

# UM CONTROLADOR DIGITAL INCORPORANDO TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL

Luís Fernando Alves Pereira  
Takashi Yoneyama  
Eleri Cardozo  
Divisão de Engenharia Eletrônica  
Instituto Tecnológico de Aeronáutica  
CTA - ITA - IEEE  
12.225 - São José dos Campos

**Resumo** – Este trabalho tem como objetivo apresentar uma aplicação prática de técnicas de Inteligência Artificial (Nilsson, 1980; Winston, 1984) no controle de sistemas. O algoritmo desenvolvido permite que os parâmetros do controlador sejam alterados dinamicamente em função das variáveis a serem controladas, utilizando para isto um conjunto de regras do tipo SE(condições) ENTÃO(ações). O conjunto de regras, é escrito em OPS5 (Velasco, 1987; Browston et alii, 1985) e descreve uma série de heurísticas extraídas de experimentos com o sistema a ser controlado. Através de leituras dos conversores análogo-digítals são obtidas informações referentes ao comportamento do sistema que está sendo controlado e a partir daí começa o ciclo de execução do OPS5. As regras executadas modificam de forma adequada os parâmetros do controlador. Devido à grande disponibilidade de computadores do tipo IBM-PC compatível, o desenvolvimento foi baseado em uma máquina deste tipo, apresentando um bom desempenho nas simulações de controle de plantas realísticas em que foi testado.

**Abstract** – This work concerns a practical presentation of Artificial Intelligence (Nilsson, 1980; Winston, 1984) in the control of dynamical systems. The developed algorithm has the capability to change the parameters of the controller, using a production system based on a set of rules with the following structure: IF (conditions) THEN (actions). The set of rules was written using the OPS5 (Velasco, 1987; Browston et alii, 1985) and describes a series of heuristics about the system that will be controlled. Informations about the performance of the system under control is acquired by analog-digital converters and starts the execution cycle of the OPS5. The scheme was developed using an IBM-PC compatible computer, and good performance was achieved with realistic test systems.

## 1. INTRODUÇÃO

Constata-se, através de observações práticas, que na maioria dos processos industriais, embora inexistam modelos matemáticos que reproduzam fielmente o comportamento dos seus

sistemas, operadores experientes conseguem obter um desempenho satisfatório empregando conhecimentos heurísticos.

Desta forma, desenvolveu-se um controlador digital onde o conjunto de heurísticas, na forma de regras de produção, é incorporado ao Sistema Especialista que gera os sinais de controle em tempo real. Controladores deste tipo tem sido desenvolvidos por alguns autores (Silva et alii, 1987; Pagano et alii, 1988; Kraus e Myron, 1984).

Inicialmente é efetuada a aquisição do sinal de entrada pelo microcomputador através da leitura do conversor A/D, ligado a saída da planta "y". A partir disto, o sinal de erro dado por  $e = y_{ref} - y$  é calculado. A análise de "e" juntamente com outras variáveis medidas possibilitam a seleção adequada do conjunto de regras ativas, ou seja aquelas que possuem todas as condições satisfeitas naquele ciclo de controle.

As regras ativas, quando executadas, geram o sinal de controle "u", sendo este o responsável pelo desempenho desejado da planta a ser controlada. A seguir, o sinal de controle é aplicado na entrada da planta utilizando o conversor D/A e a partir daí um novo ciclo de controle é inicializado.

As rotinas de entrada e saída bem como as rotinas numéricas utilizadas para a geração do sinal de controle foram completamente desenvolvidas em linguagem C e "linkadas" com as regras heurísticas do controlador feitas em OPS5.

## 2. A LINGUAGEM OPS5

O Controlador Inteligente proposto utiliza técnicas de Inteligência Artificial, principalmente Sistemas Especialistas. A linguagem utilizada no desenvolvimento deste controlador

foi o OPSS, que apresenta a estrutura indicada na figura abaixo:

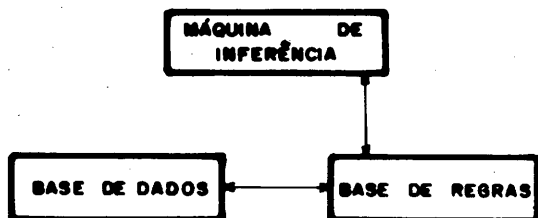


Fig. 2.1 – Estrutura do Sistema Especialista responsável pelo ajuste do algoritmo de controle.

