Controlo Digital da Posição Angular de um Motor CC

Guia da Componente Prática 2 (CP2)

Projeto Integrador em Engenharia Eletrónica Industrial e Computadores 2

Inês Garcia, Joana Figueiredo, Luís Barros, Luís Louro

Revisão: 2V2 2024

Licenciatura em Engenharia Eletrónica Industrial e Computadores

Escola de Engenharia da Universidade do Minho

1 Introdução

O objetivo principal do guia da componente prática 2 (CP2) é a implementação e a validação do controlo digital da posição angular de uma carga rotativa colocada no sistema mecânico. Para a implementação do controlo digital será utilizado o controlador proporcional-integral-derivativo (PID).

A Figura 1 apresenta o modelo 3D e o protótipo real da maqueta do sistema mecânico, a qual inclui um motor de corrente contínua (CC), uma carga rotativa e o suporte. O motor (MOT02043, DFRobot) já possui um *encoder* (SJ01) e uma caixa redutora 120:1. O *encoder* gera 16 pulsos em quadratura por cada volta do motor. Tendo em consideração a caixa redutora 120:1, cada volta do eixo corresponde a 1920 pulsos em quadratura (Sensor A e Sensor B em conjunto, sendo que cada sensor gera 960 impulsos).

Relativamente à carga rotativa, esta apresenta uma inércia suficiente para que a constante de tempo de resposta do sistema apresente valores facilmente observáveis. No âmbito da CP2, a carga rotativa já se encontra acoplada à maqueta.

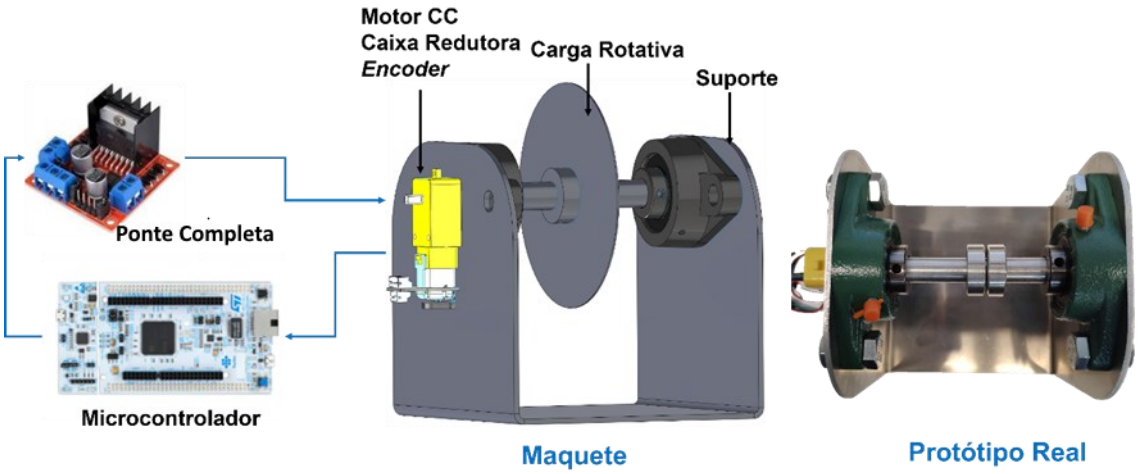

Figura 1 - Representação do sistema mecânico (maqueta e protótipo real), microcontrolador e ponte-completa que serão utilizados para controlar o motor CC do sistema mecânico.

Para implementar o controlo digital da posição angular da carga rotativa é necessário construir o módulo de sensor e de atuação, realizando para isso os dois passos seguintes: i) implementação de funções para medir a posição e velocidade angular da carga a partir da informação digital monitorizada pelo *encoder* de quadratura; e ii) implementação da função de atuação do motor a partir do microcontrolador (STM32). Para tal será necessária a geração de sinais modulados em largura de pulso, em inglês *Pulse Width Modulation* (PWM), e a sua aplicação a um atuador bipolar (um conversor de eletrónica de potência em ponte-completa, vulgarmente designado por *ponte-h*) para acionamento do motor CC. Recomenda-se a utilização de um módulo já com um

conversor de eletrónica de potência em ponte-completa integrado, como por exemplo o driver BTS7960. O terceiro passo, utilizado na secção II deste CP2, consiste na modelização do sistema mecânico e na determinação dos parâmetros do controlador PID, ambos em ambiente de simulação. Por fim, será necessário implementar no microcontrolador a função do controlador PID e validar, em tempo real, o desempenho no protótipo real.

O conjunto de conhecimentos de suporte a esta CP inclui matérias de UCs da área de Automação e Controlo e de Eletrónica de Potência, que devem ser revistas em tudo o que se mostrar necessário. Além disso, deve consultar o Guia Teórico da CP2.

Ao longo desta CP, todas as simulações serão realizadas em Matlab®, enquanto o software para controlar o protótipo real deverá ser implementado no microcontrolador STM32 em linguagem C, utilizando o STM32CubeIDE.

2 Objetivos

A CP2 encontra-se estruturada em duas secções: secção I relativa ao módulo sensor e de atuação; secção II relativa ao sistema de controlo da carga rotativa. Para a secção I é necessária a configuração de todos os periféricos necessários para a aquisição e atuação da carga mecânica. Esta validação deve ser realizada e validada de forma sequencial. Por sua vez, a secção II endereça a modelização do sistema mecânico e a implementação do controlador PID.

Neste trabalho o planeamento de objetivos está previamente definido, sendo necessário entregar os objetivos nas seguintes aulas:

Aulas	Objetivo(s)
1ª aula	0
3ª aula	1, 2
5ª aula	3, 4
7ª aula	5, 6
8ª aula	7, 8

Os objetivos podem ser apresentados antes das aulas indicadas. A não apresentação do objetivo na data estipulada terá uma penalização de 30% por cada aula de atraso.

2.1 Objetivo 0

Neste objetivo inicial deve apresentar:

- **diagrama de blocos** dos elementos constituintes do projeto, identificando não só a ligação entre blocos (sinais de entrada e saída), mas também os valores máximos de operação de cada bloco. Isto é, do motor CC, STM32 (em especial os periféricos de entrada e saída) e módulo da ponte-completa (tensão, corrente, frequências);
- **análise do sistema de controlo**: deve ser apresentado o esquema de controlo da posição angular do sistema mecânico (Figura 1), uma lista das variáveis de controlo (controlada, de atuação, de perturbação, de comando e medida) e deve ser indicado como são constituídos os 4 subsistemas do sistema de controlo;
- **Diagrama da máquina de estados (5 estados, identificados de 0 a 4 no objetivo 1)**: indicar os diferentes estados, as transições entre eles (comando CS), bem como as funcionalidades de cada estado. Identificar as funções de controlo invocadas para cada estado da máquina de estados.

