

Imagens/UFRN-eps-converted-to.pdf

Imagens/DCA-eps-conve

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E AUTOMAÇÃO
CURSO DE ENGENHARIA MECATRÔNICA

RELATÓRIO DA 2º PARTE DO 1º TRABALHO DE
CONTROLE FUZZY DE SISTEMAS DINÂMICOS

DANIEL HENRIQUE SILVA FERNANDES: 2015010099

FERNANDO LEANDRO FERNANDES: 20150146106

ROSENILDO PEREIRA AGUIAR FURTADO: 20150146554A

TIAGO BATISTA SILVA SOUSA: 20150146198

Natal-RN

2016

DANIEL HENRIQUE SILVA FERNANDES: 2015010099

FERNANDO LEANDRO FERNANDES: 20150146106

ROSENILDO PEREIRA AGUIAR FURTADO: 20150146554A

TIAGO BATISTA SILVA SOUSA: 20150146198

RELATÓRIO DA 2º PARTE DO 1º TRABALHO DE CONTROLE FUZZY DE SISTEMAS DINÂMICOS

Primeiro Relatório apresentado à disciplina de Controle Fuzzy de Sistemas dinâmicos, correspondente à avaliação da 1º unidade do semestre 2016.2 do 9º período do curso de Engenharia Mecatrônica da Universidade Federal do Rio Grande do Norte, sob orientação do **Prof. André Henrique Matias Pires**.

Professor: André Henrique Matias Pires.

Natal-RN
2016

RESUMO

A segunda parte do primeiro trabalho desta disciplina tem como objetivo implementar o controle, usando prolog em duas configurações diferentes, para uma planta de segunda ordem. A primeira configuração do controle baseado em prolog fornecerá o valor de entrada do sistema. Na segunda configuração, o prolog fornecerá os valores dos parâmetros K_p , K_i e K_d para o controlador PID que controlará a planta. A planta escolhida foi um sistema formado por dois tanques, onde uma bomba alimenta o primeiro tanque, que por sua vez, alimenta o segundo por gravidade. Esse relatório pretende explicar o desenvolvimento do projeto, dando uma base teórica no conhecimento necessário para simular o controle do nível do segundo tanque usando as duas configurações feitas em prolog e também mostrar e comentar os resultados obtidos com o final da simulação.

Palavras-chave: Controle PID, prolog, Sistemas de tanques.

LISTA DE SÍMBOLOS

$e(t)$	Erro.
L_1, K	Nível do Tanque 1.
L_2, K	Nível do tanque 2.
K_p, K	Constante de proporcionalidade.
K_d	Constante da ação derivativa.
K_i	Constante da ação integral.
K_m	Constante da bomba.
$r(t)$	referência.
$u(t)$	Degrau unitário.
$x(t)$	Estado do sistema.
$y(t)$	Saída do sistema.

LISTA DE ABREVIATURAS E SIGLAS

MV	Variável Manipulada, em inglês
P	Proporcional
PD	Proporcional Derivativo
PI	Proporcional Integrativo
PID	Proporcional Integrativo Derivativo
SP	Set Point

Lista de Figuras

Lista de Equações

Sumário

1 INTRODUÇÃO

O prolog é uma das linguagens mais utilizadas em inteligência artificial. Um dos motivos para isso é porque ela permite que uma aplicação possa inferir fatos baseados em outros fatos e regras pre-estabelecidas para tomar decisões. Esse conceito também está sendo usado na automação de processos, como por exemplo o controle inteligente que está implantado desde eletrodomésticos e brinquedos até plantas industriais complexas.

As plantas industriais precisam de controladores porque muitas vezes, o modelo matemático dela não atende, satisfatoriamente, aos parâmetros desejados, como tempo de subida, overshoot, entre outros. Eles atuam em variáveis do sistema, denominadas MV (Variáveis Manipuladas, em inglês) que modificam a sua resposta de saída, seja por inserir polos, zeros ou alterando o seu ganho. Essas variáveis de saídas são chamadas de PV (Variáveis Controladas, em inglês).

