



Controlador de velocidade Proporcional-Integral para motor DC utilizando funções do LabVIEW

Santos BJ*, Leão TF*§, Bock EGP*

**Instituto Federal de Educação, Ciência e Tecnologia de São Paulo – IFSP, São Paulo, Brasil.*

§*Instituto Dante Pazzanese de Cardiologia – IDPC, São Paulo, Brasil.*

Resumo. Este trabalho apresenta o desenvolvimento de um controlador de velocidade (RPM) para motor de Corrente Contínua (DC, da sigla em inglês) utilizando técnicas de Controle de Processos e funções do *LabVIEW*® (National Instruments, EUA). Um sistema atuando em malha fechada possibilita uma resposta eficaz do controlador aos distúrbios pelas perturbações e ruídos dos sinais externos a planta, pois o controlador monitora e executa atuações de reação em tempo real aos fenômenos físicos previamente convertidos em pulsos elétricos digitais ou analógicos por meio de transdutores, desta forma mantendo o sistema estabilizado na aplicação da variável controlada independente do ambiente externo nas condições de contorno especificadas, garantindo assim a homogeneidade inerentes em aplicações industriais, assim como resguardando o sistema de eventuais problemas técnicos. O controlador foi desenvolvido por meio de linguagem gráfica no *LabVIEW*, o atuador é um módulo de ponte H L298N (STMicroelectronics, Suíça) que varia a tensão de saída proporcionalmente a porcentagem de *Duty Circle* PWM da Entrada *Enable*, o L298N foi acoplado eletricamente a um motor EC 353518 (Maxon, Suíça) que varia a velocidade (RPM) linearmente à tensão fornecida pelo L298N e o transdutor é um sensor *encoder* rotacional KY-040 (Keyes, EUA), que converte a rotação do rotor do motor em pulsos elétricos digitais, podendo assim calcular a velocidade do motor por uma quantidade exata de pulsos. A leitura da velocidade do motor foi satisfatória e o controle PI da velocidade apresentou erro em regime estacionário menor que 2%.

Palavras-chave. *Controlador, Motor DC, Controle PI, LabVIEW.*

Introdução. O controle sobre um sistema significa a atuação de um conjunto de componentes em prol da execução de um determinado objetivo de forma eficiente. O uso de técnicas de controle baseia-se em teorias de linearidades, onde é inferida uma relação direta de causa e efeito entre os componentes do sistema, desta forma, podendo ser previsto e ajustado a relação de cada componente do sistema com a variável controlada (Objetivo). Um sistema pode ser representado de forma simplificada como mostrado na Figura 1.



Figura 1. Representação simplificada de um Sistema.

Um sistema pode ser controlado de duas maneiras: Malha Aberta, onde um controlador com um determinado objetivo atua diretamente sobre o processo, assumindo que tal objetivo estará sempre sendo concretizado na saída, simplificado na Figura 2; Malha Fechada, onde o controlador atua sobre o processo com base na atuação anterior, monitorando a saída realimentada por meio de um sensor, garantido assim que o objetivo estará sendo concretizado de forma efetiva e estável na saída (1), simplificado na Figura 3.

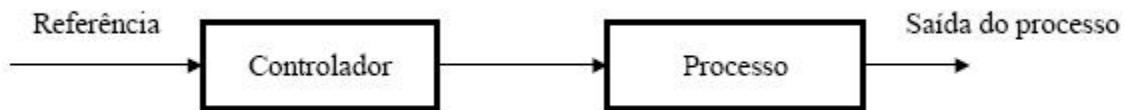


Figura 2. Representação simplificada de um Sistema controlado em Malha Aberta.

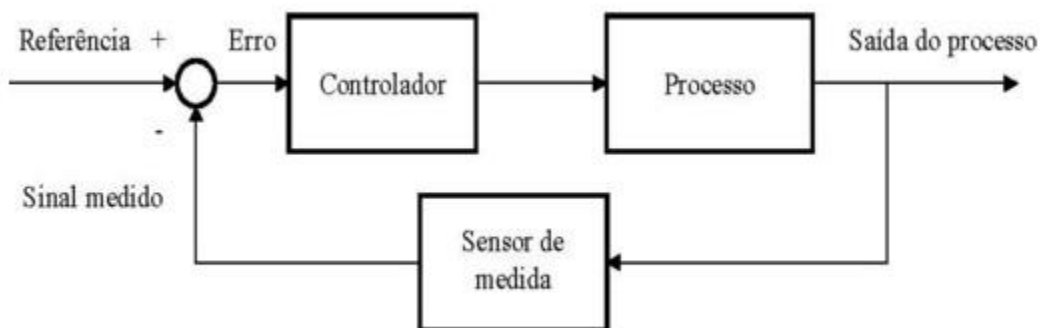


Figura 3. Representação simplificada de um Sistema controlado em Malha Fechada.

