



**FEUP** Universidade do Porto  
Faculdade de Engenharia

# Controlo de um sistema através de PID industrial/autómato

## Tecnologia de Sistemas de Controlo e Automação

António Manuel Lopes de Azevedo  
António Pedro Gomes Sousa e Silva

Alunos:  
020503057  
020503058

Docente:  
António Paulo Gomes Mendes Moreira

# Índice

1	Introdução .....	2
2	PID .....	3
2.1	Controlo ON/OFF .....	4
2.2	Controlo Histerese .....	5
2.3	Controlo PID .....	6
2.3.1	Controlo P .....	6
2.3.2	Controlo PI .....	7
2.3.3	Controlo PID .....	8
2.4	Resultados experimentais .....	9
2.4.1	Auto-tuning .....	9
2.4.2	Cálculo do modelo .....	10
2.5	Regras de Chien .....	12
2.5.1	Análise das situações "reais" .....	13
3	Autómato .....	16
3.1	Cartas .....	16
3.1.1	MAD42 .....	16
3.1.2	OD211 .....	18
3.1.3	ID211 .....	18
3.2	PID .....	19
3.3	TPO .....	22
4	Conclusões .....	24

# 1 Introdução

Com este trabalho prático vamos ter um contacto mais prático com os controladores do tipo PID para tal vamos testar alguns modos de funcionamento deste tipo de controladores. Pretende-se através do PID controlar a temperatura de um forno simulado por uma lâmpada.

O ajuste dos parâmetros deve ser feito numa primeira fase utilizando o "auto-tuning" disponibilizado pelo próprio controlador, posteriormente calcular o modelo do sistemas e através dele calcular os parâmetros utilizando o método de Chien.

Por fim deve ser feita uma análise comparativa do desempenho do controlador de temperatura para os diferentes parâmetros.

Assim com este trabalho prático pretende-se ganhar uma maior familiaridade com os controladores do tipo PID percebendo a influência que cada um dos parâmetros de ajuste (P - proporcional; I - integral e D - derivativo) tem no funcionamento do sistema.

## 2 PID

O controlador utilizado no trabalho prático foi o E5Ck da OMRON. Este controlador disponibiliza diferentes tipos de funcionalidades que devem ser configuradas pelo utilizador. Para perceber o funcionamento deste foi então necessário consultar o manual de instruções e configurar os seguintes parâmetros:

- Métodos de controlo: *controlo frio*; *controlo quente*; *controlo frio/quente*;
- Tipo de controlo: *controlo PID* ou *ON/OFF*;
- Ajuste dos parâmetros  $P$ ,  $I$ ,  $D$ ;
- Período de controlo  $T$ ;
- Tipo de entrada (sensor de temperatura - tensão proporcional à temperatura) com ajuste dos limites inferior e superior;
- Tipo de saída do PID (*PWM*);
- Alarmes.

Após ter configurado os principais parâmetros começa então por fazer alguns ensaios onde testamos os alarmes e verificamos a correcto funcionamento do sistema.

Após esta etapa inicial de configurações começamos então por realizar alguns testes.

## 2.1 Controlo ON/OFF

Numa primeira etapa experimentamos o controlo ON/OFF. De seguida está ilustrado um diagrama de blocos que ilustra o funcionamento deste tipo de controlo.

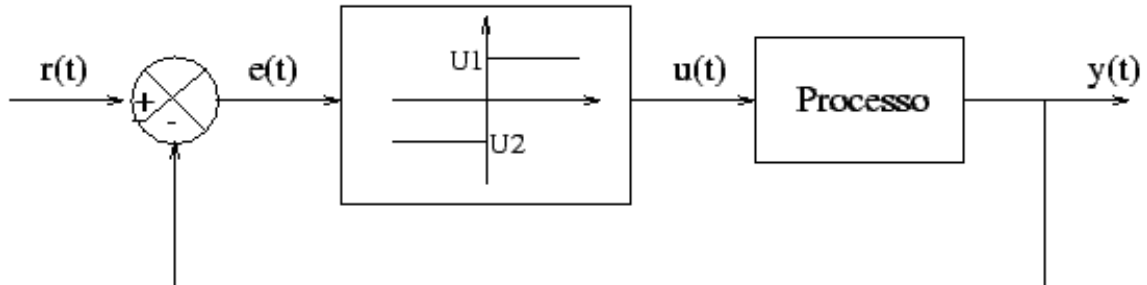


Figura 1 - Controlo ON/OFF

$$u(t) = \begin{cases} U_1 & \text{se } e(t) > 0 \\ U_2 & \text{se } e(t) < 0 \end{cases}$$

Este tipo de controlo é simples e sem parâmetros para calibrar. O processo a controlar atinge a referência, contudo a maioria das vezes oscila.

Foi experimentado este controlar para o nosso processo todavia não foram registados os dados obtidos contudo podemos verificar que com este tipo de controlo o comportamento do sistema não era o melhor já que a variável de controlo apenas tem dois estados comutando entre eles conforme o erro é positivo ou negativo, conforme se pode ver na Figura 1. Verificamos assim tal como era de esperar que o sistema oscilava bastante em torno da referência.

## 2.2 Controlo Histerese

Foi ainda experimentado o controlo com histerese. Neste tipo de controlo a variável de controlo tem em função do erro o comportamento abaixo ilustrado.

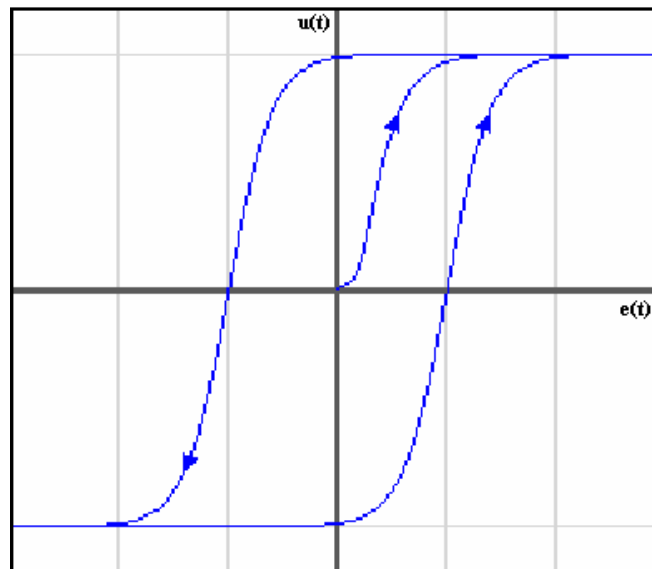


Figura 2 - Controlo histerético

De notar que com a introdução da histerese a “sensibilidade” do controlador a pequenas variações em torno da referência é menor como se pode ver no gráfico acima e foi possível visualizar no processo em causa uma vez que o sistema apesar de ainda oscilar em torno da referência essa oscilação era bastante menor.

## 2.3 Controlo PID

Antes de implementarmos o controlo PID no sistema e apesar de ter sido objectivo de estudo já em cadeiras anteriores optamos por fazer algumas simulações no Matlab para relembrar a influência dos diferentes parâmetros no controlador.

Assim testamos a resposta de um sistema com a seguinte função de transferência:  $G(s) = \frac{1}{(s+1)^3}$  com um controlador P, PI e PID com diferentes parâmetros e para uma entrada em degrau com amplitude 10.

### 2.3.1 Controlo P

Para o controlador do tipo proporcional testamos três ganhos diferentes obtendo os resultados abaixo ilustrados.