No final do objetivo 0 devem apresentar um resumo das notas de aviso existentes no guia. **O documento relativo a este documento deve ser submetido no diário da *blackboard* antes do início da primeira aula.**

Secção I – Módulos de sensorização e atuação

2.2 Objetivo 1- Implementação da interface com o utilizador

Preparação de aula: fluxograma representativo do código a implementar (transição entre estados de funcionamento/máquina de estados). Mencionar as funções de controlo invocadas para cada estado da máquina de estados tendo em consideração a implementação gradual do sistema, bem como os diferentes estados de funcionamento ativos. Analisar o estado de *reset*, mencionando as variáveis que necessitam de reinicializar nesse estado. Esta preparação vai em conta ao último tópico do objetivo 0, sendo que no início da segunda aula deve existir um fluxograma mais representativo e completo do que se pretende implementar, considerando as recomendações do docente da avaliação do objetivo 0 da primeira aula.

Com este objetivo pretende-se a implementação da estrutura base da algoritmia de controlo. O sistema a implementar necessita de uma interface com o utilizador, do tipo linha de comandos, entre o microcontrolador e o computador local através de comunicação série com norma USB ou RS-232. Como programa de interface pode utilizar o *Terminal* ou similares. Esta interface deve permitir a indicação do **estado de funcionamento** do microcontrolador, bem como a **leitura e escrita**, em tempo real, de valores das variáveis e parâmetros do controlador PID. Para a implementação da interface pode reaproveitar a implementações de guias anteriores, e considere a gramática de interface apresentada no Guia Teórico.

Considere os seguintes estados de funcionamento, os quais devem ser indicados no início do funcionamento do programa principal do microcontrolador:

0) Modo de *reset*:
   a. *Reset* das variáveis de controlo, somatórios e das configurações;
   b. Desativar os pinos de *enable*, de controlo de sentido de rotação e sinal de PWM.
1) Modo configuração (imprimir menu de configuração do utilizador/processamento do microcontrolador):
   a. Configuração de amostragem contínua ou limitada (utilizador);
   b. Configuração do período de amostragem de posição e/ou de velocidade. Também utilizado para o período de amostragem do controlador PID (utilizador);
   c. Configuração de número de amostras (utilizador);
   d. Configuração do valor de *duty-cycle* inicial para o sinal de PWM (utilizador);
   e. Configuração da frequência do sinal de PWM (utilizador);
   f. Configuração da posição zero para o contador de posição (utilizador);
   g. Configuração de parâmetros do controlador em malha fechada (utilizador);
   h. Configuração dos periféricos necessários (microcontrolador);
   i. *Reset* dos somatórios de erros do controlo PID (microcontrolador);
2) Modo controlo manual:
   a. Leitura de velocidade angular ou posição do disco a partir do *encoder*;
   b. Acionamento direto do motor CC, via PWM, em malha aberta.

3) Modo de ensaio experimental em malha aberta:
    a. Análise da resposta forçada e livre do sistema, atuando o motor por um período de 5 s, seguido de um período em resposta livre. Deve ser realizada a monitorização periódica das variáveis de posição ou velocidade angular.

4) Modo automático –controlo PID em malha fechada (obriga a ativar o pino de *enable*).
    a. Controlar motor CC através do controlador PID convencional.

Note que o estado de funcionamento em modo automático corresponde ao controlo em malha fechada, que, por sua vez, também implica a ativação do estado de leitura de valores do *encoder.* A implementação computacional destes estados de funcionamento deve ser realizada no programa principal através de *switch case*, que, por sua vez, permite a ativação de funções específicas para executar cada estado de funcionamento.

Cada estado de funcionamento tem a sua finalidade, existindo uma interface com o utilizador. Contudo, subentenda-se que alguns modos apresentam configurações intrínsecas ao microcontrolador. Assim, para cada caso, considere:

0) *Modo de reset*:
    - Modo de limpeza das configurações iniciais do utilizador (número de amostras, leitura de posição *vs* velocidade, posição zero, modo de ensaio, entre outros);
    - Deve ser desativado qualquer periférico de atuação.

1) *Modo de configuração*:
    - Modo de interface com o utilizador para a configuração de diferentes variáveis necessárias para o controlo (tipo de amostragem, número de amostras, valores de configuração de *duty-cycle* e frequência do sinal de PWM, entre outros);
    - Subentende-se que sempre que este modo é iniciado, deve-se limpar os somatórios de erros existentes no controlo PID;
    - Subentende-se que ao entrar neste modo de funcionamento, a variável de *Enable* deve tomar o valor de 0, e microcontrolador deve inicializar os restantes periféricos (PWM, GPIO, …);
    - Considere que a variável *Enable* do utilizador, além de ter um papel preponderante na gestão dos algoritmos de controlo em funcionamento, deve ter uma manipulação direta ao GPIO de ENABLE existente na ponte-completa;
    - Pode considerar algumas variáveis previamente definidas e que não necessitam de constante configuração do utilizador (por exemplo: frequência do sinal de PWM igual a 1 kHz, *duty-cycle* de 50%).

2) *Modo controlo manual*:
    - Este modo serve para uma manipulação contínua e direta ao valor de *duty-cycle* do sinal de PWM;
    - Assim, após a configuração no *estado 1) Modo de configuração*, o utilizador deve definir este estado (variável CS), sendo que o sinal de PWM e o GPIO de ENABLE apenas ficam ativos após a ativação da variável *Enable* pelo utilizador;
    - A manipulação das variáveis de *Enable* e do *duty-cycle* pode ser realizada no decorrer deste estado. Qualquer outra configuração, o utilizador deve voltar ao *estado 1) Modo de configuração*;

- Ao sair deste modo, o sinal de *Enable* e o GPIO de ENABLE deve ser desativado de forma automática.

3) *Modo de ensaio experimental em malha aberta:*
   - Este modo serve para análise da resposta do sistema, devendo ser previamente configurado os métodos de amostragem no *estado 1) Modo de configuração*;
   - Após configuração e ativação do estado 3, o sistema deve permanecer em modo *espera* até ativação do comando *Enable* pelo utilizador;
   - Após ativação do *Enable*, o microcontrolador deve atuar o motor por um período de 5 s, seguido de um período em resposta livre (verificar o módulo como configurar os semicondutores todos abertos, não forçar os semicondutores superiores ou inferiores a fechar);
   - O microcontrolador deve ser capaz de determinar o tempo de desaceleração e consequente paragem da carga mecânica (após a desativação da ponte-completa, até a velocidade ser nula ou não existir variação de posição);
   - Apenas durante este processo o microcontrolador deve iniciar a amostragem ao utilizador;
   - Para facilitar o estudo e análise, é recomendável imprimir os valores de posição e velocidade a intervalos de 100 ms ao longo dos 10 s (5 s de aceleração + 5 s de resposta livre). Exportar pontos para *Excel* e analisar;
   - Finalizado o processo, a variável de *Enable* deve ser automaticamente desativada;
   - Se o utilizador ativar de novo a variável *Enable*, novo ensaio deve ser iniciado com a configuração anterior.
   - Ao sair deste modo, o sinal de *Enable* e o GPIO de ENABLE deve ser desativado de forma automática.

