sob atuação do controlador "PID-AA". Os tempos de malha aberta e malha fechada do processo são calculados em função das constantes de tempo e atrasos de transporte (com as variáveis auxiliares em seus valores médios).

A seqüência de eventos "set2_-90" consiste em:

- inicializar os parâmetros do controlador (Inicialização 1);
- realizar um loop sinalizado no qual se altera o valor da referência (LOOP 1);
- analisar o sinal de resposta em cada ciclo do loop (Análises 1, 2 e 3).

O "Sistema de Avaliação" (ver figura 4.2) é responsável pelo fornecimento das seguintes informações:

- Inicialização : tempo necessário à inicialização do controlador;
- Análises : medições de ISE, ITAE, ISU, IU e tempo de análise;
- LOOPS : número de ciclos necessários ao ajuste do controlador.

Além das informações providas pelo "Sistema de Avaliação", os relatórios de experimento contêm informações providas pelo controlador adaptativo sob teste, as quais são facilmente identificadas por estarem delimitadas por colchetes ("[" e "]").

O relatório do banco de provas é gerado ao final da execução de um banco de provas, e seu nome é formado da seguinte maneira:

relat_NomeDoBancoDeProvas

O principal objetivo do RBP é realizar uma análise "estatística" dos loops sinalizados presentes nas seqüências de eventos que compõem o banco de provas. Como conclusão final teremos o número médio, e o desvio padrão, de ciclos de ajuste para que o controlador consiga atingir os objetivos de controle. Um exemplo de RBP é apresentado na figura 4.4.

```

BANCO DE PROVAS : bc_tese

CONTROLADOR : PID-AA

Sequencia : seq1
  Processo : Proc3
    LOOP 1 : 1 Ciclos
    LOOP 2 : 3 Ciclos
    media = 2
  Processo : Proc6
    LOOP 1 : 6 Ciclos
    LOOP 2 : 15 Ciclos (nao ajustado)
    media = 10.5
    media = 6.25

Sequencia : seq5
  Processo : Proc3
    LOOP 1 : 7 Ciclos
    media = 7
  Processo : Proc6
    LOOP 1 : 4 Ciclos
    media = 4
    media = 5.5

Media Final (LOOPS ajustados) = 4.2 Ciclos (21/5)
Desvio Padrao = 2.135 Ciclos

LOOPS nao ajustados : 1

Media com LOOPS nao ajustados = 6 Ciclos (36/6)

```

Fig. 4.4: Arquivo "relat_bc_tese"

Este RBP contém as informações da execução do banco de provas "bc_tese", o qual consiste de quatro experimentos. São realizados, ao todo, seis loops sinalizados, sendo que para um destes o controlador "PID-AA" não conseguiu atingir as especificações de controle após os quinze ciclos programados. Nos cinco loops realizados com sucesso foram necessários vinte e um ciclos, o que resulta em uma média de 4,2 ciclos de ajuste. Considerando-se todos os seis loops, foram realizados trinta e seis ciclos, resultando em uma média de 6 ciclos de ajuste por loop.

As informações contidas nos relatórios apresentam-se suficientemente abrangentes para avaliar controladores baseados em reconhecimento de formas. O estudo de outros tipos de controladores adaptativos pode conduzir à definição de novos índices. No estudo comparativo de estruturas adaptativas, é possível definir uma ponderação para os índices de desempenho de forma a quantificar a performance global de cada estrutura.

I.5 : PROGRAMAÇÃO DE CONTROLADORES

Nos capítulos precedentes abordou-se todos os detalhes de utilização do pacote SADECA. Neste capítulo apresenta-se um tópico especial do programa, de interesse a uma limitada quantidade de usuários: a programação de novos controladores adaptativos. Para efetuar esta programação é necessário um certo conhecimento da linguagem C++ e do pacote XView.

A apresentação desta programação será realizada passo a passo, baseando-se em um exemplo de implementação de controlador, o qual denominaremos "EXMPL". Na seção 5.1 é apresentada a forma mínima do controlador e o procedimento para ligá-lo ao pacote SADECA. A seção seguinte, 5.2, apresenta os recursos que o controlador dispõe para a realização de simulações manuais. Concluindo-se este capítulo, com a seção 5.3, apresenta-se os recursos disponíveis para avaliação do controlador adaptativo, no modo de simulação automática.

I.5.1. IMPLEMENTAÇÃO MÍNIMA

Todo controlador deve ser implementado como uma nova classe, na linguagem C++, sendo esta herdeira da classe base "controlador". Através desta herança o novo controlador passará a conter conversores A/D e D/A, bem como uma interface de conexão com o pacote SADECA.

A descrição da nova classe estará contida em um arquivo de extensão ".h" e o conteúdo dos métodos estará no arquivo ".cc" correspondente. Em nosso exemplo estes arquivos serão denominados "exmpl.h" e "exmpl.cc". A primeira versão destes arquivos, figura 5.1, já pode ser editada, criando uma classe "EXMPL", com métodos construtor e destrutor, herdeira da classe "controlador".

"exmpl.h"	"exmpl.cc"
<pre>#ifndef controlador_exmpl_DEFINED #define controlador_exmpl_DEFINED #include "controlador.h" class EXMPL : public Controlador { public: EXMPL(); ~EXMPL(); double avancar(double); private: }; #endif ~controlador_exmpl_DEFINED</pre>	<pre>#include "exmpl.h" EXMPL::EXMPL() { } EXMPL::~EXMPL() { } double EXMPL::avancar(double c) { double u = 2.0 * c; return u; }</pre>

Fig. 5.1 : Versão Mínima de EXMPL

As duas linhas iniciais, de "exmpl.h", e a última são diretivas de compilação, cuja função é evitar que a classe "EXMPL" seja declarada mais de uma vez, durante uma compilação. O método "avancar" conterá uma nova implementação para o método virtual declarado na classe base "controlador". Este método é o principal de um controlador, pois sua função é fornecer um valor para o sinal de controle (U), dado o valor do erro entre a referência (Yr) e o sinal de saída do processo (Y). Como implementação mínima, o controlador EXMPL será simplesmente um amplificador de sinal de ganho fixo dois (2,0).