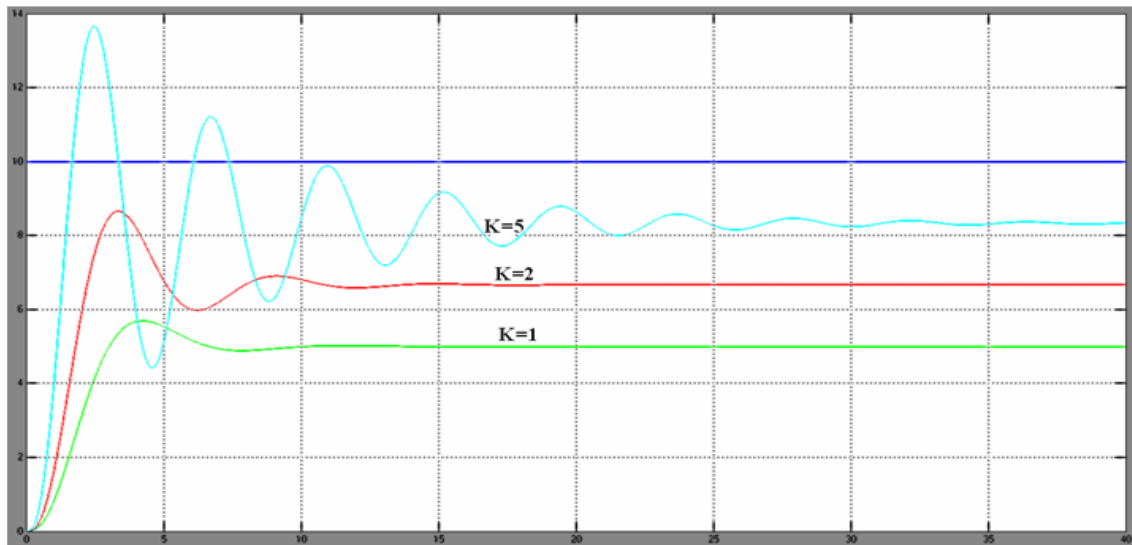


Figura 3 - Diferentes tipos de resposta ao controlo proporcional

Como se pode ver na figura acima apenas com um ganho K, o sistema tem um erro em regime permanente ao degrau tanto maior quanto menor é o ganho. Contudo, o aumento do ganho não é a forma melhor de resolver este erro em regime permanente já que com esse aumento o sistema oscila como se pode constatar nas simulações.

### 2.3.2 Controlo PI

De seguida temos então um controlador do tipo PI onde  $K=1$  e testamos três “tempos integrais”- $t_i$  diferentes. Os resultados obtidos para esta simulação estão ilustrados no gráfico abaixo. De notar que a acção integral tem como finalidade melhorar a estabilidade do sistema em malha fechada.

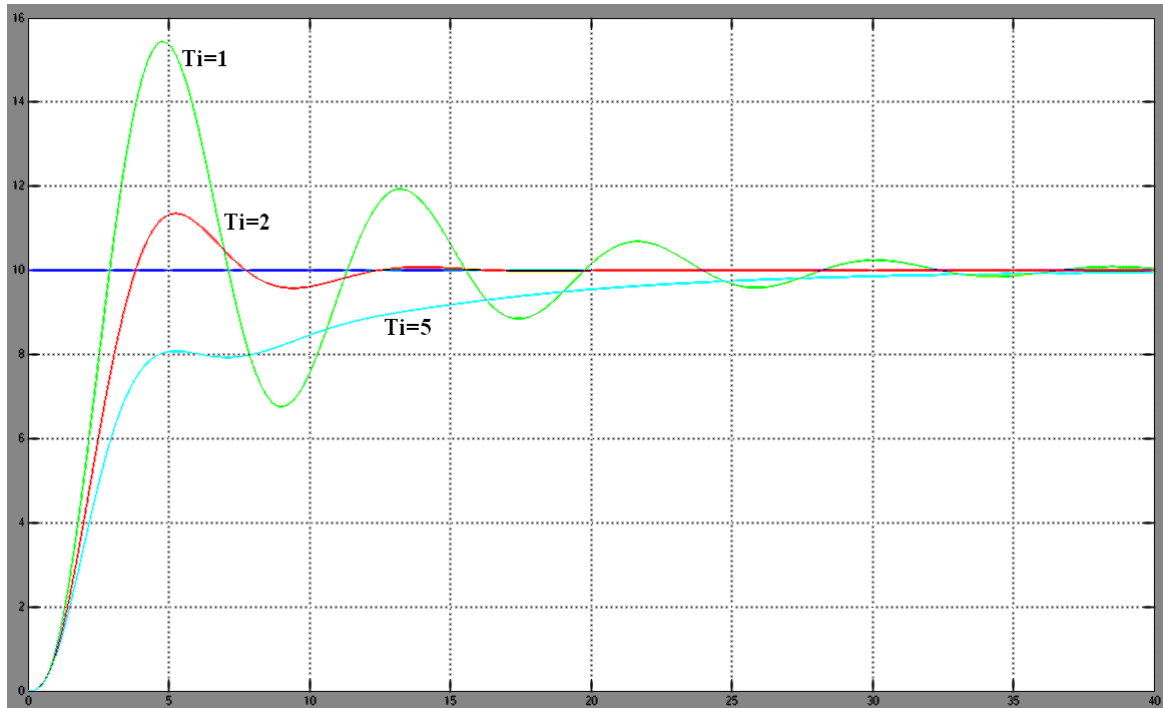


Figura 4 - Diferentes tipos de resposta ao controlo PI

Como se pode ver com a introdução da acção integral o erro em regime permanente ao degrau é anulado. Contudo convém não diminuir muito o parâmetro  $T_i$  já que apesar de tornar o sistema mais rápido, aumenta a oscilação e overshoot.



### 2.3.3 Controlo PID

Por fim simulamos então como seria o comportamento do sistema com um controlador do tipo PID. Os resultados estão na figura abaixo ilustrado. Foram utilizados um  $k=3$  e um  $T_i=2$ .