4) *Modo automático – controlo PID em malha fechada (obriga a ativar o pino de enable).*
   - Este modo visa o controlo do motor cc considerando o controlo PID implementado
   - Este modo após ativo, através do CS e do sinal de *Enable*, deverá permanecer em constante funcionamento independentemente de qualquer perturbação externa causada à carga mecânica;
   - Caso o utilizador desative (*Enable* =0), o sistema deverá parar e reinicializar o somatório de erros. Ativando de novo (*Enable* =1), deverá ser inicializar um novo ensaio experimental.
   - Ao sair deste modo, o sinal de *Enable* deve ser desativado.

Comandos a configurar:

Para a implementação dos estados de funcionamento, os comandos a serem implementados devem seguir a seguinte estrutura:

a) CS – *Control System*: define o estado de funcionamento a aplicar:

Control system: <char> += "CS β <dig>$^{+}$"

i. <dig> = 0 para *Modo de reset*;
ii. <dig> = 1 para *Modo de configuração*;
iii. <dig> = 2 para *Modo de controlo manual*;
iv. <dig> = 3 para *Modo de ensaio experimental em malha aberta*;

v. <dig> = 4 para *Modo automático*.

NOTA: Para a seleção do modo automático, o comando deve ser "CS 4". Este comando pode ser manipulado em qualquer estado.

b) EN – *Enable*: ativa o controlo dos motores. Isto é, ativa o pino de ENABLE que ativa o funcionamento da ponte-completa (modo manual, ensaio experimental e automático):

Enable: <char> += "EN β <dig>$^{+}$"

i. <dig> = 0 desativa o controlo e o pino de ENABLE
ii. <dig> = 1 ativa o controlo do estado previamente selecionado e o pino de ENABLE.

NOTA: Para a ativação do sinal de PWM, o comando deve ser "EN 1". Este comando pode ser manipulado em qualquer estado.

c) HW – *Período de amostragem*: configura o período de amostragem para o cálculo do controlador PID, considerando como base 1 ms

Período de amostragem: <char> += "HW β <units>$^{+}$"

i. <units> deve apresentar um valor compreendido entre 1 e 1000, apresentando assim um período de amostragem de 1 ms a 1 s, respetivamente;

NOTA: Para a configuração de um período de amostragem de 20 ms, o comando deve ser "HW 20". Este comando <u>apenas</u> pode ser manipulado no *estado 1) Modo de configuração*.

d) CR – *Continuous refresh* (Amostragem contínua ao utilizador): configura o método de amostragem (velocidade e/ou posição) como sendo contínua ou limitada.

Continuous refresh: <char> += "CR β <dig>$^{+}$"

i. <dig> = 0 desativa a amostragem;
ii. <dig> = 1 configura o método de amostragem como sendo uma amostra contínua, apresentando apenas uma amostra a cada segundo.
iii. <dig> = 2 configura o método de amostragem como sendo uma amostragem limitada, apresentando apenas **KA** amostras considerando o período definido por **HW**. No final de **KA** amostras, a amostragem deve parar até novo comando. Dica: este método pode originar problemas na porta de série dependendo da sua configuração, sendo recomendado a utilização de um buffer antes de imprimir.

NOTA: Para a configuração de amostragem contínua, uma amostra por segundo, o comando deve ser "CR 1". Este comando <u>apenas</u> pode ser manipulado no *estado 1) Modo de configuração*.

O que mostrar ao docente:

- Mostrar os estados de configuração (0 a 4) implementados;
- Mostrar as funções em cada estado (maioria **<u>ainda</u>** irão implementar nos objetivos seguintes, mas identificar já as funções previstas);
- Mostrar como validam o comando para a manipulação de variáveis/modos de funcionamento (comando correto *vs* comando errado);

- Demonstrar o correto funcionamento e transição entre estados (*print* dos estados e informação relevante);
- Demonstrar que a manipulação de alguns comandos em alguns estados é limitada e dá *feedback* ao utilizador de comando inválido (comando errado) ou comando bloqueado (comando bloqueado do estado de operação).

2.3 Objetivo 2 - Implementar a medida de posição e velocidade

Preparação de aula: devem apresentar a algoritmia (um esboço) dos principais algoritmos a serem implementados (por exemplo, leitura do sentido de velocidade, converter o número de pulsos em velocidade e posição angular).

Para implementar a medida de posição e de velocidade usar-se-á o método de **contagem de pulsos** (ISR_P) e o método de **contagem de tempo** entre pulsos (ISR_T), como referido no guia teórico. Para tal, o sinal de *feedback* do motor (Sensor A) deve ser ligado de forma a provocar uma interrupção cuja ISR que corre com máxima prioridade.

A interrupção externa, originada pela transição de estado do pino de leitura do Sensor A, deverá invocar a função ISR_P, responsável por determinar o sentido de rotação (lendo o pino a que está ligada a saída do sensor B) e por gerir o método de contagem, considerando:

- Método de contagem de pulsos: incrementa (rotação no sentido horário) ou decrementa (rotação no sentido anti-horário) os contadores de contagem de pulsos (para a posição e velocidade) e retorna. Este método deve apresentar um mecanismo de proteção caso o número de rotações exceda as 3 voltas (número máximo de voltas admitido para o contador de posição – nesse caso reinicializa o contador);
- Método de contagem de tempo: a ISR_P para o temporizador que está a medir o tempo desde a última interrupção, guarda o valor do tempo contado para processamento pela ISR_T e dispara de novo o temporizador.

Para a execução deste objetivo deve iniciar pela implementação da medição da velocidade. Para esta medição, é necessária uma outra ISR, chamemos-lhe ISR_T, que amostrará os valores estabelecidos pela ISR_P. Para tal, deverá ser programado um temporizador que com um período de HW segundos (período de amostragem da velocidade) gera uma interrupção que invoca a ISR_T. Esta adquire o valor do contador de pulsos da velocidade e coloca a 0 o contador de pulsos da velocidade. O contador para a velocidade, *inc_speed*, e posição, *inc_pos*, devem ser diferentes, isto porque o da posição apenas deve ser reinicializado em caso de ser solicitada a reinicialização. Para a velocidade, se o valor do contador for inferior a 2 (ou outro limite, *lm_res*, escolhido de resolução inferior do método de contagem de pulsos) adquire o último valor do tempo contado pela ISR_T e calcula o correspondente valor em frequência.

A ISR_P pode também fazer a gestão de inibir ou não o método de contagem de tempo. Para isso, defina-se um valor limite, *lm_en*, de ativação ou inibição de pulsos contados (*lm_en* será superior a *lm_res*). Se a ISR_P determinar que o número de pulsos contados é superior (ultrapassou) a *lm_en*, fará o *reset* de um *bit* que inibe a contagem de tempo. Se determinar que o número de pulsos contados é inferior a (abaixo de) *lm_en*, fará o *set* de um *bit* que permite a contagem de tempo. Devem empiricamente, experimentalmente, determinar o melhor valor de *lm_en* de forma a obter a melhor solução.

Comandos a configurar:

Para o objetivo 2 pode enumerar-se os seguintes comandos de configuração:

a) L – *Leitura*: define o tipo de leitura a ser realizada:

Leitura: <char> += "L β <dig>$^{+}$"

i. <dig> = 0 para leitura apenas de posição;
ii. <dig> = 1 para leitura apenas de velocidade;
iii. <dig> = 2 para leitura de posição e velocidade;

NOTA: Para imprimir o valor da velocidade e da posição ao mesmo tempo, o comando deve ser "L 2". Este comando apenas pode ser manipulado no *estado 1) Modo de configuração*.