Com estes dois arquivos já se pode dizer que um novo controlador foi criado, restando apenas conectá-lo ao pacote de simulação. Esta conexão é realizada através da classe "Decl_Cont", bastando apenas adicionar algumas linhas ao arquivo "decl_cont.cc". A figura 5.2 apresenta em **negrito** estas linhas relativas ao controlador EXMPL, dando uma visão de suas posições dentro do arquivo.

```

#include "exmpl.h"
:
void
Decl_cont::inicializar(Frame frame1,
Controlador** cont)
{
:
tipo = (Panel_item)xv_create(painel1,
PANEL_CHOICE,
XV_HELP_DATA, "Sadeca:tipo",
PANEL_NEXT_ROW, 30,
PANEL_CHOICE_NCOLS, 2,
PANEL_LABEL_STRING, "Tipo:",
PANEL_CHOICE_STRINGS,
"PID",
"PID-AA",
"FUZZY",
"PID_Liu",
"GPC",
"Relé",
"PIP",
"EXMPL",
NULL,
NULL);
:
}
:
void
Decl_cont::destruir_controlador()
{
switch(tipo_atual)
{ case 0:
delete (Cont_PID*)*controlador;
break;
:
:
case 6:
delete (Cont_PIP*)*controlador;
break;
case 7:
delete (EXMPL*)*controlador;
break;
default:
break;
}
:
:
void
Decl_cont::alterar_cont(int tip)
{
destruir_controlador();
switch(tip)
{ case 0:
*controlador = new Cont_PID;
break;
:
:
case 6:
*controlador = new Cont_PIP;
break;
case 7:
*controlador = new EXMPL;
break;
default:
*controlador = new Cont_PID;
break;
}
(*controlador)->inic_interface(frame);
tipo_atual = tip;
}
}

```

Fig. 5.2 : Modificações no Arquivo "decl_cont.cc"

A forma mais simples e inteligente de comandar a compilação de novos controladores é através da inclusão dos nomes dos arquivos correspondentes, no arquivo "Makefile" do pacote SADECA. Este arquivo pode ser encontrado junto com o ambiente. Para promover a compilação e linkagem do pacote, basta executar o comando "make" no subdiretório em que se encontra o arquivo "Makefile". Realizando-se todos estes passos já é possível observar o comportamento do controlador EXMPL, dentro do ambiente SADECA. Alguns aperfeiçoamentos serão introduzidos na seção seguinte, de forma a prover recursos básicos ao controlador.

1.5.2. IMPLEMENTAÇÃO DOS RECURSOS BÁSICOS

Entende-se por recursos básicos os mecanismos envolvidos no desenvolvimento de um controlador, ou seja, todos os recursos que possam ser úteis, ao controlador, no modo de operação:

simulação manual. A apresentação destes recursos estará dividida entre as funções de inicialização e configuração, seção 5.2.1, e os conversores A/D e D/A, seção 5.2.2.

1.5.2.1. INICIALIZAÇÃO E CONFIGURAÇÃO

A primeira deficiência do controlador EXMPL é o valor fixo do ganho. É extremamente interessante ao usuário alterar o valor do ganho, de forma a melhorar o comportamento do sistema, em malha fechada. A forma disponível, ao usuário, de interação com o controlador é através dos painéis de *inicialização* e *configuração*, implementados com objetos do pacote de interface gráfica XView.

O valor do ganho é uma grandeza dependente do processo sob controle, portanto relativa à função de inicialização. Desta forma será criado um painel de inicialização com um campo para entrada do valor do ganho e o tradicional (para quem utiliza o ambiente OpenWindows) botão "aplicar". Isto parece simples, mas diversos agravantes tornam laboriosa esta atividade:

- Não há no pacote XView um objeto para entrada de dados em ponto flutuante. Contorna-se esta deficiência utilizando-se um objeto de entrada de seqüência de caracteres (string), a qual é posteriormente convertida no valor numérico correspondente.
- O pacote XView, implementado na linguagem C, não está preparado para realizar chamadas a métodos de classes (C++). A única forma de contornar isto é criar métodos estáticos (comuns a todos objetos de uma classe) e associar o ponteiro do objeto (this) da classe "EXMPL" ao objeto XView, através de "XV_KEY_DATA".

O que em palavras apresenta-se bastante confuso, talvez seja melhor esclarecido com o prosseguimento do exemplo. Na figura 5.3, a esquerda, pode ser vista uma nova listagem para o arquivo "exmpl.h", contendo a declaração de dois novos métodos virtuais da classe base: *inic_interface*, utilizado para criar os objetos da interface, e *inicializar*. Tudo que está em negrito, nesta figura, indica alteração em relação à listagem da figura 5.1.

"exmpl.h"	"exmpl.cc"
<pre>#ifndef controlador_exmpl_DEFINED #define controlador_exmpl_DEFINED #include "controlador.h" class EXMPL : public Controlador { public: EXMPL(); ~EXMPL(); double avancar(double); void inic_interface(Frame); void inicializar(char, double); private: double ganho; Frame subframe_inic; Panel_item item_ganho; static void aplicar(Panel_item); static void done_proc(Frame); }; #endif ~controlador_exmpl_DEFINED</pre>	<pre>#include <xview/panel.h> #include <string.h> #include "exmpl.h" EXMPL::EXMPL() { ganho = 1.; } EXMPL::~EXMPL() { xv_destroy_safe(subframe_inic); } double EXMPL::avancar(double e) { double u = ganho * e; return u; } void EXMPL::inic_interface(Frame frame)</pre>

```

{
    subframe_inic = (Frame)xv_create(frame, FRAME_CMD,
        FRAME_LABEL, "EXMPL (Inicialização)",
        FRAME_DONE_PROC, EXMPL::done_proc,
        XV_KEY_DATA, 10, this,
        NULL);
    Panel panel = (Panel)xv_get(subframe_inic,
        FRAME_CMD_PANEL);
    item_ganho = (Panel_item)xv_create(panel, PANEL_TEXT,
        PANEL_LABEL_STRING, "Ganho:",
        PANEL_VALUE, "1",
        NULL);
    (void)xv_create(panel, PANEL_BUTTON,
        PANEL_LABEL_STRING, "Aplicar",
        PANEL_NOTIFY_PROC, EXMPL::aplicar,
        XV_KEY_DATA, 10, this,
        NULL);
}

void
EXMPL::inicializar(char, double)
{
    xv_set(subframe_inic, XV_SHOW, TRUE, NULL);
}

void
EXMPL::done_proc(Frame subframe)
{
    EXMPL* este = (EXMPL*)xv_get(subframe,
        XV_KEY_DATA, 10);
    xv_set(este->subframe_inic, XV_SHOW, FALSE, NULL);
    este->fim_inic();
}

void
EXMPL::aplicar(Panel_item item)
{
    EXMPL* este = (EXMPL*)xv_get(item,
        XV_KEY_DATA, 10);
    char* string = (char*)xv_get(var[i], PANEL_VALUE);
    char** endptr = &string;;
    double valor = strtod(string, endptr);
    if (**endptr != NULL)
    { // Tratamento de erro, a critério do usuário.
    }
    else
        este->ganho = valor;
    este->fim_inic();
}