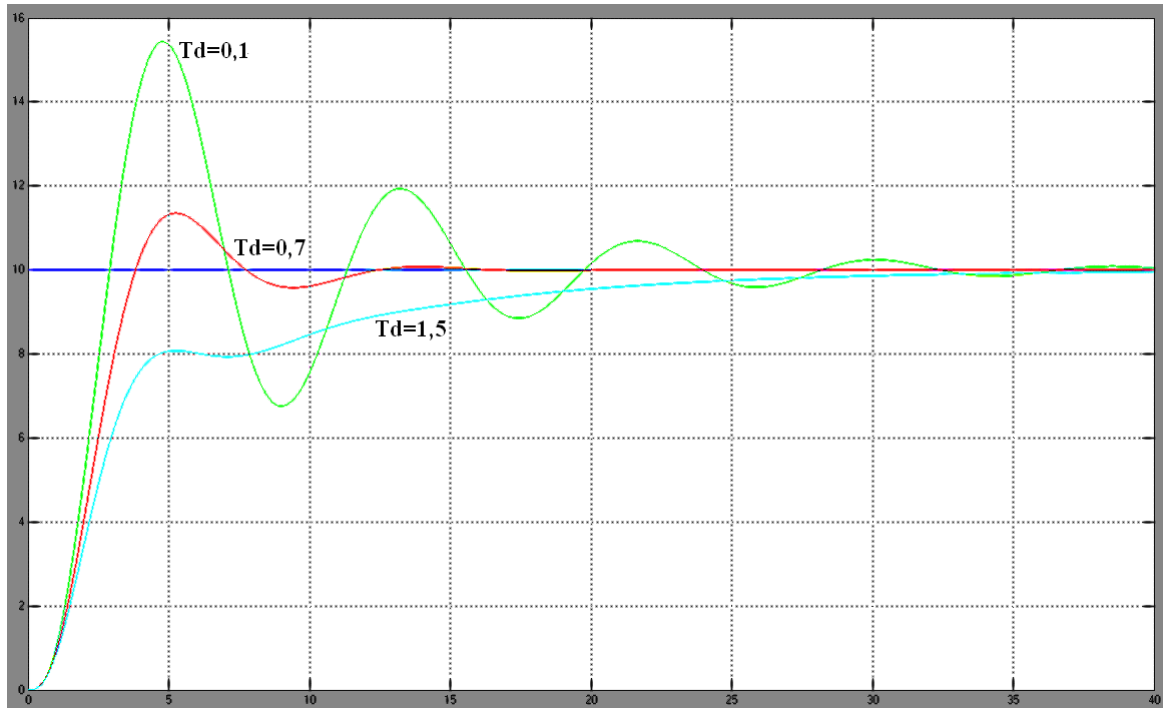


Figura 5 - Diferentes tipos de resposta ao controlo PID

Como se pode observar na simulação, a introdução da acção derivativa fez com que o sistema pudesse ter uma resposta mais rápida sem oscilar, já que este factor proporcional à derivada do erro permite ao controlador “prever” o erro futuro.

Assim com um controlador do tipo PID e desde que os parâmetros estejam bem calibrados podemos controlar um sistema com erro nulo ao degrau em regime permanente, boa resposta dinâmica principalmente em sistemas lentos (como é o caso do trabalho prático - sistema térmico) com um controlo simples de implementar.

A influência dos parâmetros de calibração do PID foi também testada no sistema real do trabalho prático, todavia a evolução da temperatura não foi registada de forma a poder traçar a resposta do sistema, contudo os resultados obtidos foram em tudo semelhantes aos atrás analisados.

## 2.4 Resultados experimentais

### 2.4.1 Auto-tuning

Depois de termos tido um primeiro contacto com o PID quer com a sua configuração quer com um controlo elementar do tipo ON/OFF, passamos para o controlo PID com "auto-tuning", isto é, controlamos o processo com um controlador PID onde os parâmetros P, I, e D foram calculados automaticamente pelo próprio controlador.

Os resultados obtidos foram registados numa folha de excel para desenhar a resposta do sistema que se encontra abaixo ilustrada.

Os parâmetros calculados pelo controlador foram:

<b>BP</b>	10,9
<b>I</b>	2
<b>D</b>	42

Tabela 1 - Parâmetros do auto-tuning

Para testar o controlador, foi estabilizado o sistema numa dada temperatura e feita uma variação em degrau da referência (essa variação foi de aproximadamente 10°C).

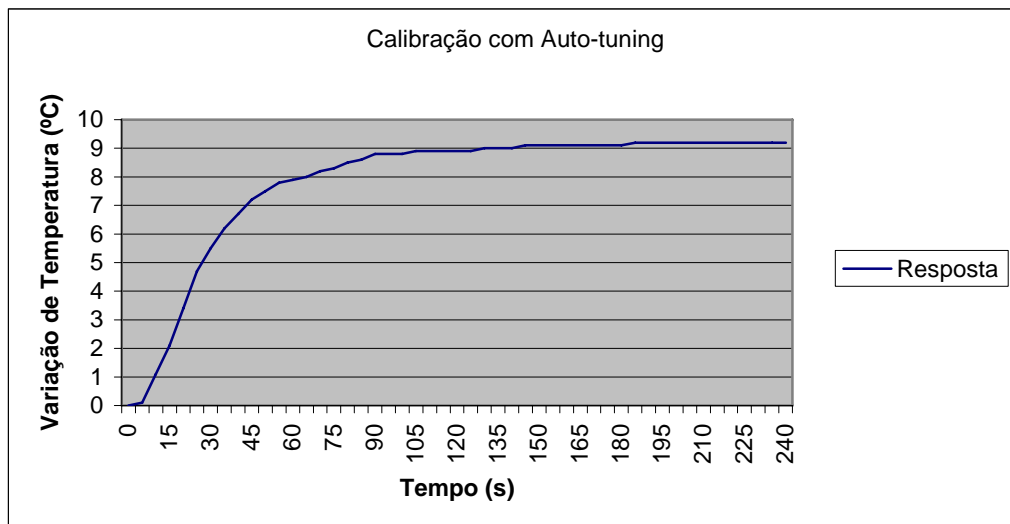


Figura 6 - Resposta com auto-tuning

Como se pode ver a resposta do sistema foi boa já que não existe overshooting, e é relativamente rápida tempo em conta que é um sistema térmico.

## 2.4.2 Cálculo do modelo

Contudo nem sempre os controladores PID nos disponibilizam esta opção de “auto-tuning”, logo é necessário saber calcular os parâmetros do PID. Para se poder calcular esses parâmetros é necessário ter o modelo do processo a controlar.

A seguinte etapa do trabalho foi assim calcular o modelo do processo de forma a poder depois calcular os parâmetros para o controlador.

Para calcular o modelo do sistema estabilizou-se o sistema aplicando na entrada um PWM constante. Após o sistema estar estabilizado aplicou-se um degrau e registou-se os valores da saída de cinco em cinco segundos até o sistema estabilizar. Os valores foram registados numa folha de cálculo excel de forma a poder traçar o gráfico da saída.

O sistema aproximado foi de primeira ordem para facilitar os cálculos contudo convém realçar que na realidade o sistema é de ordem superior, como se pode observar na curva da saída que para os instantes iniciais é côncava.

A expressão do sistema de primeira ordem é a seguinte:

$$y(t) = K \left( 1 - e^{-t/\tau} \right)$$

Foi feita uma aproximação grosseira do K e do tau e calculado o erro quadrático do sistema e da aproximação. Utilizando o solver do excel de forma a minimizar o erro quadrático foi possível calcular os novos parâmetros K e  $\tau$  que modelam o processo em causa.

Esses parâmetros estão na tabela abaixo. De seguida está também ilustrado um gráfico comparativo entre o sistema real e o modelo com os parâmetros calculados pelo solver do excel.