No arranjo acima, os valores de velocidade angular são calculados o mais depressa possível, sendo representados como número de pulsos por período de amostragem. Para envio pela porta série, o utilizador deve enviar o valor no SI, isto é, rad/s. Contudo, uma vez que o fabricante apresenta os valores nominais em rotações por minuto (RPM), talvez uma unidade mais familiar para alguns utilizadores, o valor em RPM deve ser igualmente apresentado.

Para o cálculo da medida da posição angular, apenas é necessário aceder ao contador do método da contagem de pulsos. Posteriormente, na secção II, este contador será convertido para a posição angular em rad.

b) KA – *número de amostras*: define o número de amostras de velocidade/posição a um período definido por HW:

Número de Amostras: <char> += "KA β <units>"

i. KA pode tomar um valor compreendido entre 0 e 10000;

NOTA: De forma a obter 250 amostras, o comando deve ser "KA 250". Este comando apenas pode ser manipulado no *estado 1) Modo de configuração*. IMPORTANTE: o envio de amostras a um período semelhante ao definido por HW, pode originar entupimento do canal de comunicação (HW baixo, pode originar uma frequência demasiado alta para um *braud rate* baixo previamente configurado para a porta de série, ou muita informação enviada pelo *terminal*).

c) IPOS – *Reinicialização da posição 0 do disco*: define a posição atual como a posição *zero* do disco:

IPOS: <char> += "IPOS"

i. O comando IPOS define a posição atual da carga mecânica como posição zero;

Nota: Este comando pode ser manipulado em qualquer estado.

Metodologia de validação:

Para validar a algoritmia implementada da leitura de velocidade angular, deve seguir o seguinte procedimento:

a) Conectar a saída do gerador de sinais ao periférico de leitura do sensor A. O gerador de sinais deve ser configurado para um valor conhecido (representativo de uma dada velocidade) e ajustado aos limites de operação do periférico (ter cuidado com amplitudes e valor médio, só depois ligar o STM32). Por sua vez, o periférico do

microcontrolador deve ser configurado para provocar uma interrupção no modo ascendente do pino;

b) Validar a interface com o utilizador, configurando os registos necessários e analisando os valores obtidos.
c) Realizar um *toggle* a um pino de forma a demonstrar a frequência de amostragem dos dados.

Para validar a algoritmia implementada da leitura de posição angular, deve seguir o seguinte procedimento:

a) Conectar as saídas do *encoder* do motor ao periférico de leitura do sensor A e do sensor B. **Com os enrolamentos do motor em aberto (sem ponte-completa)**, devem mexer manualmente o disco, alterando a sua posição. O movimento deve ser suave e não brusco, sendo que o som das engrenagens não deve evidenciar qualquer anomalia mecânica;
b) Validar a interface com o utilizador, configurando os registos necessários e analisando os valores obtidos;
c) Reinicializar o contador da posição do disco via terminal, verificando o seu bom funcionamento.

Nota: Não esquecer do limite máximo de contagem de voltas de 3 (para ambos os sentidos).

O que mostrar ao docente:

- Demonstrar a correta configuração dos comandos (atualização dos registos é correta?);
- Demonstrar a correta descodificação da velocidade (auxílio do gerador de sinais para diferentes velocidades);
- Demonstrar a correta descodificação da posição (rodar a carga mecânica e mostrar a posição).
- Demonstrar a correta reinicialização da posição do disco, demonstrar o passo anterior de novo.

2.4 Objetivo 3 – Implementação do comando para um canal PWM bipolar e acionamento do motor

Preparação de aula: devem apresentar um esboço da algoritmia dos principais algoritmos a implementar. Devem fazer um estudo de como realizar a configuração do PWM no STM32.

Existem diversos parâmetros que são predefinidos por compilação e não alteráveis em funcionamento, a tensão CC da fonte de alimentação para efeitos de proteção (a tensão nominal do motor é 6 V), o pino de PWM a utilizar, o pino de ativação/desativação (ENABLE) e o pino da direção (*forward/reverse* - FW/RV). **Nota: estes pinos podem ser diferentes dependendo do módulo adquirido da ponte-completa, devendo ajustar a função lógica de *enable* e de controlo de direção ao respetivo módulo.**

O procedimento a adotar neste objetivo de trabalho, numa fase inicial sem a utilização da maqueta, será:

a) FW – *Switching frequency* (frequência de comutação): define o valor múltiplo da frequência do sinal de PWM, considerando como base 100 Hz. Isto é, enviando 1 a frequência é definida em 100 Hz, enviando 5 a frequência é definida em 500 Hz, enviando 50 a frequência é definida em 5 000 Hz:

   Switching frequency: <char> += "FW β <signval>$^{+}$"

   i. <signval> valor compreendido entre 1 (configurando como 100 Hz) e 100 (configurando 10 000 Hz).

   NOTA: Para um valor de frequência de 100 Hz, o comando deve ser "FW 1". Para um valor de frequência de 2 000 Hz, o comando deve ser "FW 20". Este comando <u>apenas</u> pode ser manipulado no *estado 1) Modo de configuração*.

b) UN – *Tensão Normalizada*: define o valor da tensão normalizada (*duty-cycle*) a aplicar:

   Tensão Normalizada: <char> += "UN β <signval>$^{+}$"

   ii. <signval> valor compreendido entre +100 (máx. FW) e –100 (máx. RV), com incrementos de 1. O valor absoluto indica a percentagem de PWM (ou de *duty cycle*) e o sinal indica o sentido da tensão aplicada à carga (pino FW/RV);
   iii. A tensão média aplicada ao motor igualará a tensão da fonte vezes a Tensão Normalizada a dividir por *cem* (V_{cc}*<signval>/100);
   iv. <signval> deve ler-se como <sign><dig>$^{+}$.

   NOTA: Para um valor de *duty-cycle* de 50% no sentido *reverse*, o comando deve ser "UN -50". Para um valor de *duty-cycle* de 25% no sentido *forward*, o comando deve ser "UN +25". Este comando pode ser manipulado no *estado 1) Modo de configuração* e *estado 2) Modo Controlo Manual*.

c) Deverão ser implementados dois novos carateres de controlo para que, de uma forma expedita, seja possível incrementar e decrementar o valor da tensão normalizada (sem necessidade de enviar o ENTER). Sugere-se para isso o carácter '/' para produzir um aumento de 5 unidades e o carácter '\' para reduzir 5 unidades. Verificar que a mudança do quadrante de operação é suave e gradual (por exemplo, com uma mudança gradual

do valor de *duty-cycle*: 10%; 5%; 0%; -5%; -10%. **Evitar: 10%; 5%; 0%; -95%; -90%). Este comando apenas pode ser manipulado no *estado 2) Modo Controlo Manual*.**

Metodologia de validação:

- Comprovar o correto funcionamento do sinal de PWM com auxílio de um osciloscópio. Podem considerar uma frequência de 1 kHz.
- Testar o software desenvolvido utilizando o modo *debug* para o efeito, verificando a correta atualização das variáveis de controlo. Com auxílio de um osciloscópio, comprovar o correto funcionamento do sinal de PWM (*duty-cycle* e frequência).