```

Fig. 5.3 : Segunda Versão de EXMPL

Observa-se na figura 5.3 que os métodos acionados por objetos XView, "aplicar" e "done_proc", são declarados estáticos. O método "aplicar" é chamado pelo botão de mesmo nome, contido no painel de inicialização. Sempre que o usuário encerrar uma inicialização (ou configuração), seja "aplicando" novos valores ou soltando o "PUSHPIN" de subframe, o ambiente deve ser informado,

o que é feito através da função **fim_inic (fim_conf)**. Este é o motivo pelo qual o subframe precisa do método "done_proc", conforme observado na listagem de "exmpl.cc".

Os métodos estáticos são comuns a todos objetos de uma classe, não contendo, portanto, referências a atributos de uma determinada instância de objeto. Uma referência (apontador) à instância da classe "EXMPL" é então passada para o método estático, via objetos XView.

No método "aplicar" a string que descreve o valor do ganho é convertida para um valor do tipo **double**, mas, por erro de digitação, o usuário poderia vir a declarar uma seqüência de caracteres não válida como número. Portanto deve ser realizado um tratamento de erros, geralmente via comunicação ao usuário através de "NOTICE" (ver listagens de controladores já implementados).

Diversas alterações podem ser realizadas sobre o painel de inicialização, no sentido de melhorar sua apresentação visual, bastando conhecer um pouco de programação com XView. O painel de configuração é criado exatamente da mesma forma que o painel de inicialização. Para iniciar uma configuração utiliza-se o método virtual "**void configurar()**", devendo a função **fim_conf** ser chamada no término desta configuração. Os métodos de início da *inicialização* e da *configuração* diferenciam-se apenas pelo fato do primeiro apresentar dois parâmetros, o que será explicado mais adiante, na seção 5.3.

A inicialização de um controlador adaptativo pode ser realizada mediante interação com o usuário ou através de um ensaio denominado pré-ajuste. Este pré-ajuste consiste na identificação de parâmetros de caracterização do processo e cálculo dos parâmetros do controlador. Um exemplo de pré-ajuste consiste na aplicação de um degrau em malha aberta, e observação da curva de resposta do processo. Esta atuação sobre a dinâmica do processo, durante a realização de uma inicialização, é efetuada através da função "**double calc_proc(double)**", a qual retorna o valor na saída do processo (Y), em função da atuação do valor passado como parâmetro, durante o tempo de uma amostragem. Salienta-se que a atuação dos conversores A/D e D/A é automática.

1.5.2.2. CONVERSORES A/D E D/A, E OUTROS RECURSOS

Cada novo controlador herda, da classe base, os seguintes atributos: **cad**, **cda**, **referencia** e **t_am**, relativos a: conversor A/D, conversor D/A, valor da referência (Yr) e valor do tempo de amostragem, respectivamente. A utilização destes atributos pode ser bastante útil na implementação de um controlador, como por exemplo a inserção de ação integrativa no controlador EXMPL.

A figura 5.4 apresenta as listagens dos arquivos "exmpl.h" e "exmpl.cc" com as alterações necessárias para prover esta ação integrativa. Observa-se na listagem de "exmpl.h" a criação de um ganho para a ação integral (**ganho_integ**), alterável via painel de inicialização. Outro destaque é a redefinição do método virtual "limpar", utilizado para resetar a dinâmica do controlador no início de cada simulação.

"exmpl.h"	"exmpl.cc"
<pre>#ifndef controlador_exmpl_DEFINED #define controlador_exmpl_DEFINED #include "controlador.h" class EXMPL : public Controlador { public:</pre>	<pre>: EXMPL::EXMPL() { ganho = 1.; ganho_integ = 1.; limpar();</pre>

```

:
void limpar();
private:
:
double ganho_integ, u_acum;
Panel_item item_ganho_integ;
};
#endif ~controlador_exmpl_DEFINED
}
:
double
EXMPL::avancar(double e)
{
double u_acum_ant = u_acum;
u_acum += ganho_integ * e * t_am;
double u = ganho * e + u_acum;
if ((u < cda.lim_inf()) || (u > cda.lim_sup()))
{ double u_aux = cda.converter(u);
u_acum = u_acum_ant + ganho_integ*e*t_am * u_aux / u;
}
return u;
}

void
EXMPL::limpar()
{
u_acum = 0.;
}

void
EXMPL::inic_interface(Frame frame)
{
:
item_ganho = (Panel_item)xv_create(panel, PANEL_TEXT,
PANEL_LABEL_STRING, "Ganho:",
PANEL_VALUE, "1",
NULL);
item_ganho_integ = (Panel_item)xv_create(panel,
PANEL_TEXT,
PANEL_LABEL_STRING, "Ganho Integral:",
PANEL_VALUE, "1",
NULL);
:
}
:
void
EXMPL::aplicar(Panel_item item)
{
:
else
este->ganho = valor;
string = (char*)xv_get(este->item_ganho_integ,
PANEL_VALUE);
endptr = &string;
valor = strtod(string, endptr);
if (**endptr != NULL)
{ /* Tratamento de erro, a critério do usuário. */
}
else
este->ganho_integ = valor;
este->fim_inic();
}
}

```

Fig. 5.4 : Controlador EXMPL com Ação Integrativa

Destaca-se, no arquivo "exmpl.cc", a utilização do valor do tempo de amostragem (t_{am}) no cálculo da integral do erro, e da estratégia de redução de sobrecarga da ação integral. Esta estratégia foi criada de forma bem simples, e nunca testada, mas atende ao propósito de apresentar alguns métodos dos conversores. Os métodos **lim_inf** e **lim_sup** retornam o valor dos limites de conversão, enquanto o método **converter** realiza uma discretização em amplitude, simulando a atuação de conversores reais. Os métodos **resol** e **defin**, não apresentados no exemplo, também podem ser utilizados, retornando os valores da resolução (número de bits) e definição (mínima variação de amplitude) do conversor, respectivamente.