Como pode ser observado na figura 2.1 o Sistema Especialista divide-se em três módulos:

- a) base de dados
- b) base de regras
- c) máquina de inferência

A base de dados contém informações atualizadas do sistema como um todo, processo e controlador. A cada ciclo o banco de dados é consultado pelo Sistema Especialista e um novo conjunto de regras é executado.

O banco de conhecimentos é codificado sob forma de regras de produção do tipo

SE (condições) ENTÃO (ações)

Tipicamente, uma condição é referente a um particular estado do banco de dados e uma ação modifica o presente estado.

A máquina de inferência controla a execução de regras e opera conforme o seguinte ciclo:

**Fase1:** Seleção do conjunto de conflito: Determinação do conjunto de regras que possuem todas as suas condições satisfeitas.

**Fase2:** Resolução do conflito: Escolha de uma das regras pertencentes ao conjunto de conflito, usando os seguintes critérios:

- 1) antiguidade dos dados existentes em sua memória de trabalho - é dada preferência aos dados mais recentes.
- 2) especificidade das produções - produções com maior número de condições são executadas primeiramente.

**Fase3:** Ação: Execução das ações especificadas pela regra que foi selecionada na fase anterior.

**Fase4:** Retornar a fase 1.

A linguagem OPS5 foi escolhida por apresentar uma série de facilidades que auxiliam o usuário no projeto de “controladores inteligentes” (Pereira, 1989), como por exemplo:

- (i). Inserção de pontos de quebra que podem interromper a execução do programa quando uma determinada regra é habilitada;
- (ii). Possibilidade de examinar a qualquer instante o conjunto de conflito, “CS”;
- (iii). Simplicidade com que esta linguagem pode ser integrada com a linguagem C utilizada no processamento de rotinas numéricas.

### 3. ESTRUTURA DO CONTROLADOR IMPLEMENTADO

A estrutura adotada para o “controlador inteligente” encontra-se ilustrada na figura 3.1. O Sistema Especialista previamente descrito é composto pela máquina de inferência, banco de dados e uma sintaxe para as regras que irão compor o banco de conhecimentos.

Os módulos de entrada e saída são constituídos de interfaces dotadas de conversores análogo-digitais e digitais-análogos. Estas interfaces são ativadas via máquina de inferência por ações e regras. O módulo de entrada aquisição na grandezas de saída do processo e o módulo de saída produz o sinal de atuação sobre o processo.

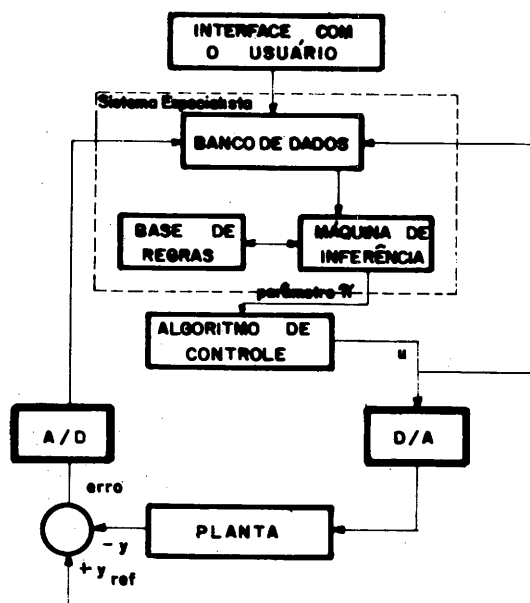


Fig. 3.1 – Arquitetura do “controlador inteligente” implementado.

O módulo controlador é composto de rotinas numéricas que computam a função de controle previamente determinada. O algoritmo de controle pode ser, por exemplo, um PID cujos parâmetros são os elementos de um vetor  $\pi = [K_p, K_i, K_d]$ , gerado anteriormente pela base de regras, que representam os ganhos proporcional, integral e derivativo, respectivamente. Outra forma seria controlar o sistema utilizando técnicas de realimentação de estados, desde que estes sejam acessíveis e suas medidas confiáveis.

### 4. EXEMPLOS DE APLICAÇÃO

Como ilustração do desempenho deste controlador, vamos supor uma estrutura de controle do tipo PID, conforme o

diagrama em blocos dado abaixo:

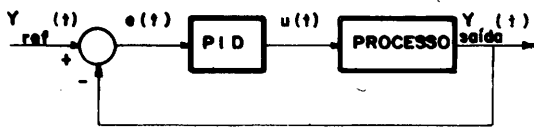


Fig. 4.1 – Diagrama de blocos de um sistema que utiliza um controlador do tipo PID.

onde:

PROCESSO: suposto ter dinâmica desconhecida.