Resultados experimentais:

Depois de implementado o comando de PWM segue-se o acionamento do motor com o PWM. Para isso é necessário perceber qual a influência da frequência de comutação no comportamento do motor. Assim, para diferentes valores de frequência de comutação, devem registar o valor de *duty-cycle* mínimo de forma a obter uma rotação contínua da carga (sem parar) por um período mínimo de 10 segundos. Os alunos podem considerar o exemplo da Tabela 1, registando também o valor médio da corrente da máquina CC. **Verificar se o módulo da ponte-completa do grupo permite operar com as frequências da Tabela 1. Considerar uma tensão de alimentação na entrada da ponte-completa de 7,5 V.**

Tabela 1 – Valores de *duty-cycle* para o arranque da máquina elétrica CC e corrente média em regime permanente (apenas no sentido FW).

Frequência (Hz)	*Duty-Cycle* (período *On*)	Corrente Média (mA)
100	_____ % (_____ ms)	
200		
500		
1 000		
2 000		
5 000		
10 000		

Analisar os resultados obtidos, escolhendo a frequência de comutação que apresenta melhores resultados para os ensaios experimentais seguintes. Com o conversor de eletrónica de potência, manipular, **com cuidado**, o valor de *duty cycle* do sinal PWM, de uma forma incremental, até que este atinja os 100 % (velocidade máxima de rotação do motor). Após ser atingida a velocidade máxima num dado sentido, enviar o comando de paragem do motor ('UN 0') e observar/anotar o comportamento do sistema. Quando este estiver parado, efetuar o mesmo procedimento com o PWM negativo, decrementando com a tecla de controlo até um valor de *duty cycle* de "–100 %". Deverá ser observado e registado qual o valor de PWM necessário até se obter o movimento do veio (em ambos os sentidos). Registar, de igual modo, o tempo que demora desde o instante de repouso até ao instante que é atingida a velocidade nominal (utilizando um temporizador, os sinais provenientes do *encoder* e o instante que é realizado o *disable*, podem utilizar uma função para uma maior precisão). Se tiverem o periférico do DAC podem imprimir esta rampa de aceleração e desaceleração, tendo em consideração as unidades rpm/div.

Registar os valores de velocidade, corrente e tensão medidos para cada 10 % de PWM (de –100 % a +100 %). Desenvolver um gráfico de linhas com os dados adquiridos (*duty cycle vs* velocidade), apresentando o gráfico ao docente.

O que mostrar ao docente:

- Apresentar a Tabela 1 preenchida, justificando a escolha da frequência;
- Apresentador os dados obtidos (velocidade, corrente e tensão) para a frequência escolhida para diferentes valores de *duty-cycle*;
- Apresentar tempos de desaceleração e conclusões (texto posterior à Tabela 1);

Algumas questões para análise:

- Com o osciloscópio do laboratório, pode medir-se a tensão aos terminais da máquina CC e do sinal de PWM à entrada do módulo da ponte-completa? Justificar
- Aplicar diferentes valores de *duty-cycle* e analisar as formas de onda aos terminais da máquina CC. Identificar os estados de operação da ponte-completa e da máquina CC.
- A tensão aplicada ao motor apresenta uma amplitude igual à tensão de entrada da ponte-completa? Porquê?

NOTA: Qualquer aluno está proibido de pegar e guardar a maqueta dos armários. O docente deve entregar a cada aluno, averiguando o correto funcionamento da mesma. No final, o docente deve guardar ou verificar que as maquetas estão corretamente organizadas.

O aluno: devem verificar primeiro o correto acoplamento do eixo do motor ao veio mecânico do sistema. Em caso de folga, proceder ao ajuste do mesmo, tal como demonstrado no tópico "*4. Procedimentos para a adição/alteração de carga mecânica ao/do sistema de acionamento*" existente no guia de apoio teórico à CP2. Certificar que o veio mecânico roda livremente, sem apresentar atrito mecânico ou demasiada folga. Durante os ensaios experimentais devem verificar periodicamente a temperatura do motor, não devendo colocar o sistema em excesso de carga durante longos períodos de tempo com risco de danificar o sistema de engrenagem. De forma regular, devem verificar se o motor está em sobreaquecimento, desligando o circuito em caso positivo. Em caso de aplicar uma tensão excessiva no motor ou danificar as engrenagens, reflete-se numa penalização da nota em 40% em todos os objetivos do guia, ficando responsáveis pela correta reparação da maqueta.

Secção II – Módulos de controlo

2.5 Objetivo 4 – Análise e determinação dos parâmetros do sistema a controlar

Este objetivo visa a análise da modelação do sistema implementado (visualização da subsecção 3.2 do guia teórico) – modelação em espaço de estados - e cálculo experimental dos parâmetros do sistema mecânico apresentado na Figura 1. Para realizar este objetivo será necessário proceder à leitura do Guia Teórico (Secção 3.2 e 3.2.1, realizando na maqueta, as experiências indicadas para determinar os parâmetros do motor e da fricção na maqueta atribuída ao grupo (deve ser sempre a mesma). Os valores nominais dos parâmetros são:

$J = 0{,}02$ kg/m^2
$B = 0{,}01$ N/(m/s)
$K_m = 0{,}265$ N/A
$R = 2{,}5\ \Omega$
$K_w = 1$

Estes têm de ser adaptados aos valores obtidos na maqueta atribuída a cada grupo. O momento de inércia do sistema mecânico (J) não vai ser calculado, devendo manter-se idêntico ao nominal, 0,02 Kg/m^2, e o ganho do PWM (K_w) vai ser considerado unitário. Todos os objetivos que realizarão posteriormente devem ser sempre realizados com a mesma maqueta, caso contrário, podem existir disparidades entre resultados de simulação e reais.

O que mostrar ao docente:

- Apresentar qual foi o procedimento experimental realizado para obter os valores de K_m e B, indicando quais os valores obtidos para R, i_a e ω, para o cálculo de K_m, e m_m e ω_{SS} para B.

2.6 Objetivo 5 – Controlo discreto em malha aberta do sistema e validação dos parâmetros calculados

Modelizado o sistema, e tendo já calculados os parâmetros do sistema, já é possível analisar o comportamento do sistema em malha aberta.

Em simulação, é necessário primeiro implementar o programa indicado na subsecção 3.3 do guia teórico (*Análise do sistema/processo em malha aberta – modelo contínuo)* e correr as duas simulações do controlo em malha aberta que permite obter a resposta livre e resposta forçada do sistema (alterar os parâmetros do sistema do script Matlab para os valores obtidos na experiência anterior). Tal como indicado no guia teórico, inicialmente é necessário correr o script Matlab (*define_variables_continuous.m*) e só depois o ficheiro Simulink (*continuous_system.slx*). Inicialmente, para proceder à análise da resposta livre, deve ser aplicada uma velocidade angular inicial igual a 1 rad/s, posição angular inicial igual a 0 rad, não aplicando tensão ao motor, nem qualquer binário de perturbação, e proceder à análise da resposta livre durante 5 s. Para análise da resposta forçada deve ser aplicada uma tensão ao motor de 6V, sem qualquer condição inicial, ou binário de perturbação, correndo a simulação durante 5 s. Correndo o programa é possível verificar, que a informação dos polos do sistema também é disponibilizada, sendo útil para a determinação do período de amostragem a utilizar.