Para concluir a apresentação dos recursos básicos, resta apenas o sinal de referência. Além do acesso ao atributo **referencia**, o usuário pode redefinir o método virtual **Yr**, como por exemplo para medir o tempo de subida do sinal Y, frente a uma mudança de referência. Esta medição é adicionada ao controlador EXMPL, conforme mostra a figura 5.5.

"exmpl.h"	"exmpl.cc"
<pre>#ifndef controlador_exmpl_DEFINED #define controlador_exmpl_DEFINED #include "controlador.h" class EXMPL : public Controlador { public: : void Yr(double); private: : double degrau; char flag_ts; }; #endif ~controlador_exmpl_DEFINED</pre>	<pre>#include <math.h> : double EXMPL::avancar(double e) { static double t_sub; if ((flag_ts == 1) && (fabs(e) < 0.9*degrau)) { t_sub = 0.; flag_ts = 2; } else if (flag_ts == 2) { t_sub += t_am; if (fabs(e) < 0.1*degrau) { printf("Tempo de subida = %g s\n", t_sub); flag_ts = 0; } } } : } void EXMPL::limpar() { u_acum = 0.; flag_ts = 0; } void EXMPL::Yr(double y_ref) { degrau = fabs(y_ref - referencia); flag_ts = 1; referencia = y_ref; }</pre>

Fig. 5.5 : Listagens do Controlador EXMPL Básico

1.5.3. IMPLEMENTAÇÃO DOS RECURSOS DE AVALIAÇÃO

A partir do momento em que o controlador adaptativo, em desenvolvimento, necessita ter seu desempenho analisado de uma forma quantitativa, torna-se extremamente útil a utilização dos recursos de avaliação descritos nesta seção. Estes recursos têm ligação direta com o modo de simulação automática e seus mecanismos de avaliação, como as seqüências de eventos e os relatórios.

A apresentação dos recursos de avaliação está dividida em duas seções, onde na primeira, seção 5.3.1, apresenta-se os recursos "obrigatórios" em um controlador adaptativo, e na segunda seção, 5.3.2, apresenta-se o recurso opcional de envio de mensagens aos relatórios de experimento.

1.5.3.1. MECANISMOS GERAIS

Os mecanismos classificados como "gerais" são, na realidade, quatro métodos virtuais que devem ser redefinidos na implementação de cada controlador adaptativo. O primeiro, denominado **obter_nome**, é o mais simples deles, consistindo apenas na informação do nome do controlador, através de uma seqüência de caracteres. Este é o nome do controlador que aparecerá nos relatórios de avaliação.

O segundo método, denominado **inicializar**, já foi apresentado na seção 5.2.1., mas sua utilização permanece incompleta até o presente momento. Este método possui dois parâmetros, onde o primeiro atua como um flag, sinalizando se esta inicialização deve ser realizada normalmente (via interação com o usuário através do painel) ou com valores "default". Para que estes valores "default" não fiquem completamente arbitrários, pode-se utilizar o segundo parâmetro do método "inicializar", o qual contém uma indicação grosseira do tempo de resposta do processo a um degrau em malha aberta. Como geralmente a inicialização de controladores adaptativos se dá através de mecanismos de pré-ajuste, este segundo parâmetro pode ser bastante útil na determinação da unidade de tempo esperada à uma resposta do processo.

Os outros dois métodos, denominados **limpar_sinal_ajuste** e **ajustado**, relacionam-se com o mecanismo de identificação de adaptação do controlador adaptativo, ou seja, o controlador adaptativo deve possuir um mecanismo de supervisão capaz de verificar se os parâmetros do algoritmo de controle estão adaptados à atual dinâmica do processo. O método "limpar_sinal_ajuste" é chamado para iniciar novo processo de identificação de adaptação, enquanto que "ajustado" é chamado ciclicamente até que o controlador esteja ajustado.

Novas modificações são efetuadas no controlador EXMPL, figura 5.6, de forma a exemplificar a utilização dos mecanismos gerais de avaliação. O segundo parâmetro do método "inicializar" é utilizado na determinação do tempo de subida esperado para o processo. Enquanto **t_sub** (ver figura 5.5) for maior que o tempo esperado, aumenta-se o ganho em 10 %. Estes procedimentos nada têm de científicos, servindo apenas para demonstrar a utilização dos métodos.

"exmpl.h"	"exmpl.cc"
<pre>#ifndef controlador_exmpl_DEFINED #define controlador_exmpl_DEFINED #include "controlador.h" class EXMPL : public Controlador { public: :</pre>	<pre>:: char* EXMPL::obter_nome() { return ("EXMPL"); }</pre>

```

char* obter_nome();
void limpar_sinal_ajuste();
char ajustado();
private:
:
char flag_aj;
double t_sub_esp;
};
#endif ~controlador_exmpl_DEFINED

double
EXMPL::avancar(double c)
{
:
{ printf("Tempo de subida = %g s\n", t_sub);
flag_ts = 0;
flag_aj = (t_sub < t_sub_esp);
if (!flag_aj)
ganho *= 1.1;
}
:
}
:
void
inicializar(char modo, double tempo_ma)
{ //modo == 0 => valores default.
t_sub_esp = tempo_ma / 3.;
if (modo)
xv_set(subframe_inic, XV_SHOW, TRUE, NULL);
else
{ ganho = ganho_integ = 1.;
fim_inic();
}
}
:
void
EXMPL::limpar_sinal_ajuste()
{
flag_aj = 0;
}

char
EXMPL::ajustado()
{
return flag_aj;
}