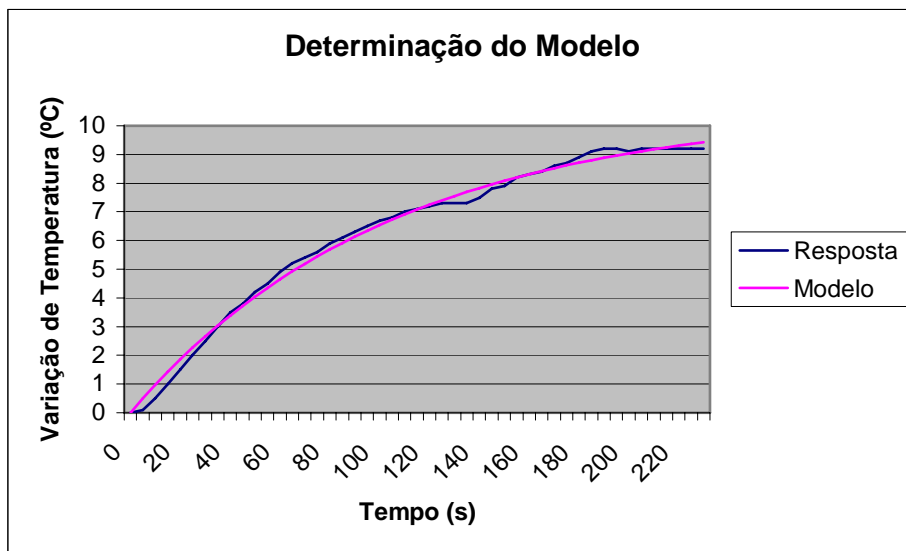


Figura 7 - Modelo do sistema

K	0,783501
$\tau$	103,8195

Tabela 2 - Parâmetros do modelo

## 2.5 Regras de Chien

Depois de termos o modelo do sistema estamos agora em condições de calcular os parâmetros do controlador para o respectivo processo. Para tal foram utilizadas as regras de Chien para modelos de primeira ordem através da seguinte tabela:

Process model	$k_c k_p$	$T_I$	$T_D$
$\frac{k_p e^{-Ls}}{1+\tau s}$	$\frac{\tau}{\tau_{cl}+L}$	$\tau$	—
$\frac{k_p(\tau_3 s+1)e^{-Ls}}{(\tau_1 s+1)(\tau_2 s+1)}$	$\frac{\tau_1+\tau_2-\tau_3}{\tau_{cl}+L}$	$\tau_1 + \tau_2 - \tau_3$	$\frac{\tau_1 \tau_2 - (\tau_1 + \tau_2 - \tau_3)\tau_3}{\tau_1 + \tau_2 - \tau_3}$
$\frac{k_p(-\tau_3 s+1)e^{-Ls}}{(\tau_1 s+1)(\tau_2 s+1)}$	$\frac{\tau_1+\tau_2+\frac{\tau_3 L}{\tau_{cl}+\tau_3+L}}{\tau_{cl}+\tau_3+L}$	$\tau_1 + \tau_2 + \frac{\tau_3 L}{\tau_{cl}+\tau_3+L}$	$\frac{\tau_3 L}{\tau_{cl}+\tau_3+L} + \frac{\tau_1 \tau_2}{\tau_1 + \tau_2 + \frac{\tau_3 L}{\tau_{cl}+\tau_3+L}}$
$\frac{k_p e^{-Ls}}{s}$	$\frac{2\tau_{cl}+L}{(\tau_{cl}+L)^2}$	$2\tau_{cl} + L$	—

Tabela 3 - Regras de Chien

Onde  $\tau_{cl}$  é a constante de tempo em malha fechada,  $\tau$  a constante de tempo do processo e L o atraso que neste caso é zero.

Este tipo de regra tente colocar o zero do controlador no mesmo ponto que o pólo do sistema assim idealmente o lugar de raízes do conjunto controlador mais processo seria semelhante ao abaixo ilustrado.

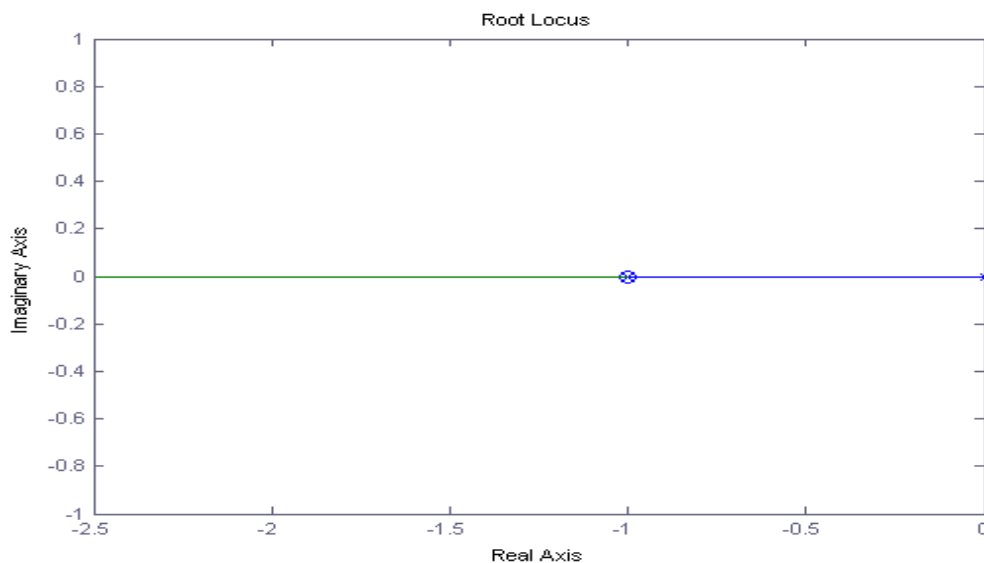


Figura 8 - Caso óptimo

Como se pode ver o sistema é estável e com um boa resposta. Contudo este é o caso ideal e que muito dificilmente ocorre na realidade.

Assim podemos ter duas situações distintas que de seguida vão ser analisadas.

### 2.5.1 Análise das situações “reais”

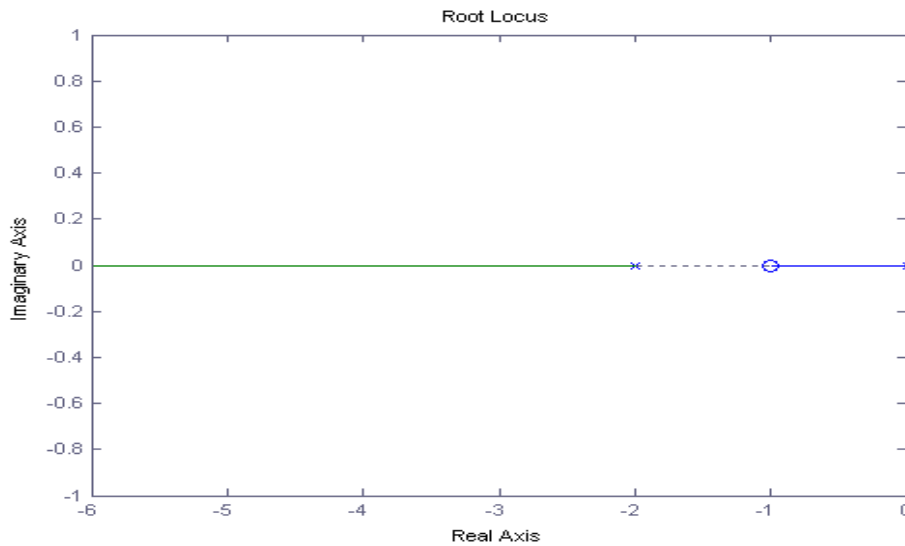


Figura 9 - Caso bom

Quando o pólo introduzido pelo controlador se situa à direita do pólo do sistema este pode ter uma resposta inicial rápida e depois demora algum tempo até chegar à referência, contudo o sistema não oscila.