PID: Controlador Proporcional + Integral + Derivativo.

$y_{ref}(t)$ : Sinal de Referência.

$y_{out}(t)$ : Sinal de Saída do Processo.

$e(t)$ : Sinal de erro.

$u(t)$ : Variável de Controle.

Uma metodologia bastante utilizada no ajuste dos parâmetros de controladores do tipo PID foi desenvolvida por Ziegler-Nichols (Ziegler-Nichols, 1942). Neste caso, utiliza-se esta técnica apenas na fase inicial de sintonia dos parâmetros do controlador, sendo em seguida realizado um “ajuste fino”, baseado no seguinte procedimento (Pereira et alii, 1989):

- (i). Os parâmetros obtidos pelas regras de Ziegler-Nichols são passados ao controlador, vetor  $\Pi$  (fig. 3.1), sendo então analisada a resposta ao degrau do sistema.
- (ii). Os três primeiros picos da variável de saída são detectados (fig. 4.2), e os seguintes parâmetros são calculados:
  - Sobrepasso máximo da variável de saída.
  - Subpasso máximo da variável de saída.
  - Período de oscilação amortecida (eq. 4.1).
  - Razão entre os dois primeiros picos (eq. 4.2).
  - Índice de Convergência (eq. 4.3).

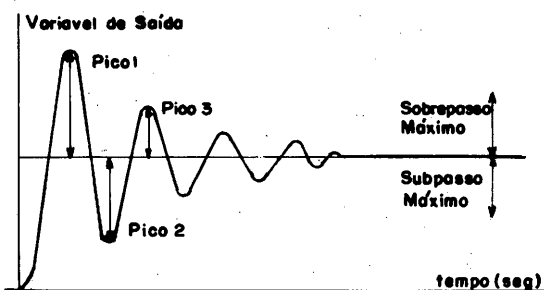


Fig. 4.2 – Resposta ao degrau de sistema que satisfaz as especificações de controle.

$$T_o = 2 * (tpico 2 - tpico 1) \quad 4.1$$

onde:

tpico 1 = instante de tempo em que ocorreu o pico 1.

tpico 2 = instante de tempo em que ocorreu o pico 2.

$$r_{12} = \frac{|\text{pico 2}|}{|\text{pico 1}|} \quad 4.2$$

$$c_{nv} = \frac{|\text{pico 2}| + |\text{pico 3}|}{|\text{pico 1}| + |\text{pico 2}|} \quad 4.3$$

- (iii). Verifica-se se a variável de saída do sistema,  $y_{saida}(t)$  obedece as especificações de desempenho desejadas pelo usuário. Caso estas especificações tenham sido atendidas o passo (viii) é executado.
- (iv). A fase do controlador é calculada (eq. 4.4), e as constante de tempo integral  $T_i$  e a constante de tempo derivativo  $T_d$  são ajustadas de forma a adequá-la para o processo em questão (Shinsky, 1979).

$$\phi_{PID} = \arctan \left[ \frac{2 * \Pi * T_d}{T_o} - \frac{T_o}{2 * \Pi * T_i} \right] \quad 4.4$$

- (v). Multiplica o ganho proporcional,  $K_p$ , pela razão entre o sobrepasso desejado e sobrepasso medido.
- (vi). Calcula-se então os ganhos integral,  $K_i$  e derivativo  $K_d$ . Os novos ganhos já calculados são inseridos no algoritmo do controlador.
- (vii). Retornar ao passo (ii).
- (viii). Manter os parâmetros do vetor  $\Pi$  constantes, concluindo a etapa de sintonia.

Neste exemplo será mostrado o desempenho do controlador proposto atuando junto a um processo representado pela seguinte função de transferência:

$$G(s) = \frac{10 e^{-0.5s}}{s + 3} \quad 4.5$$

As especificações de controle colocadas sobre o sistema são as seguintes:

Sobrepasso máximo = 15%.

Subpasso máximo = 7.5%.

Estas especificações garantirão que a razão de decaimento na amplitude do sinal de saída (eq. 4.2), seja de aproximadamente 1/4. As figuras 4.3, 4.4, e 4.5 mostram o ciclo total de ajustes necessários para que o sistema atendesse as especificações desejadas.