```

Fig. 5.6 : Controlador EXMPL com Mecanismos Gerais de Avaliação

1.5.3.2. MECANISMOS PARA ENVIO DE MENSAGENS

Para compreender o mecanismo de envio de mensagens é necessário, antes, conhecer a classe "mensagem". Objetos desta classe contêm um conjunto com até trinta strings (seqüências de caracteres), alocados dinamicamente na memória. Os métodos da classe "mensagem" que têm utilidade ao implementador de um controlador adaptativo são:

- **void inserir(char*);** : Com este método o usuário insere novas strings no conjunto. A seqüência de caracteres passada como parâmetro é copiada para uma nova área de memória, podendo ser destruída pelo objeto que a criou.
- **void limpar();** : Este método é utilizado para limpar o conjunto de strings, desalocando-as da memória.

A utilização do recurso de envio de mensagem aos relatórios de experimento é opcional, mas pode ser extremamente útil durante o desenvolvimento e aperfeiçoamento de um controlador adaptativo. Qualquer informação que o usuário considerar importante pode ser enviada, havendo três

oportunidades para fazê-lo: ao final de uma inicialização, ao final de uma análise (ver seção 4.4) e ao final de um loop sinalizado.

Para qualquer uma destas oportunidades há a chamada de um método no início e a chamada de outro método na conclusão. A tabela 5.1 apresenta estes métodos de início e conclusão, todos virtuais.

	Método de Início	Método de Conclusão
Inicialização	inicializar	informe inicializacao
Análise	inic analise	fim analise
Loop Sinalizado	limpar sinal ajuste	informe seq ajuste

Tab. 5.1 : Métodos do Mecanismo de Mensagens

A utilização das mensagens é exemplificada no controlador EXMPL, de uma forma extremamente simples, mas de fácil compreensão. Observa-se as listagens de "exmpl.h" e "exmpl.cc" na figura 5.7, contendo a última versão deste controlador.

"exmpl.h"	"exmpl.cc"
<pre>#ifndef controlador_exmpl_DEFINED #define controlador_exmpl_DEFINED #include "controlador.h" class EXMPL : public Controlador { public: : mensagem informe_inicializacao(); void inic_analise(); mensagem fim_analise(); mensagem informe_seq_ajuste(); private: : mensagem msg_anls; char flag_anls; char flag_aj_1; double t_sub_inic; }; #endif ~controlador_exmpl_DEFINED</pre>	<pre>: EXMPL::EXMPL() { : flag_anls = flag_aj_1 = 0; } : double EXMPL::avancar(double c) { : flag_aj = (t_sub < t_sub_esp); if (!flag_aj) ganho *= 1.1; if (flag_aj_1) { t_sub_inic = t_sub; flag_aj_1 = 0; } if (flag_anls) { char string[60]; sprintf(string, "(%g %g => %g)", ganho, ganho_integ, t_sub); msg_anls.inserir(string); } : } : void EXMPL::limpar_sinal_ajuste() { flag_aj = 0; flag_aj_1 = 1; } : mensagem EXMPL::informe_inicializacao()</pre>

```

{
    mensagem msg_inic;
    char string[100];
    sprintf(string,
        "Ganho=%g Ganho_Integ=%g T_sub_esp=%g",
        ganho, ganho_integ, t_sub_esp);
    msg_inic.inserir(string);
    return msg_inic;
}

void
EXMPL::inic_analise()
{
    msg_anls.limpar();
    flag_anls = 1;
    sprintf(string, "(%g %g => %g)", ganho, ganho_integ,
        t_sub);
    msg_anls.inserir(string);
}

mensagem
EXMPL::fim_analise()
{
    flag_anls = 0;
    return msg_anls;
}

mensagem
EXMPL::informe_seq_ajuste()
{
    mensagem msg_aj;
    char string[50];
    sprintf(string, "T_SUB_Inic = %g ; T_SUB_Final = %g",
        t_sub_inic, t_sub);
    msg_aj.inserir(string);
    return msg_aj;
}

```

Fig. 5.7 : Versão Final do Controlador EXMPL

A mensagem a respeito da inicialização contém os valores do ganho proporcional, do ganho da ação integrativa e do valor esperado para o tempo de subida. Em uma análise informa-se os valores dos ganhos e do tempo de subida correspondente, tanto iniciais quanto para cada mudança de referência. Finalmente, em seqüências de ajuste (loops sinalizados) informa-se os valores inicial e final para o tempo de subida.

ANEXO II : ALTERAÇÃO NA ORDEM DO MODELO DOS BLOCOS DLC_FT

Conforme mencionado na seção 4.4.1, uma função de transferência descrita por:

$$G(s) = \frac{B[s]}{A[s]} + d = \frac{b_{n-1}s^{n-1} + b_{n-2}s^{n-2} + \dots + b_1s + b_0}{a_n s^n + a_{n-1}s^{n-1} + \dots + a_1s + a_0} + d \quad (\text{II.1})$$

onde $a_n \neq 0$; pode ser representada através das seguintes equações de estado:

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & 1 \\ -\frac{a_0}{a_n} & -\frac{a_1}{a_n} & -\frac{a_2}{a_n} & \dots & -\frac{a_{n-2}}{a_n} & -\frac{a_{n-1}}{a_n} \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ \frac{1}{a_n} \end{bmatrix} u$$

$$y = \begin{bmatrix} b_0 & b_1 & b_2 & \dots & b_{n-2} & b_{n-1} \end{bmatrix} x + \begin{bmatrix} d \end{bmatrix} u$$

Esta representação, bastante parecida com a "forma canônica controlável", permite que seja facilmente alterada a ordem do modelo durante a simulação. Inicialmente será analisado o caso de redução na ordem do modelo e posteriormente o caso de aumento.