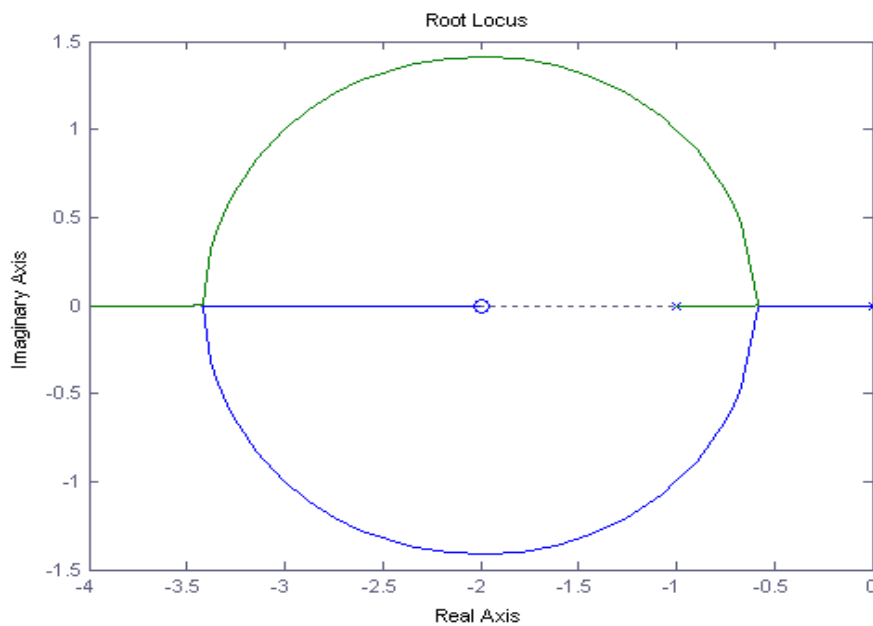


Figura 10 - Caso mau

Caso o pólo introduzido pelo controlador fique à esquerda do pólo do processo, o sistema pode oscilar tal como se pode ver pelo lugar de raízes acima desenhado.

Tendo em conta os aspectos acima realçados, estudamos o comportamento do sistema (processo mais controlador) onde os parâmetros do controlador foram calculados pelas regras de Chien.

O cálculo dos parâmetros do controlador para um sistema de primeira ordem através do método de Chien é feito através das equações seguintes:

$$k_c k_p = \frac{\tau}{\tau_{cl} + L}$$

$$T_i = \tau$$

A constante  $L$  representa o atraso do processo, que no nosso caso é 0 e a constante integrativa é igual à constante de tempo do processo.

A banda proporcional para o controlador PID no caso do Chien 2 é calculada da seguinte maneira:

$$BP = \frac{100}{k_c} \text{ logo } BP = 100 \times k_p \times \frac{\tau_{cl}}{\tau}$$

$$\text{Como } \tau_{cl} = \frac{\tau}{2}, BP = \frac{100 \times k_p}{2} \approx 39,2.$$

Na tabela abaixo estão calculados os ganhos para diferentes constantes de tempo de malha fechada (2, 4, 8 vezes mais rápidas). De notar que o aumento das constantes vai diminuir a banda proporcional, logo diminuir o tempo de estabelecimento.

	<b>Chien 1</b>	<b>Chien 2</b>	<b>Chien 4</b>	<b>Chien 8</b>
<b>Kp</b>	0,783501	0,783501	0,783501	0,783501
<b>Kc</b>	1,276322	2,552644	5,105289	10,21058
<b>Bp</b>	78,35013	39,17506	19,58753	9,793766
<b>Ti</b>	103,8195	103,8195	103,8195	103,8195

Figura 11 - Parâmetros de Chien para os vários ensaios

Os resultados práticos obtidos estão ilustrados de seguida.

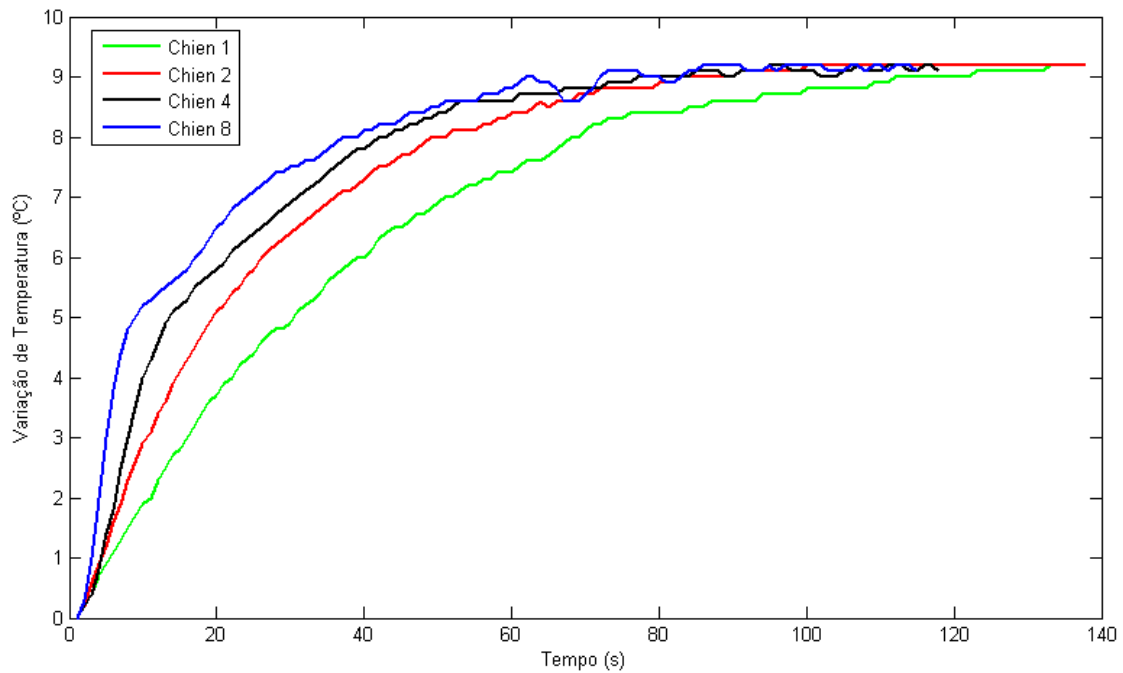


Figura 12 - Resposta do sistema com diferentes parâmetros de Chien

Como se pode ver na figura acima o tempo de estabelecimento do sistema diminui à medida que diminuimos a constante de tempo em malha fechada. Contudo nota-se que para os valores mais baixos dessa constante o sistema tem oscilação. Podemos assim concluir que o aumento da rapidez do sistema só é possível até determinado ponto a partir do qual começa a ser notória oscilação em regime permanente.

Para além da oscilação na saída do sistema foi-nos possível observar uma grande variação na variável de controlo, isto é, apesar de o sistema não oscilar muito a variável de controlo tem grande oscilação.



## 3 Autómato

A segunda parte do trabalho consistia em controlar o processo utilizando as funcionalidades de um autómato. Assim, para executarmos este trabalho temos de interligar o processo e o autómato e de seguida, configurar o PID que é implementado pelo autómato.

O autómato é o Omron SYSMAC CJ1M CPU11e possui a carta de entradas digitais *ID211*, a carta de saídas digitais *OD211* e a carta de entradas e saídas analógicas *MAD42*.

Iremos de seguida explicar como interligamos o processo às cartas do autómato e como o programamos.

### 3.1 Cartas

#### 3.1.1 MAD42

A carta MAD42 é uma carta de entradas e saídas analógicas que nos foi fornecida para trabalhar. Com esta carta, foi-nos possível adquirir a leitura do sensor de temperatura e convertê-la para digital.

Como a gama do sinal de saída do sensor é compatível com a gama de entrada da carta, não foi necessário nenhum circuito adicional.

Assim, escolhemos a entrada 2 para entrada da carta e efectuamos as ligações correspondentes, o sinal ao pino B5 e a massa ao pino B6.