Elaborou-se regras em prolog para controlar um sistema de tanques de segunda ordem, em malha fechada, onde o programa fornecerá os dados para a entrada da planta ou para o controlador dela, conforme o erro da saída.

2 REFERENCIAL TEÓRICO

O modelo matemático usado para simulação foi baseado no sistema de tanques da Quanser. O sistema em questão é composto por dois tanques conectados um ao outro por meio de um orifício. Uma bomba eleva o líquido do depósito até o primeiro tanque, onde desce pelo orifício para o segundo tanque. A distribuição dos tanques é dada da seguinte forma, como mostrado na Figura ??.

Figura 1: Configuração dos tanques



Fonte: Roteiro 1 do laboratório Sistemas de Controle

A equação diferencial deste sistema tem o seguinte formato:

$$ay''(t) + by'(t) + cy(t) = dx(t) \quad (1)$$

Consultando o roteiro do 5º experimento de Sistemas de Controle, obteve-se as equações diferenciais que descrevem a dinâmica dos dois tanques.

$$\begin{cases} \dot{L}_1 = -\frac{a_1}{A_1} \sqrt{\frac{g}{2l_1}} L_1 + \frac{K_m}{A_1} V_p \\ \dot{L}_2 = \frac{a_1}{A_2} \sqrt{\frac{g}{2l_1}} L_1 - \frac{a_1}{A_2} \sqrt{\frac{g}{2l_2}} L_2 \end{cases} \quad (2)$$

onde os valores das constantes também foram fornecidos: A_1 e A_2 são as áreas da base dos tanques; a_1 e a_2 são as áreas dos orifícios de saídas dos tanques; e K_m é a constante da bomba.

$$A_1 = A_2 = 15,5179 \text{ cm}^2;$$

$$a_1 = a_2 = 0,17813919765 \text{ cm}^2;$$

$$l_2 = 15 \text{ cm};$$

$$l_1 = \frac{a_2^2}{a_1^2} l_2 \text{ cm}$$

$$K_m = 3,3;$$

Substituindo estes valores nas equações acima, obtém-se o seguinte sistema:

$$\begin{cases} \dot{L}_1 = 0.99346032L_1 + 0.01477313V_p \\ \dot{L}_2 = 0.00326984L_1 + 0.99346032L_2 \end{cases} \quad (3)$$

2.1 Controlador Proporcional + Integral + Derivativo (PID)

Um controlador PID é a junção das ações proporcional, integral e derivativa. Onde a ação proporcional reajusta o ganho do sistema, a ação derivativa melhora a resposta transitória e a integrativa age na resposta em regime permanente para eliminar o erro de estado estacionário. A equação 4 refere-se a ação do PID no tempo e a equação 5, na frequência.

$$u(t) = K_p \left(e(t) \frac{1}{\tau_i} \int_0^t e(\tau) d(\tau) + \tau_d \frac{d}{dt} e(t) \right) \quad (4)$$

$$U(S) = \left(K_p + \frac{K_i}{S} + K_d S \right) E(S) \quad (5)$$

2.2 Prolog

Prolog é uma linguagem descritiva e prescritiva. Ela lida com objetos que são as coisas sobre as quais desejá-se raciocinar. Declara-se fatos e regras a respeito destas coisas e depois faz-se perguntas a respeito delas. O programa responderá se a questão é verdadeira ou falsa ou, dependendo da pergunta, inferir uma resposta lógica baseado nos fatos ou nas regras que foram pre-estabelecidas.