Seguidamente. é necessário proceder ao cálculo do período de amostragem (*h*), considerando a resposta do sistema e os polos (ver subsecção 3.4 do guia teórico).

Na secção 3.5 é descrito como se passa do modelo de espaço de estados continuo para discreto. Esse novo sistema discreto será usado para validação dos parâmetros do sistema. Assim, é necessário realizar a simulação apresentada nessa secção para parâmetros da maquete do grupo, mas aplicando uma tensão de 4,5 V, durante um período de 5 s, e visualizando os valores de posição e velocidade durante 10 s (primeiro correr e adaptar os parâmetros do script Matlab *define_variables_discrete.m* e depois correr o ficheiro Simulink *discrete_system.xls)*.

Pretende-se agora realizar exatamente a mesma experiência com a maqueta real, isto é, aplicando uma tensão de 4,5 V ao motor (PWM de 75%) durante 5 s e monitorizar os valores de velocidade e posição do disco durante 10s (uma leitura de posição e velocidade a cada 100 ms). Para isso, é necessário selecionar como estado de funcionamento do programa principal o *Modo de Ensaio Experimental*, criando uma função de controlo em malha aberta durante o tempo definido. No final, é necessário proceder a uma análise crítica dos resultados obtidos em simulação e na maqueta real.

O que mostrar ao docente:

- Deve ser apresentado o resultado obtido em simulação da resposta livre e resposta forçada para o modelo contínuo;
- Indicar também a localização dos polos do sistema, explicando qual o procedimento realizado para determinar o período de amostragem;
- Apresentar os resultados obtidos em simulação para a experiência de aplicação durante 5 s de uma tensão de 4,5 V e depois visualização da resposta livre durante mais 5 s;
- Apresentar em excel os resultados obtidos na experiência realizada na maquete para as mesmas condições que a realizada em simulação. Deve proceder a uma análise crítica dos resultados obtidos, e compare-os com os resultados de simulação.

2.7 Objetivo 6 – Determinação, em simulação, dos parâmetros do controlador PID

Com o modelo do sistema discreto obtido nos objetivos anteriores, é possível modelizar o sistema em malha fechada e determinar os parâmetros do controlador PID em ambiente de simulação, no Matlab. Para tal será necessário seguir o Guia Teórico (Secção 3.6), realizando os seguintes procedimentos (não esquecer de usar os vossos parâmetros da maqueta, e fazer uso do script Matlab *define_variables_pid.m* e do ficheiro Simulink *pid_system.xls*):

i. Determinar o ganho proporcional para o modelo;
ii. Verificar a resposta do sistema a uma perturbação externa (aplicar em *m_d_value=0.06* no script Matlab).
iii. Determinar o ganho integral do modelo, seguindo as regras do método Ziegler–Nichols;
iv. Verificar novamente a resposta do sistema a uma perturbação externa;
v. Determinar o ganho derivativo do modelo, seguindo as regras do método Ziegler--Nichols.
vi. Identificar as vantagens e desvantagem da componente derivativa do controlador PID.

Deste estudo devem resultar os parâmetros do controlador PID.

O que mostrar ao docente:

- Deve ser apresentada a forma como foi obtido o valor de K_p para o controlo apenas proporcional;
- Mostrar o comportamento do sistema ao controlo proporcional sem e com a perturbação indicada;
- Apresentar o procedimento realizado para obter o ganho proporcional e integral para o controlador PI;
- Mostrar o comportamento do sistema ao controlo proporcional-integral (PI) sem e com a perturbação indicada;
- Identificar as vantagens e limitações da componente integral no controlador PI;
- Apresentar o procedimento realizado para obter o ganho proporcional, integral e derivativo para o controlador PID;
- Mostrar o comportamento do sistema ao controlo proporcional-integral-derivativo (PI) sem e com a perturbação;
- Identificar as vantagens e limitações da componente derivativa no controlador PID.

2.8 Objetivo 7 - Implementação do algoritmo do controlador PID e validação da algoritmia implementada

Este objetivo é constituído por três partes, que devem ser realizadas de forma sequencial.

Parte 1: Implementação de comandos

Para ser possível realizar o controlo do sistema, é necessário criar comandos que possibilitem definir os parâmetros do controlador PID e os valores das variáveis, nomeadamente:

- Variável de referência y_r,
- Ganho proporcional K_p,
- Ganho integral K_i,
- Ganho derivativo K_d,
- Constante do filtro passa-baixo na ação derivativa, a.

A definição do período de amostragem para o controlador PID já tinha sido realizada no objetivo 1.

Assim, na interface do utilizador, devem implementar os seguintes comandos, de uso exclusivo no modo configuração.

Definir o valor da posição de referência em rad: <char>+="PIDyr β <float>+"

i. <float> valor indicado para a posição desejada para o disco, em radianos, podendo ser um valor positivo ou negativo e que esteja compreendido entre o valor em radianos para 3 voltas para a frente ou para trás.

Definir ganho proporcional do controlador PID: <char>+="PIDkp β <float>+

i. <float> valor indicado para o ganho proporcional do controlador PID.

Definir ganho integral do controlador PID: <char>+="PIDki β <float>+"

i. <float> valor indicado para o ganho integral do controlador PID.

Definir ganho derivativo do controlador PID: <char>+="PIDkd β float>+"

i. <float> valor indicado para o ganho derivativo do controlador PID.

Definir constante do filtro passa-baixo na ação derivativa do controlador PID: <char>+="PIDa β <float>+"

i. <float> valor indicado para a constante do filtro passa-baixo da ação derivativa do controlador PID.

Para realizar uma interface segura em termos de operação do sistema de controlo, deve ser colocado, por defeito, todos os parâmetros com o valor 0 e definir valores positivos para o período de amostragem (h). Os valores devem estar em unidades do SI. A interface deve limitar os intervalos de valores dos parâmetros que o utilizador pode escolher. O limite inferior para o intervalo de valores do período de amostragem (h) fica definido pelo que é realizável pelo *hardware / software*. Este período mínimo deve ter em conta a restrição de a soma do tempo de execução da rotina de interrupção do PID com os tempos de execução de outras rotinas de interrupção necessárias ao controlo ter de ser inferior ao período de amostragem, de forma a permitir a execução do programa principal.

Parte 2: Implementação do algoritmo PID

Este objetivo visa a implementação do controlador digital PID no microcontrolador para calculo da variável de comando $u(k)$. A variável controlada será a posição angular da carga rotativa. Os valores da variável de referência são indicados através da interface, considerando o sistema de unidades SI, como implementado na Parte 1. Para obter os valores da variável medida é necessário utilizar as funções de cálculo da posição angular do motor, realizado a partir do sinal em quadratura do *encoder* (utilizando o algoritmo do Objetivo 2). Na programação do controlo pode utilizar os parâmetros do controlador PID encontrados em simulação, e ajustar ao protótipo real conforme necessário. Por último, deve integrar a variável de comando na atuação PWM (utilizando o algoritmo do Objetivo 3).