De seguida, foi efectuada a restante configuração da carta. O número da unidade escolhido foi o 0. Desta forma, fica reservado espaço na memória de I/O, 2000 a 2009, para ler o valor de entrada. É também atribuído espaço na memória de dados interna, 20000 a 20099, para configurar o modo como é lido o valor.

Posição de memória	Word escrita	Explicação
D20000	0020	Entrada 2 activada.
D20001	0C00	Gama da entrada 2 0-5V.
D20007	0000	Utilização de dois buffers para o cálculo.
D20018	0000	Bits 0-7 da word. Modo de operação normal. Bits 8-15 da word. Tempo de conversão de 1ms e resolução de 4000.
D20029	0000	Limite inferior da escala - 0 V.
D20030	1388	Limite superior da escala - 5 V (5000).
CIO2006	-	Valor lido.

Tabela 4 - Configuração da MAD42

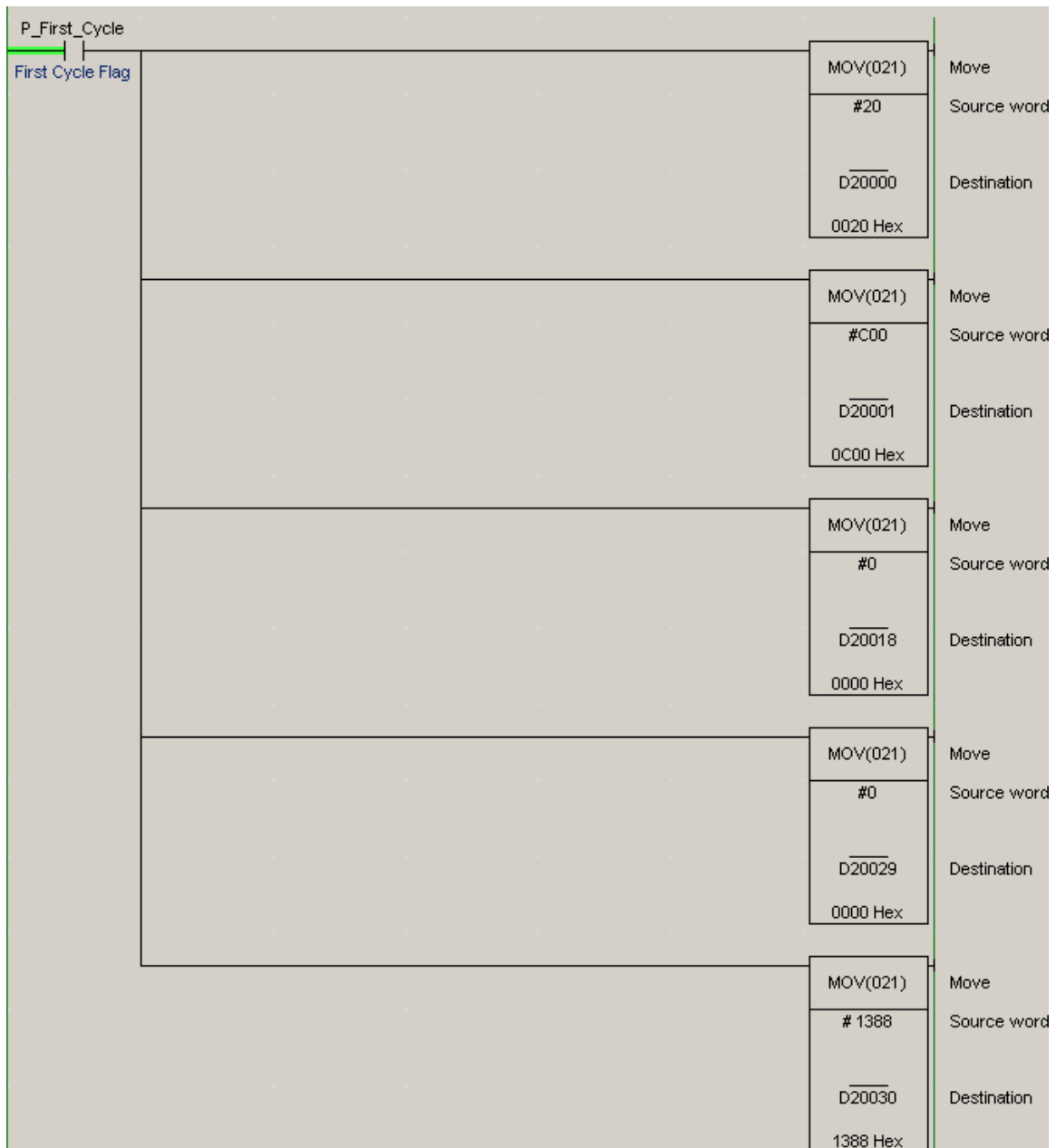


Figura 13 - Implementação das configurações da carta no autômato

Estas configurações são feitas logo que o autômato é iniciado. Com esse objectivo, utilizamos a instrução *P\_First\_Cycle* que este segmento de código nessa altura.

Após a configuração da carta, confirmamos que a leitura era efectuada na posição de memória CIO2006, como era esperado.

### 3.1.2 OD211

A carta OD211 é uma carta de saídas digitais que nos serviu para ligar a saída do autômato, PWM, ao processo a controlar.

Para fazer a ligação correctamente precisamos de saber a configuração da saída da carta. Através da consulta da *datasheet*, concluímos que era do tipo npn. Assim, a interligação foi feita da seguinte maneira, usando a saída digital Q1.00.

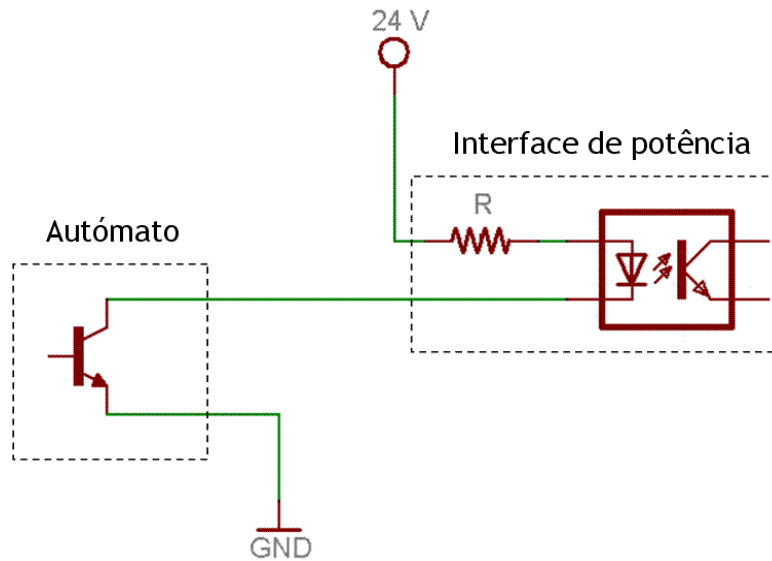


Figura 14 - Ligação da saída do autômato ao processo

### 3.1.3 ID211

Após a ligação da saída digital do autômato, fizemos um pequeno programa de teste para confirmarmos o funcionamento desta.

O programa consistiu em simular o funcionamento de um interruptor. Para isso, ligamos uma entrada digital, I0.00, da carta ID211 à saída digital do autômato. Quando a entrada é accionada, a luz acende-se.



Figura 15 - Teste efectuado

### 3.2 PID

Para realizar a implementação do PID no autômato, utilizamos o bloco de instruções PIDAT. Este bloco, para além de ter a função de PID normal, pode funcionar no modo de auto-tuning, ajustando os parâmetros automaticamente.