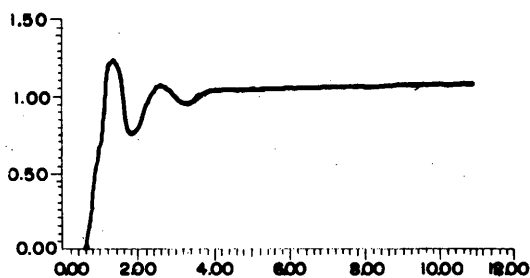


Fig. 4.3 — Comportamento do sistema no primeiro ciclo de ajuste dos parâmetros do controlador.

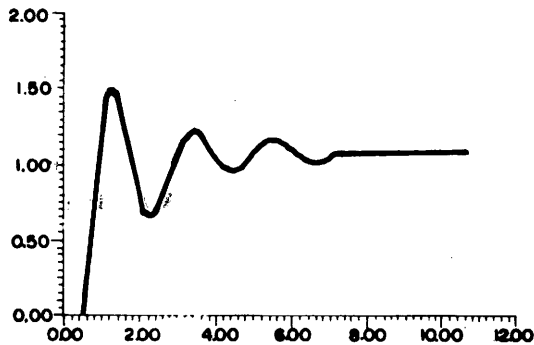


Fig. 4.4 — Comportamento do sistema para o segundo ciclo de ajuste de parâmetros do controlador.

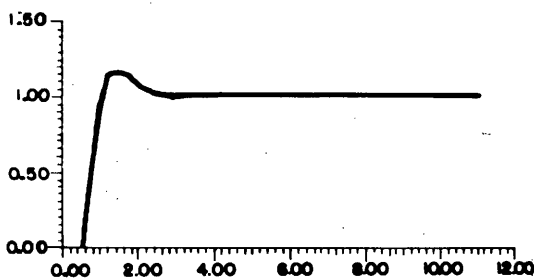


Fig. 4.5 — Comportamento do sistema para o terceiro ciclo de ajuste dos parâmetros do controlador.

A tabela 5.1 especifica os parâmetros do controlador e as características de desempenho do sistema, obtidos em cada um dos ciclos de ajuste anteriormente mostrados.

Ciclo de Ajuste	Parâmetros do Controlador			Desempenho do Sistema	
	Kp	Ki	Kd	Sobrepasso Máximo	Subpasso Máximo
1º	0.268	0.417	0.051	23.52%	17.60%
2º	0.268	0.625	0.052	55.27%	30.70%
3º	0.147	0.342	0.029	6.25%	0.84%

Tabela 5.1 — Parâmetros do controlador e desempenho do sistema (exemplo 5.2), em cada ciclo de sintonia.

Pode-se verificar, através da tabela 5.1, que ao final do terceiro ciclo de ajustes o sistema comporta-se conforme o desejado, satisfazendo as especificações de desempenho.

O exemplo acima desenvolvido continha um total de 14 regras de produção. A título de ilustração da sintaxe utilizada, são listadas a seguir algumas destas regras:

```

(p ajusta-picos
  (Desvio - deltasob { <dsob> > 0 } - deltasub { <dsub> > 0 })
  (Desempenho - razao { <r12> <1.0 }
    - convergencia { <cnv> <1.0 }
    - variacao { > -0.1 <0.1 })

  (Ganhos - proporcional <kp>
    - tintegral <tint>
    - tdiferencial <tdif> )

-->

(remove 1)
(bind <pcsob> (compute <dsob> // 100 ))
(bind <nkp> (compute <kp> - ( <kp> * <pcsob> )))
(bind <nki> (compute <nkp> // <tint> ))
(bind <nkd> (compute <nkp> * <tdif> ))
(call passa -ganhos <nkp> <nki> <nkd> ))

(p diminui-derivativo
  (Desvio - deltasob { <dsob> > 0 } - deltasub { <dsub> <0 } })
  (Desempenho - razao { <r12> <1.0 }
    - convergencia { <cnv> <1.0 }
    - variacao { > -0.1 <0.1 })

  (Ganhos - proporcional <kp>
    - tintegral <tint>
    - tdiferencial <tdif> )

-->

(remove 1)
(bind <pcsob> (compute (<dsob> - <dsub>) // (100 * <cnv> )))
(bind <nkp> (compute <kp> - ( <pcsob> * <kp> )))
(bind <nki> (compute <nkp> // <tint> ))
(bind <nkd> (compute <nkp> * <tdif> ))
(call passa -ganhos <nkp> <nki> <nkd> ))

```

## 5. CONCLUSÃO

A implementação deste tipo de controlador apresenta vários aspectos vantajosos, principalmente no que tange a Automação e Controle de Processos Industriais, onde modelos matemáticos precisos são raramente disponíveis, impedindo a aplicação direta de métodos de síntese baseados em Teoria de Controle.