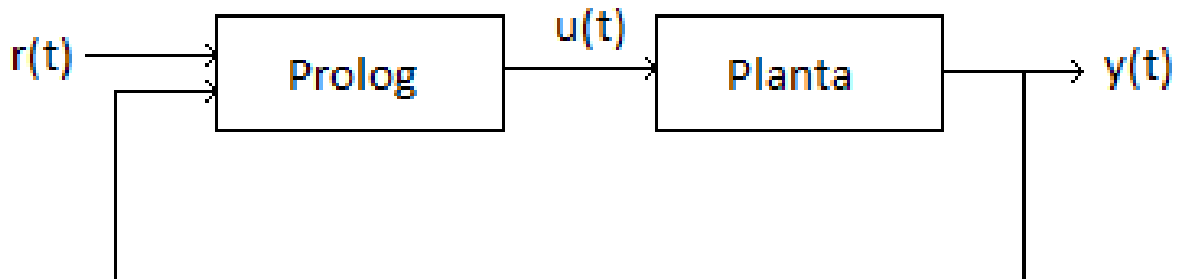
3 METODOLOGIA

Para fazer a simulação do sistema, optou-se por usar Python pela facilidade de programação e pela conectividade com o prolog. Utilizou-se a própria IDE do python para escrever os códigos. Para a conexão com o prolog, incluiu-se a biblioteca pyswip. Com ela, é possível criar regras idênticas a estrutura do prolog utilizando python. Para isso foi preciso:

- Ter Python versão 2.7.12 e a versão mais nova do Prolog;
- Instalar as bibliotecas pyswip e matplotlib do python;
- Compartilhar a pasta "C:\Program Files (x86)\swipl\bin" em variáveis de ambiente;
- Criar uma copia e renomear o libpyswip.dll para libpl.dll dentro da pasta "C:\Program Files (x86)\swipl\bin"

Em seguida, elaborou-se os códigos correspondentes a cada configuração solicitada no trabalho. A primeira onde o prolog fornecerá a entrada para o sistema, conforme está ilustrado na figura ??.

Figura 2: primeira configuração do prolog

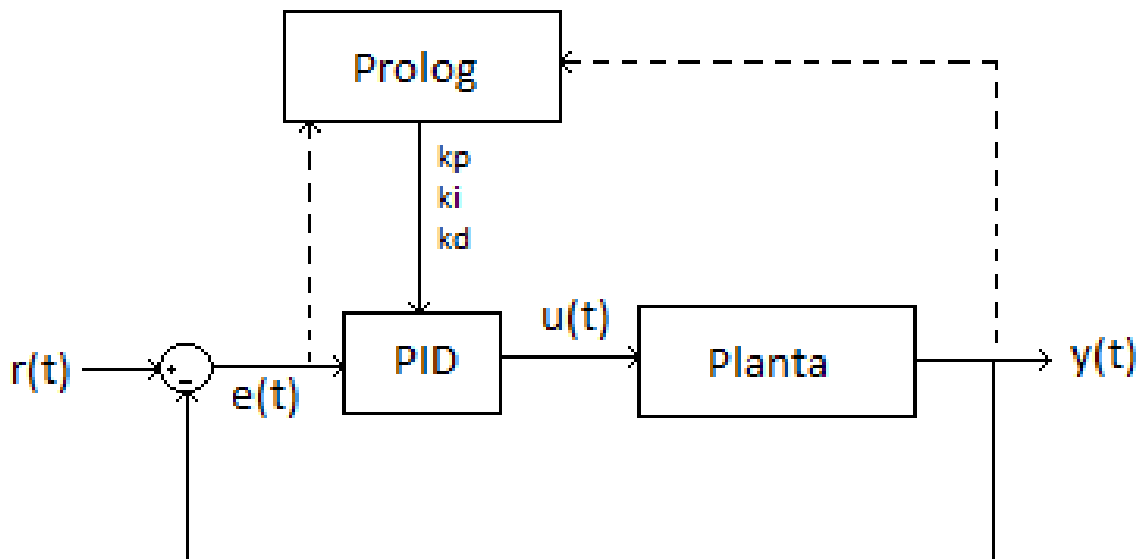


Para esta configuração, criou-se regras em prolog para que ele forneça a tensão da bomba de acordo com o nível do segundo tanque, como por exemplo:

- `tanque(N,E):- N<10, E is 4.`
- `tanque(N,E):- N>=10, N<=20, E is 2.`
- `tanque(N,E):- N>20, E is 0.`

A segunda configuração, o prolog fornecerá os parâmetros do controlador. no caso, K_p , K_i e K_d . A figura ?? mostra o esquema desta configuração.