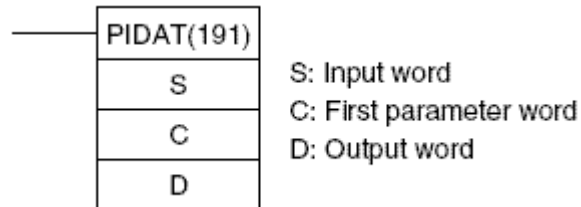


Figura 16 - Bloco PIDAT usado

Neste bloco são definidas três posições de memória, a posição que contém a variável de entrada, a que contém o primeiro parâmetro de configuração e a que contém a variável de saída. No nosso caso, utilizamos as seguintes configurações:

- S = CIO2006;
- C = D200;
- D = D400.

A *word* de saída será depois transformada num PWM a aplicar ao processo através do bloco TPO.

De seguida, configuramos os parâmetros do PID executando diversas instruções de MOV para as posições de memória de configuração do PIDAT. Essas configurações são explicadas na seguinte tabela:

Posição de memória	Word escrita	Explicação
D200	0BB8	Referência do processo
D201	BP	Banda Proporcional
D202	Ti	Tempo Integral
D203	0000	Tempo Derivativo
D204	0032	Período de amostragem. Foi usado 0,5 segundos.
D205	0009	Bits 4-15 da word. Coeficiente do filtro de entrada. Foi usado 0,65 -> 000 hex. Bit 3 da word. Especifica a saída quando o erro é nulo. Usamos saída a 50% -> 1. Bit 1 da word. Novos parâmetros do PID activados no início da instrução -> 0. Bit 0 da word. Acção proporcional com acção "forward" -> 1.
D206	0818	Bit 12 da word. Variável de saída não limitada -> 0. Bits 8-11 da word. Número de bits da word de entrada. Usamos 8 -> 16 bits. Bits 4-7 da word. Tempo integral e tempo derivativo múltiplos do tempo de amostragem -> 1. Bits 0-3 da word. Número de bits da word de saída. Usamos 8 -> 16 bits.
D207	0000	Limite mínimo da variável de saída quando o limite está activo. Neste caso, não nos interessa.
D208	FFFF	Limite máximo da variável de saída quando o limite está activo. Neste caso, não nos interessa.
D209	0(8)000	Bit 15 da word. Bit do auto-tuning. Inicia o auto-tuning na transição 0->1. Pára o auto-tuning na transição 1->0 ou quando acaba o auto-tuning. Bits 0-11 da word. Contribuição dos resultados do PID para os valores armazenados. Usamos 000.
D210	0000	Histerese do ciclo limite. Usamos o valor por defeito 0000->0,20%.

Tabela 5 - Configuração do PID

Para verificar o funcionamento do PIDAT, foi efectuado um teste onde utilizamos BP = 03E8 (100%) e Ti = 270F ( $\infty$ ). Assim garantíamos funcionamento com ganho unitário.

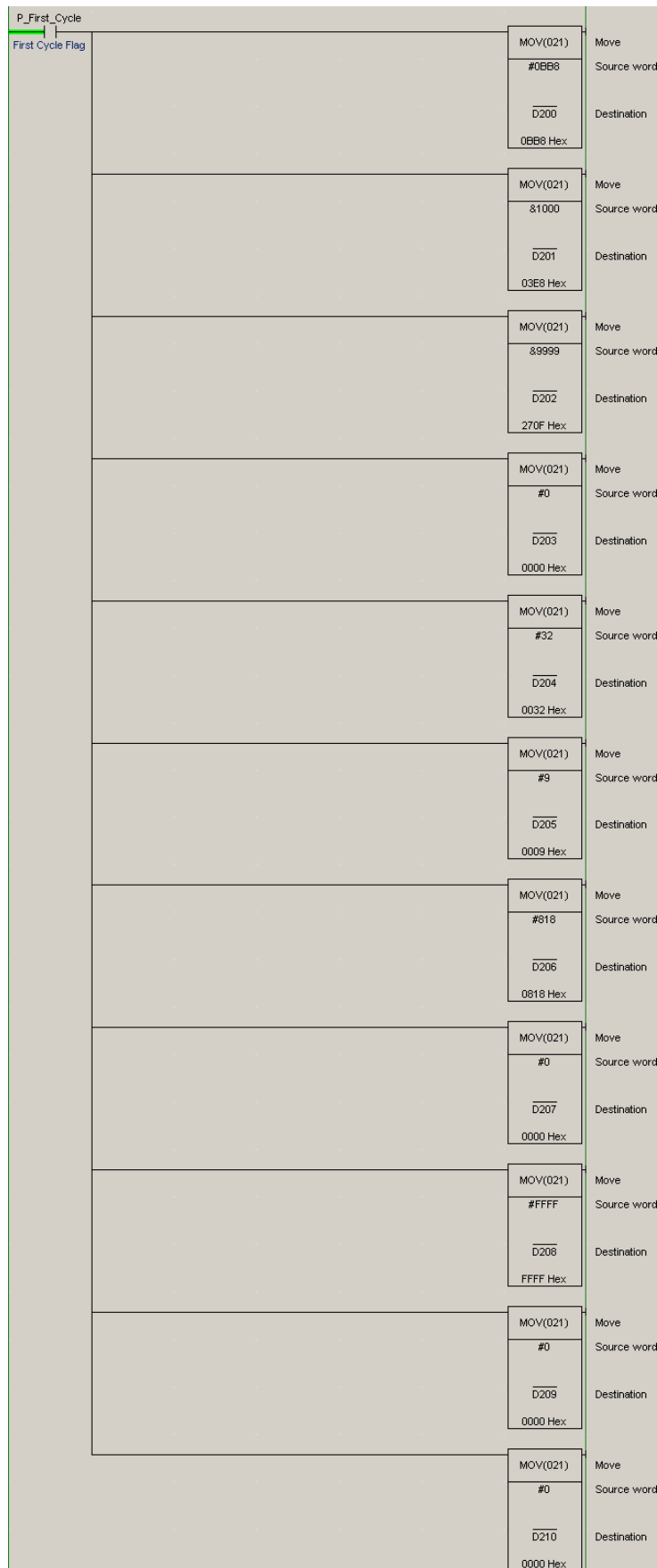


Figura 17 - Implementação das configurações do PIDAT no autômato

Após esta verificação, introduzimos no PID os valores obtidos através das regras de Chien para  $\tau = \tau_d$ .

### 3.3 TPO

Após configurar o PID, constatamos que seria necessário um bloco adicional para gerar o sinal de PWM a aplicar na saída. Desta forma, introduzimos o bloco de instruções TPO.

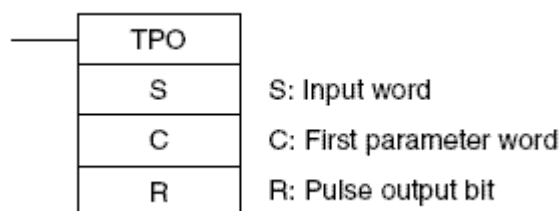


Figura 18 - Bloco TPO utilizado

Tal como no PIDAT, são aqui definidas três posições de memória, a posição que contém a variável de saída do PIDAT e de entrada do TPO, a que contém o primeiro parâmetro de configuração do TPO e a que contém a variável de saída. No nosso caso, utilizamos as seguintes configurações:

- S = D400;
- C = D5000;
- R = Q1.0.