O algoritmo digital PID é implementado como uma ISR, designada por ISR_PID, invocada com período h (definido na interface através do comando da Secção 1). Na secção 3.7 do Guia Teórico é apresentado como a ISR_PID pode ser programada para calcular a expressão $u(k)$ de uma forma robusta. Conforme indicado no Guia Teórico, deve implementar uma variante do controlador PID: **controlador D na variável medida e PI no erro**.

Para além da programação do algoritmo digital PID, pretende-se também a implementação de funcionalidades usualmente encontradas em controladores industriais: filtro na ação derivativa, deteção de saturação e limitação da ação integral. A secção 3.7.2 e seguintes do Guia Teórico explicam, passo a passo, a implementação de cada funcionalidade até ao algoritmo completo, que se apresenta seguidamente, o qual deve ser implementado.

```
ISR_PID()
    y=input(y)
    e=yr-y
    sum_e_bkp=sum_e          //in case u becomes saturated
    sum_e=sum_e+e_ant
    u_d=Kd_h*(y-y_ant)+a*u_d_ant
    u=Kp_h*e+Ki_h*sum_e-u_d
    e_ant=e
    y_ant=y
    u_d_ant=u_d
    if u>U_sat_a then        //u above upper saturation
          u=U_sat_a
          sum_e=sum_e_bkp    //sum of errors frozen,
                             //back to last value
    else  if u<U_sat_b then  //u below lower saturation
          u=U_sat_b
          sum_e=sum_e_bkp    //sum of errors frozen,
                             //back to last value
    output(u)                //unsaturated u remains unchanged
    return from interrupt
```

Note que, atendendo às características do sistema que está a controlar, *U_sat_a* (valor de saturação superior de atuação) e *U_sat_b* (valor de saturação inferior de atuação), correspondem a 6 V e -6 V, respetivamente.

Parte 3: Validação do algoritmo PID

Nesta fase pretende-se validar a algoritmia implementada para o controlador PID digital, considerando os seguintes testes. Primeiro, é necessário testar a programação do cálculo da variável de comando $u(k)$. Segundo, comparar a resposta do sistema mecânico com e sem filtro na ação derivativa.

Para teste do algoritmo do controlador PID, é importante verificar se o cálculo do valor de comando a partir do valor da variável medida está a ser realizado de forma adequada. Para isso devem colocar o controlador em funcionamento, calculando o valor de comando $u(k)$ sem o aplicar à saída para o atuador, e usando individualmente cada uma das ações, P, I ou D (testando uma de cada vez). Deve colocar o controlador numa situação em que o sinal de erro seja conhecido. Deve fazê-lo, programando na memória do microcontrolador os valores de um sinal de tensão em onda quadrada ou triangular e considerando a variável de referência constante. Testa-se a ISR_PID fazendo-a ler os valores de y de memória. Nesta condição, o sinal de erro $e(k)$ fica bem definido e o sinal de comando $u(k)$ também, consoante for selecionada a ação P, I ou D. Os valores de entrada da onda quadrada ou triangular e os resultados obtidos para *u(k)* devem ser enviados para o terminal, e depois reproduzidos no excel. No seguimento, dão-se exemplos de sinais consistentes.

Coloque-se em memória uma onda retangular de 0 a 2 V, garantindo que esta tem valor médio 1. Ao mesmo tempo, fixa-se o valor da variável de referência de forma que corresponda ao valor médio de 1 V. Então a variável de erro $e(k)$ terá valores entre +1 V e –1 V porque $e(k) = 1 - y(k)$[1]. Como $e(k)$ tem uma evolução conhecida, pode-se determinar como deve ser a evolução de $u(k)$ para os valores de K_p, K_i e K_d estabelecidos.

Por exemplo, se o período de amostragem for de 10 ms e se a onda retangular tiver uma frequência de 5 Hz e for realizado, $K_p = 1{,}5$, $K_i = 0$, $K_d = 0$ (**apenas ação proporcional**), as evoluções de $y(k)$ e $u(k)$ deverão ter o aspeto representado na Figura 2. Note-se que as figuras seguintes *não mostram o erro*, mas sim *a entrada*.

Se fizermos $K_p = 0$, $K_i = 20$, $K_d = 0$ (**apenas ação integral**) para a mesma entrada, as evoluções de $y(k)$ e $u(k)$ deverão ter um aspeto semelhante ao representado na Figura 3, dependendo do instante preciso em que o algoritmo começa a integrar o erro.

Para se **testar a componente derivativa** deve-se colocar em memória uma onda triangular entre 0 e 2 V. Como uma onda triangular tem derivada constante, provoca na saída uma onda retangular. Se fizermos $K_p = 0$, $K_i = 0$, $K_d = 0{,}05$, (apenas ação derivativa) para uma entrada triangular (0 a 2 V) em y, as evoluções de $y(k)$ e $u(k)$ deverão ter o aspeto representado na Figura 4. Note-se que a ação derivativa deve ser afetada de sinal negativo.

[1] $1 - 0 = 1$; $1 - 2 = -1$

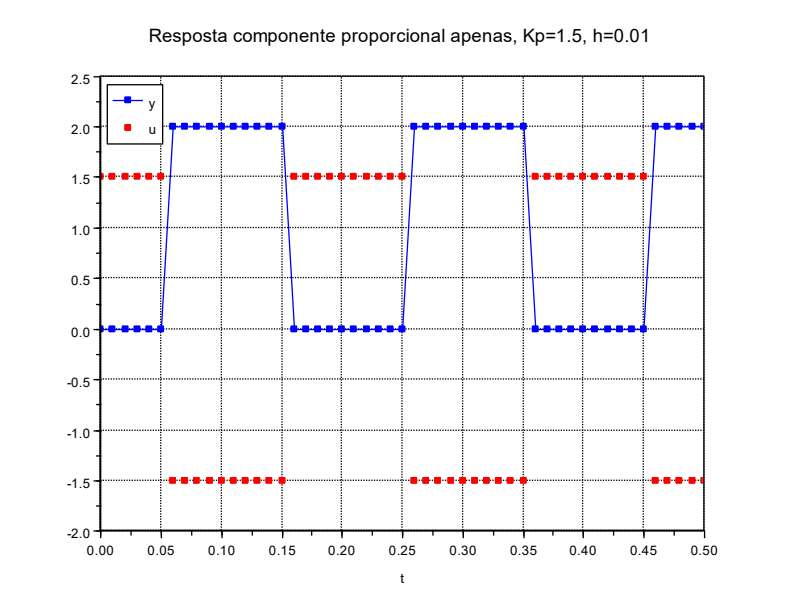

Figura 2 Resposta da componente proporcional a uma entrada que provoca um erro em onda retangular. Referência igual a 1.