Figura 3: segunda configuração do prolog



As regras usadas nesta configuração estão descritas a seguir:

- $\text{tanque}(E, K_p, K_i):- E \leq (-30), K_p \text{ is } 0.1, K_i \text{ is } 0.001.$
- $\text{tanque}(E, K_p, K_i):- E > (-30), E < 0, K_p \text{ is } 1, K_i \text{ is } 0.01.$
- $\text{tanque}(E, K_p, K_i):- E \geq 0, E < 30, K_p \text{ is } 1, K_i \text{ is } 0.01.$
- $\text{tanque}(E, K_p, K_i):- E \geq 30, K_p \text{ is } 0.1, K_i \text{ is } 0.001.$

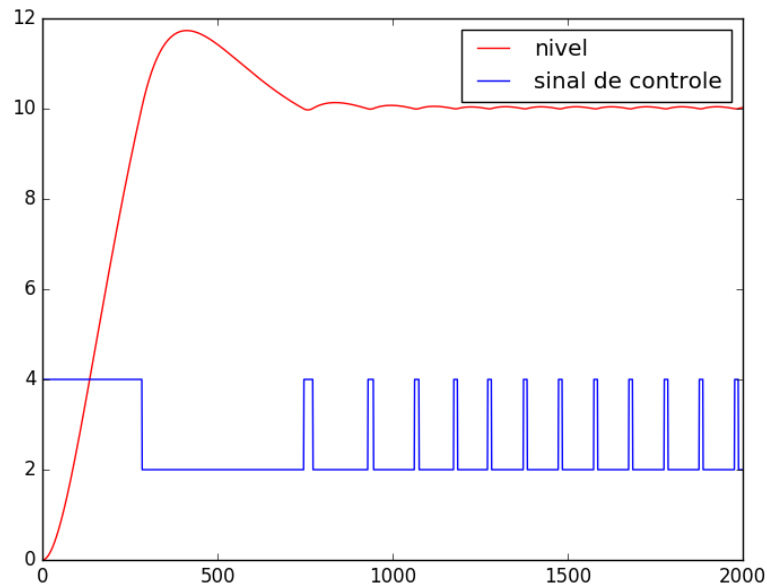
Nas regras acima, nota-se a ausência do K_d . Isso aconteceu porque nos testes, os resultados foram melhores quando $K_d=0$. Logo, ele foi retirado somente para obter um código menor, sem nenhum prejuízo no resultado.

4 RESULTADOS

A biblioteca pyswip facilitou bastante a conexão entre o python e o prolog, já que ela permitia escrever as regras em prolog dentro do código em python.

Para a simulação da primeira configuração, utilizou-se valores de setpoint entre 0 e 20cm. Conforme pode ser visto nas regras para essa configuração, só são permitidos 3 valores de tensão, 0, 2 e 4 volts, dependendo da saída. A figura ?? mostra a resposta da planta ao setpoint de 10cm.

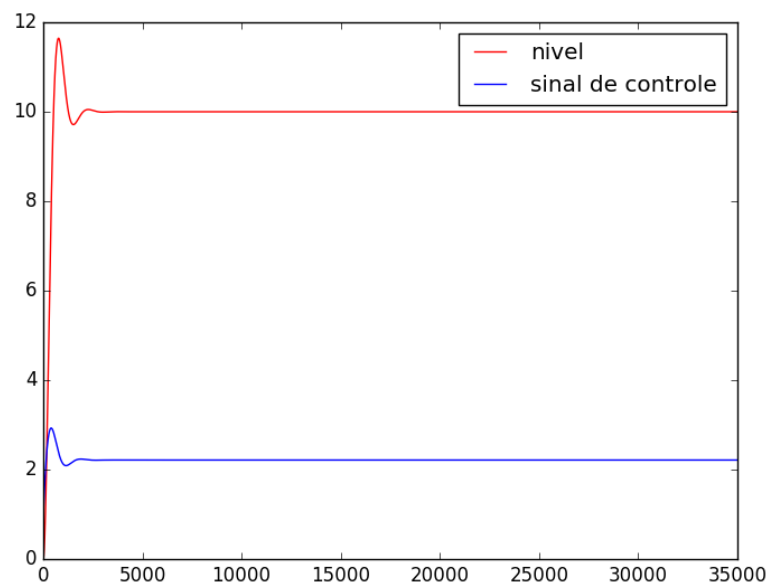
Figura 4: Resposta da planta para a configuração 1 em 10cm.