A alteração na função de transferência apresentada em II.1, para uma nova função, bastante distinta, mas de menor ordem:

$$G'(s) = \frac{B'[s]}{A'[s]} + d' = \frac{b'_{r-1}s^{r-1} + b'_{r-2}s^{r-2} + \dots + b'_1s + b'_0}{a'_r s^r + a'_{r-1}s^{r-1} + \dots + a'_1s + a'_0} + d' \quad (\text{II.2})$$

onde $a'_r \neq 0$ e $n > r$; pode ser realizada por partes, na representação por equações de estado escolhida. O primeiro passo consiste na alteração dos termos do numerador da função de transferência inicial, mantendo-se o denominador constante:

$$\hat{G}(s) = \frac{B'[s] + d' \cdot A'[s]}{A[s]} = \frac{\hat{B}[s]}{A[s]} \quad (\text{II.3})$$

Esta alteração afeta apenas as matrizes C e D, tornando as equações de estado da seguinte forma:

$$\dot{x} = Ax + Bu$$

$$\hat{y} = [\hat{b}_0 \quad \hat{b}_1 \quad \dots \quad \hat{b}_{n-2} \quad \hat{b}_{n-1}]x + [0]u$$

Em seguida reduz-se a ordem do sistema, em uma unidade, fazendo-se $a_n=0$. Para verificar o comportamento das equações de estado, com esta alteração, reescreve-se estas sem a utilização de matrizes:

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = x_3$$

$$\vdots$$

$$\dot{x}_{n-1} = x_n$$

(II.4)

$$\dot{x}_n = -\frac{a_0}{a_n}x_1 - \frac{a_1}{a_n}x_2 - \dots - \frac{a_{n-1}}{a_n}x_n + \frac{1}{a_n}u$$

(II.5)

A equação II.5 pode ser reescrita da seguinte forma:

$$a_n \dot{x}_n = -a_0 x_1 - a_1 x_2 - \dots - a_{n-1} x_n + u$$

onde pretende-se fazer $a_n=0$. Esta alteração em a_n não afetará a igualdade acima se o valor de \dot{x}_n for nulo, sendo esta uma condição necessária para que se possa realizar a redução na ordem do sistema. Com isto chega-se a:

$$x_n = -\frac{a_0}{a_{n-1}}x_1 - \frac{a_1}{a_{n-1}}x_2 - \dots - \frac{a_{n-2}}{a_{n-1}}x_{n-1} + \frac{1}{a_{n-1}}u$$

que substituída em II.4 resulta na seguinte equação de estados:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 \\ \vdots \\ \dot{x}_{n-2} = x_{n-1} \\ \dot{x}_{n-1} = -\frac{a_0}{a_{n-1}}x_1 - \frac{a_1}{a_{n-1}}x_2 - \dots - \frac{a_{n-2}}{a_{n-1}}x_{n-1} + \frac{1}{a_{n-1}}u \end{cases}$$

Para reduzir a ordem do sistema em mais uma unidade, repete-se este procedimento fazendo $a_{n-1}=0$, e assim por diante, até $a_{r+1}=0$. Nesse momento a ordem do sistema passará a estar com a dimensão desejada, ou seja, de ordem "r". Observa-se que os estados x_1 até x_r permanecem com os mesmos valores, e os demais deixam de existir.

O próximo passo consiste na substituição dos valores de a_i pelos a'_i correspondentes, resultando no seguinte conjunto de equações:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 \\ \vdots \\ \dot{x}_{r-1} = x_r \\ \dot{x}_r = -\frac{a'_0}{a'_r}x_1 - \frac{a'_1}{a'_r}x_2 - \dots - \frac{a'_{r-1}}{a'_r}x_r + \frac{1}{a'_r}u \\ y' = \hat{b}_0x_1 + \hat{b}_1x_2 + \dots + \hat{b}_{r-1}x_r + \hat{b}_rx_{r+1} \end{cases}$$

onde $x_{r+1} = \dot{x}_r$, o que resulta em:

$$y' = \hat{b}_0x_1 + \hat{b}_1x_2 + \dots + \hat{b}_{r-1}x_r + d' \cdot (-a'_0x_1 - a'_1x_2 - \dots - a'_{r-1}x_r + u)$$

onde $d' = \frac{\hat{b}_r}{a'_r}$. Relembrando-se a equação II.3, na qual $\hat{B} = B' + d' \cdot A'$, conclui-se que:

$$y' = b'_0x_1 + b'_1x_2 + \dots + b'_{r-1}x_r + d' \cdot u$$

Estas equações contêm a mesma representação de estados que teríamos obtido diretamente, partindo-se da equação II.2. Portanto pode-se concluir que em alterações na função de transferência que impliquem na redução da ordem do sistema, basta reescrever as equações de estado na forma padrão e manter os estados com os valores vigentes, caso a derivada dos estados eliminados sejam nulas, o que é garantido se o sistema estiver em regime permanente.

Quando uma alteração na função de transferência implicar no aumento da ordem do sistema, adota-se o procedimento inverso ao apresentado na redução de ordem, para demonstrar que basta reescrever as equações de estado na forma padrão e criar novos estados. Adota-se, no bloco DLC_FT, iniciar estes novos estados com valor nulo.