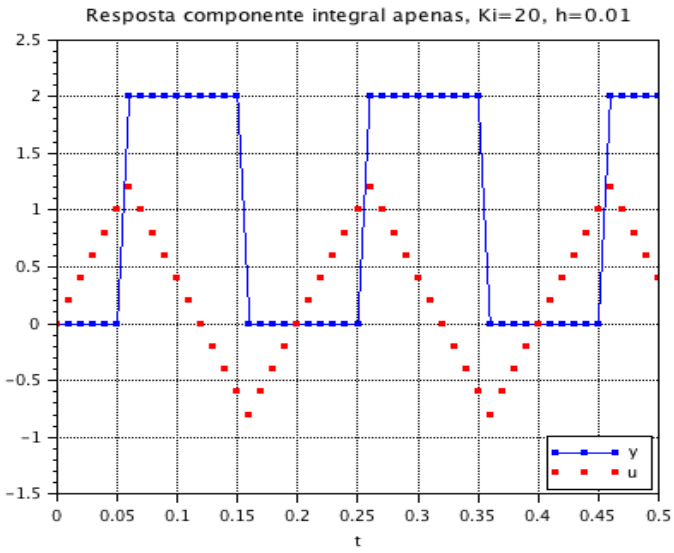

Figura 3 Resposta da componente integral a uma entrada que provoca um erro em onda retangular. As pendentes da onda triangular dependem do valor de K_i. O valor médio e desfasamento da mesma dependem do instante em que o algoritmo começa a integrar a variável de erro. Referência igual a 1.

Para o teste da ação derivativa com filtro passa-baixo, supomos a mesma onda triangular de entrada e $K_p = 0$, $K_i = 0$, $K_d = 0{,}05$, $a = 1/3$, ver Figura 5.

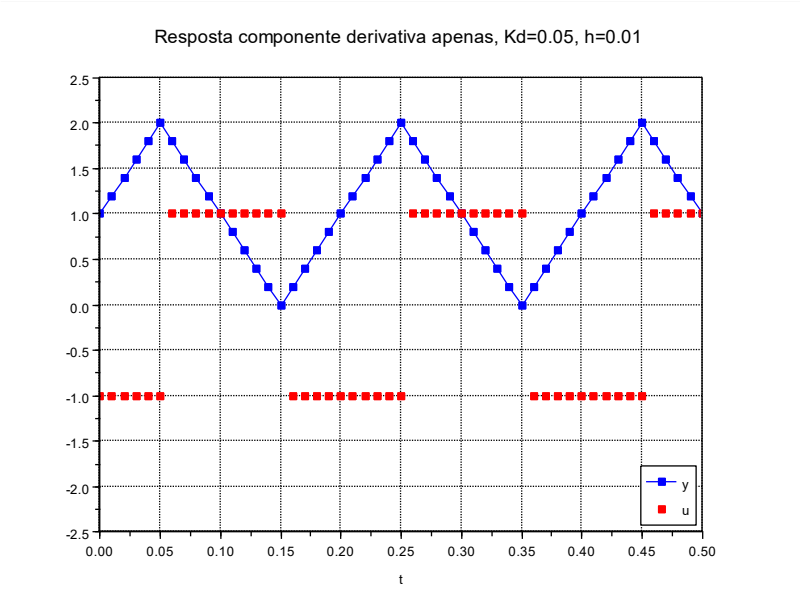

Figura 4 Resposta da componente derivativa a uma entrada em onda triangular. A derivada é uma onda retangular. Notar que a ação derivativa tem sinal simétrico da derivada.

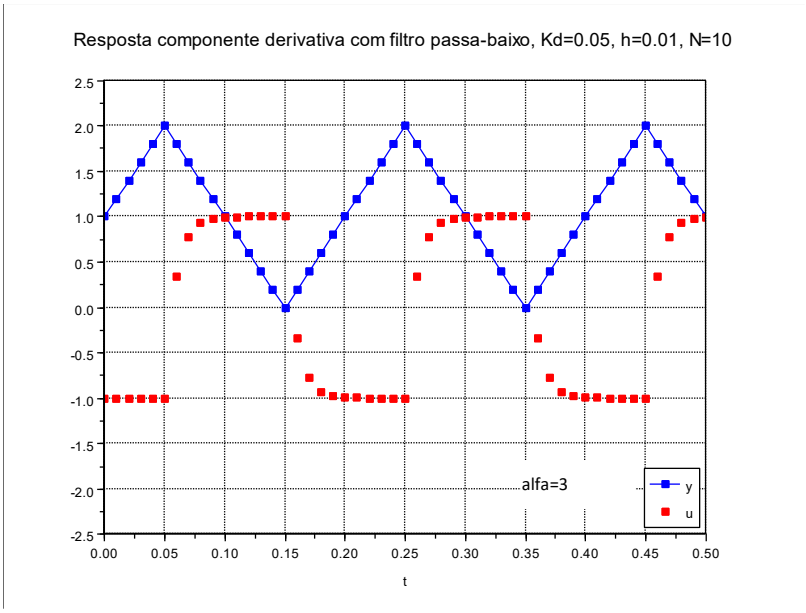

Figura 5 Resposta da componente derivativa com filtro passa-baixo a uma entrada que provoca um erro em onda triangular. A derivada é uma onda retangular com uma aproximação exponencial.

No final deve ser realizada uma verificação da ação do filtro na ação derivativa alterando o valor de a para 0, 2/3 e 1. Compare os resultados com os obtidos na Figura 5.

O que mostrar ao docente:

- Deve ser apresentada a forma como foram implementados os comandos para configuração dos valores dos ganhos do controlador PID, para a definição da posição desejada para o disco, mostrando que a sua configuração apenas é possível no modo configuração;
- Mostrar a validação realizada ao algoritmo de controlo, mostrando em primeiro lugar a resposta do controlador à entrada de um sinal retangular, e apenas ação proporcional.
- Depois mostrar a resposta do controlador aplicando apenas uma ação integral, para o mesmo sinal de entrada;
- Posteriormente mostrar o comportamento do controlador aplicando um sinal triangular e uma ação apenas derivativa, sem filtragem;
- Mostrar o comportamento do controlador para a ação derivativa e com filtragem;
- Por fim, realize uma análise critica à alteração dos valores na ação derivativa de 0, 2/3 e 1.

2.9 Objetivo 8 - Avaliação experimental do controlador

Neste objetivo pretende-se avaliar o comportamento experimental do sistema de controlo digital implementado no protótipo real representado na Figura 1. Para avaliar o sistema de controlo é necessário proceder à comparação das respostas do sistema de controlo a situações de teste experimental com as simulações das mesmas respostas usadas no modelo do sistema mecânico. Deve também avaliar o comportamento do controlador perante perturbações externas (por exemplo, perturbações manualmente aplicadas pelo utilizador).

O que mostrar ao docente:

- Mostrar o comportamento do disco da maquete à aplicação de diferentes posições desejadas para o disco, que o próprio docente poderá solicitar;
- Fazer uma análise crítica às alterações que podem ter sido realizadas aos valores das constantes do controlador PID para garantir o funcionamento adequado do sistema, explicando qual foi o procedimento realizado para a determinação desses valores;
- Mostrar o comportamento do sistema à aplicação de ações perturbadoras no disco quando se pede ao disco para movimentar-se para uma determinada posição.