Devido ao pouco número de regras, pode ser observado na imagem a mudança brusca do sinal de controle, entre 2 e 4V.

Para simular a segunda configuração, utilizou-se valores de referência entre 0 e 30cm. Primeiro, setou-se a referência em 20cm, obtendo-se o resultado mostrado na figura ??.

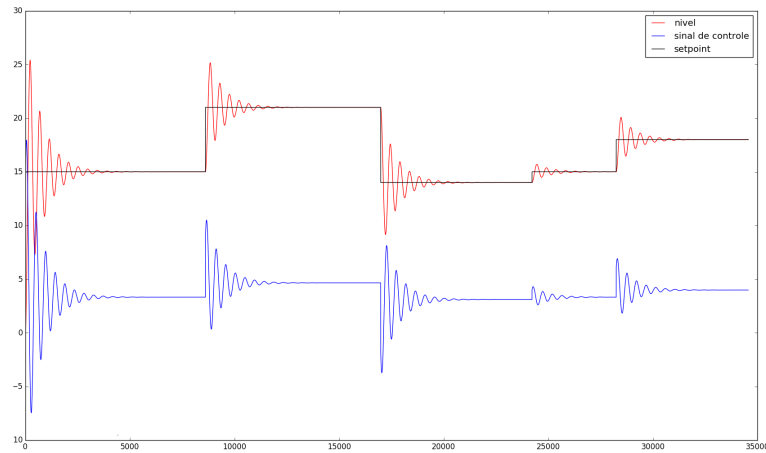
Figura 5: Resposta da planta para a configuração 2 em 10cm.



Nesta figura, nota-se a boa atuação do controlador, apesar do pouco número de regras implementadas para esta configuração, apenas 4. O regime transitório apresentou pouca oscilação e o erro de regime permanente foi de aproximadamente zero.

Depois, modificou-se o código para que ele alterá-se o setpoint, aleatoriamente, cada vez que o erro fosse menor que 0,000001. O resultado pode ser visto na figura ??.

Figura 6: Resposta da planta para a configuração 2 com setpoint aleatório.



A planta apresentou uma resposta muito oscilatória no regime transitório. Poderia-se inserir a constante da ação derivativa para reduzir este efeito, porem os resultados mostraram que uma planta pode ser controlada, eficientemente, usando regras em prolog.

5 CONCLUSÃO

Apesar da estrutura do prolog não parecer tão intuitiva e sequencial como em outras linguagem, como C/C++, python, ela permite que se possa tomar decisões lógicas baseadas em fatos e regras disponíveis. No caso desta simulação, poderia-se analisar algumas situações onde a resposta não foi tão satisfatória e adicionar mais regras para melhorá-las, como por exemplo, aumentar o número de regras para diminuir a variação de tensão da bomba na configuração 1 ou criar regras de valores por faixa de setpoint.

Esta simulação serviu para mostrar que o prolog é adequado para o controle inteligente de sistemas industriais. Embora não se tenha obtido resultados muito satisfatórios, sejam por falta de tempo para uma melhor sintonia ou pela falta de experiência no uso desta linguagem, ela serviu para implementação prática do que foi abordado em sala de aula.

6 REFERÊNCIAS

- [1] Apostila da disciplina Paradigmas Programação, Prolog, do Departamento de Engenharia de Computação e Automação – Universidade Federal do Rio Grande do Norte
- [2] Apostila da disciplina Sistemas de Controle do Departamento de Engenharia de Computação e Automação – Universidade Federal do Rio Grande do Norte
- [3] OGATA, K.: Engenharia de Controle Moderno – 4o Edição, 2003, Prentice-Hall. (OGATA, 2003)
- [4] Roteiro do laboratório 1 da disciplina Sistemas de Controle do Departamento de Engenharia de Computação e Automação – Universidade Federal do Rio Grande do Norte
- [5] Roteiro do laboratório 5 da disciplina Sistemas de Controle do Departamento de Engenharia de Computação e Automação – Universidade Federal do Rio Grande do Norte