ANEXO III : CONJUNTO DE PROCESSOS UTILIZADOS NAS ANÁLISES DO CONTROLADOR PID-AA

Processo 1: **Proc1**

$$G(s) = \frac{1e^{-Ls}}{(1+2s)}$$

$L_{\text{mín}} = 0,0$
$L_{\text{méd}} = 2,0$
$L_{\text{máx}} = 4,0$

Processo 2: **Proc2**

$$G(s) = \frac{1e^{-Ls}}{(1+10s)(1+s)}$$

$L_{\text{mín}} = 0,0$
$L_{\text{méd}} = 12,9$
$L_{\text{máx}} = 26,2$

Processo 3: **Proc3**

$$G(s) = \frac{1e^{-Ls}}{(1+3s)(1+s)}$$

$L_{\text{mín}} = 0,0$
$L_{\text{méd}} = 4,7$
$L_{\text{máx}} = 9,8$

Processo 4: **Proc4**

$$G(s) = \frac{1e^{-Ls}}{(1+2s)^2}$$

$L_{\text{mín}} = 0,0$
$L_{\text{méd}} = 4,8$
$L_{\text{máx}} = 10,1$

Processo 5: **Proc5**

$$G(s) = \frac{1e^{-Ls}}{(1+10s)(1+2s)(1+s)}$$

$L_{\text{mín}} = 0,0$
$L_{\text{méd}} = 13,5$
$L_{\text{máx}} = 29,0$

Processo 6: **Proc6**

$$G(s) = \frac{1e^{-Ls}}{(1+8s)^2(1+0,8s)}$$

$L_{\text{mín}} = 0,0$
$L_{\text{méd}} = 18,4$
$L_{\text{máx}} = 40,0$

Processo 7: **Proc7**

$$G(s) = \frac{1e^{-Ls}}{(1+s)^3}$$

$L_{\text{mín}} = 0,00$
$L_{\text{méd}} = 2,84$
$L_{\text{máx}} = 6,50$

Processo 8: **Proc8**

$$G(s) = \frac{1e^{-Ls}}{(1+2s)^4}$$

$L_{\text{mín}} = 0,0$
$L_{\text{méd}} = 6,1$
$L_{\text{máx}} = 15,0$

Processo 9: **Proc9**

$$G(s) = \frac{1e^{-Ls}}{(1+s)^6}$$

$L_{\text{mín}} = 0,0$
$L_{\text{méd}} = 3,0$
$L_{\text{máx}} = 8,8$

Processo 10: **DNM**

$$G(s) = \frac{1e^{-Ls}}{(1+15s)(1+0,75s)^5}$$

$L_{\text{mín}} = 0,0$
$L_{\text{méd}} = 17,8$
$L_{\text{máx}} = 38,8$

Processo 11: **FNM**

$$G(s) = \frac{(1-0,5s)e^{-Ls}}{(1+s)^3}$$

$L_{\text{mín}} = 0,00$
$L_{\text{méd}} = 2,13$
$L_{\text{máx}} = 5,62$

Processo 12: **ASTROM**

$$G(s) = \frac{(1-\tau s)}{(1+15s)(1+0,75s)(1+\tau s)}$$

$\tau_{\text{mín}} = 0,00$
$\tau_{\text{méd}} = 0,45$
$\tau_{\text{máx}} = 0,75$

Processo 13: **COMPLEX**

$$G(s) = \frac{1e^{-Ls}}{(1+s+0,25s^2)(1+2s)(1+s)(1+1,2s)}$$

$L_{\text{mín}} = 0,00$
$L_{\text{méd}} = 3,63$
$L_{\text{máx}} = 9,18$

Processo 14: **CLAA**

$$G(s) = \frac{4(1+0,5s)e^{-Ls}}{(1+2,5s)(1+5s)(1+10s)(1+0,25s)^2}$$

$L_{\text{mín}} = 0,0$
$L_{\text{méd}} = 17,2$
$L_{\text{máx}} = 38,2$

Processo 15: **ZERO1**

$$G(s) = \frac{0,8(1+1,25s)e^{-Ls}}{(1+s)(1+2s)}$$

$L_{\text{mín}} = 0,00$
$L_{\text{méd}} = 1,78$
$L_{\text{máx}} = 3,64$

Processo 16: **ZERO2**

$$G(s) = \frac{2(1+7,4s)(1+6,25s)e^{-Ls}}{(1+10s)(1+6,99s)(1+3,03s)}$$

$L_{\text{mín}} = 0,0$
$L_{\text{méd}} = 6,7$
$L_{\text{máx}} = 13,5$

Processo 17: **ZERO3**

$$G(s) = \frac{2,44(1+0,98s+0,41s^2)e^{-Ls}}{(1+0,5s+0,125s^2)(1+s)(1+0,5s)(1+0,66s)}$$

$L_{\text{mín}} = 0,00$
$L_{\text{méd}} = 1,65$
$L_{\text{máx}} = 3,26$

Processo 18: **FNM-ZERO**

$$G(s) = \frac{(1-s)(1+2s)e^{-Ls}}{(1+5s)(1+s)^2}$$

$L_{\text{mín}} = 0,00$
$L_{\text{méd}} = 3,03$
$L_{\text{máx}} = 7,51$

A realização dos experimentos de análise dar-se-á com os seguintes valores para a variável auxiliar, associada ao parâmetro do processo variante no tempo: -100; -90; -80; -60; -30; 0; 50 e 100.

BIBLIOGRAFIA

- [Abreu 92] A. H. Abreu, "Implementação de um Controlador PID Auto-Ajustável Baseado em Reconhecimento de Formas de Ondas", Dissertação de Mestrado, LCM I EEL UFSC, Agosto 1992.
- [Åström 84] K. J. Åström e T. Hägglund, "Automatic Tuning of Simple Regulators with Specifications on Phase and Amplitude Margins", *Automatica*, Vol. 20, No. 5, pp. 645-651, 1984.
- [Åström 89] K. J. Åström e B. Wittenmark, "Adaptive Control", Addison Wesley, 1989.
- [Bristol 83] E. H. Bristol, "The Design of Industrial Useful Adaptive Controllers", *ISA Transactions*, Vol. 22, No. 3, pp. 17-25, 1983.
- [Bruciapaglia 92] A. H. Bruciapaglia, "Sobre o Desenvolvimento do Controlador PID-AA / UFSC Nova Proposta para Processos com Atraso de Transporte Dominante", Monografia Apresentada em Concurso Público para Professor Titular da UFSC, Novembro 1992.
- [Caetano 89] S. S. Caetano e D. J. Pagano, "Documentação do SDPID (Versão 1.0) : Manual do Usuário", Nota técnica LCM I 89/10, Novembro 1989.
- [Caetano 92] S. S. Caetano, "Controlador PID-AA : Novos Desenvolvimentos", Dissertação de Mestrado, LCM I EEL UFSC, Outubro 1992.
- [Clarke 85] D. W. Clarke, P. S. Tuffs e C. Mohtadi, "Self-tuning Control of a Difficult Process", *Identification and System Parameter Estimation*, pp. 1005-1014, 1985.
- [Da Luz 90] L. T. O. da Luz, "Sistema de Identificação, Modelagem, Análise e Projeto de Sistemas de Controle para Sistemas Elétricos de Potência", Dissertação de Mestrado, LCM I EEL UFSC, Novembro 1990.
- [De Keyser 88] R. M. C. De Keyser, P. G. A. Van de Velde e F. A. G. Dumortier, "A Comparative Study of Self-adaptive Long-range Predictive Control Methods", *Automatica*, Vol. 24, No. 2, pp. 149-163, 1988.