A configuração deste bloco foi igualmente feita usando uma série de instruções MOV para as suas posições de memória de configuração. Essas configurações são explicadas na seguinte tabela:

Posição de memória	Word escrita	Explicação
D5000	0018	Bits 0-3 da word. Número de bits da word de entrada. Usamos 8 -> 16 bits. Bits 4-7 da word. Word de entrada é uma variável manipulada -> 1. Bits 8-11 da word. Word de entrada é lida no início do período de controlo -> 0. Bits 12-15 da word. Saída não limitada -> 0.
D5001	00C8	Período de controlo. Usamos 2 segundos.
D5002	0000	Limite mínimo da variável de saída quando o limite está activo. Neste caso, não nos interessa.
D5003	2710	Limite máximo da variável de saída quando o limite está activo. Neste caso, não nos interessa.

Tabela 6 - Configuração do TPO

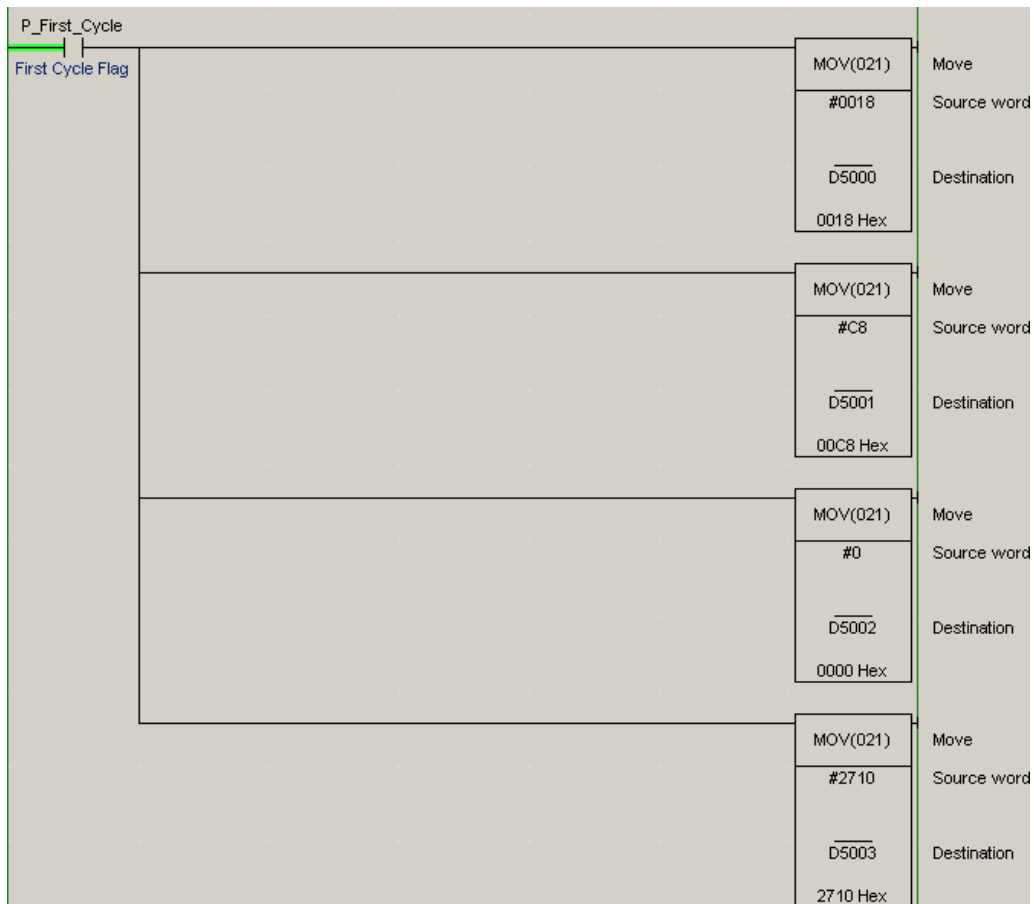


Figura 19 - Implementação da configuração do TPO no autómato

Na implementação do programa, encontramos um problema, devido ao facto de para activar o PIDAT e o TPO termos de forçar a entrada e não pudermos usar a instrução *P\_ON* para o funcionamento contínuo deste.

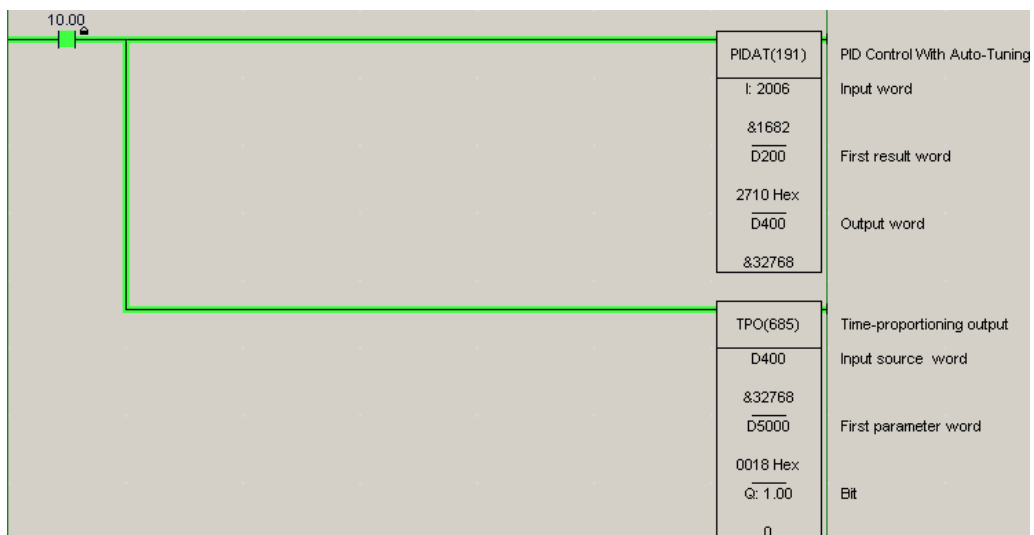


Figura 20 - Implementação dos blocos PIDAT e TPO

Foi também testado o funcionamento do PIDAT com auto-tuning. No entanto, não foi possível colocá-lo em funcionamento apesar da sua configuração ter sido efectuada correctamente.



## 4 Conclusões

Com este trabalho prático, foi-nos possível trabalhar com controladores PID que apesar de simples, são largamente utilizados na indústria.

Pudemos constatar que a modelação do sistema é importante para a calibração dos parâmetros do controlador.

Apesar de se fazer aproximação ao modelo de 1ª ordem, e como é possível constatar, este é de ordem superior, conseguimos calibrar os parâmetros do controlador, obtendo resultados satisfatórios.

Na segunda parte deste trabalho, pudemos ainda contactar com controladores do tipo PID implementados em autómatos industriais.

Foi-nos possível concluir que a utilização destes controladores nos autómatos, apesar de termos comprovado o seu bom funcionamento, só é justificável quando utilizamos o autómato para outras aplicações, uma vez que o seu preço é superior, o tipo de funcionalidades mais alargadas e necessitam de um software próprio para configuração.

Quer o PID industrial, quer o PID implementado no autómato a nível do controlo do processo em causa, tiveram resultados semelhantes e satisfatórios.

## Bibliografia

- **Pid controllers: theory, design, and tuning** - Karl J. Aström e Tore Hägglund
- **Modern control engineering** - Katsuhiko Ogata
- **Dynamical systems and automatic control** - J. L. Martins de Carvalho