Desde que existam heurísticas adequadas, o controlador proposto pode ser aplicado a uma grande classe de processos industriais, incluindo as que têm dinâmicas mal conhecidas.

Para aplicação deste tipo de controlador em tempo real deve-se levar em consideração o tempo necessário para o processamento das regras, que depende basicamente de dois fatores:

- (i). Número total de regras que compõe o banco de conhecimentos do "controlador inteligente";
- (ii). Número médio de regras que são acionadas a cada instância, compondo o conjunto de conflito.

Estes dois itens podem vir a limitar as aplicações deste controlador, portanto, o usuário deve ter o cuidado de eliminar regras desnecessárias, e se possível, reduzir o número de conflitos existentes em cada instância.

Um outro cuidado que o usuário deve ter na fase de elaboração do banco de conhecimentos é o da possibilidade de omitir regras. Caso isto ocorra, a confiabilidade no controlador ficará comprometida.

A facilidade com que o usuário pode interagir com o controlador, é um fator relevante não apenas na etapa de desenvolvimento, relativa aos ajustes de parâmetros e escolha do algoritmo do controlador, como também na fase de operação, onde algoritmo de supervisão, controle de produção e controle de qualidade, por exemplo, poderão eventualmente ser incorporados.

O uso do computador tipo IBM-PC compatível faz com que este controlador seja bem disseminável, em função da disponibilidade de "hardware" e portabilidade do "software".

## 6. AGRADECIMENTOS

Os autores desejam manifestar os seus agradecimentos à FAPESP, a qual financiou parte deste projeto através do convênio 87/1866-5, e ao ITA pela utilização dos recursos do Laboratório de Controle por Computador. O Dr. Flávio Roberto Dias Velasco, do INPE, que gentilmente cedeu o código do OPSS por ele desenvolvido.

Os autores estão ainda em dívida com diversos colegas pelas valiosas discussões técnicas e aos revisores pelas sugestões recebidas.

- NILSSON, N.J. (1980) "Principles of Artificial Intelligence". Tioga Press, Paló Alto, California.
- WINSTON, P.H. (1984) "Artificial Intelligence". Addison-Wesley, Reading, MA.
- VELASCO, F.R.D (1987) "Uma Implementação da Linguagem OPSS para Computadores Compatíveis com o IBM-PC". Proc. Quarto Simpósio Brasileiro de Inteligência Artificial, Uberlândia, 257-265, outubro.
- BROWSTON, L., FARRELL, R., KANT, E. & MARTIN, N. (1985) "Programming Expert Systems in OPSS: An Introduction to Rule-Based Programming". Addison-Wesley.
- SILVA, M.A., GOMIDE, F.A.C & AMARAL, W.C. (1987) "Um Sistema Especialista para Auto-Sintonia de Controladores Industriais do Tipo PID", 4º Simpósio Brasileiro de Inteligência Artificial, Uberlândia, outubro.
- PAGANO, D.J., BRUCIAPAGLIA, A.H., GARNOUSSET, H.E., NETO, T.A., CAETANO, S.S. (1988) "Sistema Especialista para Ajuste de Controladores PID", 3º Congresso Latino Americano de Automática, Volume 1, pp. 80-85, novembro.
- KRAUS, T.W., MYRON, T.J. (1984) "Self-Tuning PID Controller Uses Pattern Recognition Approach", Control Engineering, pp. 106-111, junho.
- PEREIRA, L.F.A. (1989) "Implementação de um Ambiente de Desenvolvimento de Controladores Inteligentes", Tese de Mestrado - Instituto Tecnológico de Aeronáutica, pp. 82-83, julho.
- ZIEGLER, J.G. & NICHOLS, B.N. (1942) "Optimum Settings for Automatic Controllers", Transactions of the ASME, vol.64, nº8, pp. 759-768, novembro.
- PEREIRA, L.F.A., YONEYAMA, T., CARDOZO, E. (1989) "An Intelligent PID Controller Based on Expert System Techniques", System Science X, Wroclav, Poland, setembro.
- SHINSKEY, F.G. (1979) "Process Control Systems", New York, 2nd Edition, Mc Graw-Hill.