- [Dumont 89] G. A. Dumont, J. M. Martin-Sanchez e C. C. Zervos, "Comparison of an Auto-tuned PID Regulator and an Adaptive Predictive Control System on an Industrial Bleach Plant", *Automatica*, Vol. 25, No. 1, pp. 33-40, 1989.
- [Eckel 91] B. Eckel, "C++ : guia do usuário", Makron, McGraw-Hill, 1991.
- [Ferreira 75] A. B. de H. Ferreira, "Novo Dicionário da Língua Portuguesa", Editora Nova Fronteira, 1975.
- [Gawthrop 90] P. J. Gawthrop, P. E. Nomikos e L. S. P. S. Smith, "Adaptive Temperature Control of Industrial Processes : a Comparative Study", *IEE Proceedings*, Vol. 137, Pt. D, No. 3, May 1990.
- [Gomide 91] F. Gomide, W. Amaral e A. Coelho, "Adaptive Intelligent Control in Distillation Columns", *Latin American Conference on Artificial Intelligence in Petroleum Exploration and Production*: 226-237, 1991.
- [Hägglund 91] T. Hägglund e K. J. Åström, "Industrial Adaptive Controllers Based on Frequency Response Techniques", *Automatica*, Vol. 27, No. 4, pp. 599-609, 1991.
- [Hang 91] C. C. Hang, K. K. Sin e T. T. Tay, "Portable Training Simulator for the Teaching of Adaptive Control", *Preprints of IFAC Conference on Advances in Control Education*, Boston, pp. 224-229, June 1991.
- [Heller 90] D. Heller, "XView Programming Manual", O'Reilly & Associates, Inc., 1990.
- [Hsu 92] Liu Hsu, Reunião Técnica, LCMI, Fevereiro 1992.
- [Ioannou 86] P. A. Ioannou e K. S. Tsakalis, "A Robust Direct Adaptive Controller", *IEEE Transactions on Automatic Control*, Vol. AC-31, No. 11, pp. 1033-1043, 1986.
- [Kammer 92a] L. C. Kammer, "Sistema para Avaliação de Desempenho de Controladores Adaptativos (SADECA)", Vol. I : Manual do Usuário, Nota interna LCMI, Julho 1992.
- [Kammer 92b] L. C. Kammer, D. J. Pagano, A. H. Bruciapaglia, "Avaliação de Desempenho de Controladores Adaptativos : um Ambiente de Simulação", 9º Congresso Brasileiro de Automática: 360-365, Vitória ES, Setembro 1992.

- [Kraus 84] T. W. Kraus e T. J. Myron, "Self-tuning PID Controller Uses Pattern Recognition Approach", *Control Engineer*, pp. 106-111, June 1984.
- [Landau 79] Y. D. Landau, "Adaptive Control - The Model Reference Approach", Marcell Dekker, 1979.
- [Lima 91] S. C. Lima, "Identificação de Sistemas Assistido por Computador", Dissertação de Mestrado, LCM I EEL UFSC, Dezembro 1991.
- [Litt 91] J. Litt, "An Expert System to Perform On-Line Controller Tuning", *IEEE Control Systems*, pp. 18-23, April 1991.
- [Meyer 88] B. Meyer, "Object-Oriented Software Construction", Prentice Hall, 1988.
- [Ogata 70] K. Ogata, "Modern Control Engineering", Prentice Hall, 1970.
- [Pagano 89] D. J. Pagano, "Desenvolvimento de um Controlador Digital PID Auto-Ajustável Baseado num Sistema Especialista", Dissertação de Mestrado, LCM I EEL UFSC, Março 1989.
- [Pagano 91] D. J. Pagano, A. H. Bruciapaglia, S. S. Caetano, L. C. Kammer, "Intelligent Tuning of PID Controllers Based on a Production Rules System", *Intelligent Tuning and Adaptive Control - ITAC'91, IFAC Symposium, Preprints, Singapore, 1991*.
- [Porter 87] B. Porter, A. M. Jones e C. B. McKeown, "Real-time Expert Tuners for PI Controllers", *IEE Proceedings*, Vol. 134, Pt. D, No. 4, July 1987.
- [Ruano 92] A. E. B. Ruano, P. J. Fleming, D. I. Jones, "Connectionist Approach to PID Autotuning", *IEE Proceedings-D*, Vol. 139, No. 3, May 1992.
- [Savi 87] V. M. Savi, "Projeto Assistido por Computador para Sistemas de Controle : Especificação e Projeto de um Pacote", Dissertação de Mestrado, LCM I EEL UFSC, Maio 1987.
- [De Souza 92] G. F. de Souza, E. Cantú, "Descrição do Sistema SP0, um Sistema de Produção de Ordem Zero", *Publicação Interna LCM I PI 92-8*, Agosto 1992.

- [Stroustrup 86] B. Stroustrup, "The C++ Programming Language", Addison-Wesley, Reading MA, 1986.
- [Sun 89] Sun Microsystems, Inc.; "OPEN LOOK Graphical User Interface Application Style Guidelines", Addison-Wesley, 1989.
- [Takahashi 88] T. Takahashi, "Introdução a Programação Orientada a Objetos", III EBAI, Curitiba PR, Janeiro 1988.
- [Versteeg 86] H. J. Versteeg, H. J. Jansma e K. Turner, "Evaluation of Commercially Available Adaptive Controllers", Journal A, Vol. 27, No. 3, pp. 120-126, 1986.
- [Wellstead 86] P. E. Wellstead, "Introduction to Self-tuning Systems", notas de aula, Control Systems Centre, UMIST, November 1986.
- [Ziegler 42] J. G. Ziegler e B. N. Nichols, "Optimum Settings for Automatic Controllers", Transactions of the ASME, Vol. 64, No. 8, pp. 759-768, November 1942.