Para um sistema ser controlado em malha fechada se faz necessário o uso de transdutores, responsáveis por converter fenômenos físicos em pulsos elétricos digitais (Geralmente, 0 Volts ou 5 volts) ou analógicos (Geralmente, valores entre 0 Volts e 5 volts), desta forma o controlador



realiza o monitoramento dos sinais externos ao sistema, que junto a uma tabela de comparação, possibilita a correlação de uma atuação proporcional objetivando induzir o processo na saída desejada.

O sistema de controle em Malha Fechada deste trabalho é constituído por: Controlador, responsável por induzir a variável controlada Velocidade (RPM) escolhido na entrada da Interface Homem-Maquina (IHM) na saída do rotor do motor, O controlador foi desenvolvido por meio de linguagem gráfica na plataforma *LabVIEW (Laboratory Virtual Instrument Engineering Workbench, acrónimo do inglês)*, que é um software de engenharia criado para aplicações que requerem teste, medição e controle, com rápido acesso ao hardware e a informações obtidas a partir dos dados (2), instalado em um PXI -8840 com módulo DAQ PCIE – 6361; O atuador escolhido, o motor EC 353518 (Maxon, Suíça)(3) que varia a velocidade (RPM) linearmente à tensão de entrada, não pode ser eletricamente acessado pelo controlador, pois a tensão e corrente de trabalho do motor exerce a fornecida pelo controlador, devido a isso foi necessário a adição de um atuador de transição entre os dois, responsável por atuar comando de Tensão/RPM do Controlador no atuador efetivo motor EC 353518, para tal foi escolhido o módulo de ponte H L298N (STMicroelectronics, Suíça) (4); Para a leitura de velocidade do motor e realimentação do sistema, o transdutor primeiramente escolhido, foi o sensor Hall KY-024 (Keyes, EUA)(5), onde com um ímã acoplado ao rotor do motor, uma vez a cada volta completa o sensor captaria a presença de um campo magnético e enviando “1 Digital” (5 Volts) para o controlador e assim incrementando o contador de voltas, o sensor KY-024 foi substituído pelo sensor *encoder* rotacional KY-040 (Keyes, EUA)(6), devido ao contador de RPM do controlador para o sensor Hall ser uma relação da somatória de voltas do rotor em 1 segundo multiplicado por 60 (segundos para minuto), com essa resolução (1:1) a taxa de erro poderia ser muito alta, com o sensor *encoder* KY-040, que converte a rotação do rotor do motor em pulsos elétricos digitais, podendo assim calcular a velocidade do motor por uma quantidade exata de pulsos, sendo 20 pulsos por volta (Resolução 1:20), o contador de RPM do controlador se tornou uma relação da somatória de voltas do rotor em 1 segundo dividido por 20 (pulsos) e multiplicado por 60 (segundos para minuto), ou seja a somatória de voltas do rotor em 1 segundo multiplicado por 3, diminuindo consideravelmente a possível taxa de erro.

Material e Métodos. Os programas em *LabVIEW* são compostos por painel frontal, que contém a interface, e por diagrama de blocos, que contém o código gráfico do programa. A programação é feita de acordo com o modelo de fluxo de dados através de ligações entre os blocos de funções, que são instrumentos virtuais (*VIs*, sigla em inglês) que podem ser utilizados em conjuntos (*Sub-VI*) ou isolados, possuindo entradas e/ou saídas. A execução de um *IV* começa quando todas as entradas estão disponíveis; os resultados do processamento são então colocados nas saídas assim que a execução do sub-programa tenha terminado. Desta forma, a ordem pela qual as tarefas são executadas é definida em função dos dados. Uma ordem pré-definida (por exemplo, "da esquerda para a direita") não existe. Uma importante consequência destas regras é a facilidade com que podem ser criados processos paralelos no *LabVIEW*. Os *sub-VIs* sem interdependência dos respectivos dados são processados em paralelo. Os *sub-IVs* que não possuem entradas são

executados no início do programa. Se o *sub-VI* não possuir saídas, os dados resultantes são ignorados ou, então, usados pelo exterior: são escritos para o disco rígido ou para a rede, ou enviados para impressão. Da mesma forma, um *sub-VI* sem entradas pode receber dados provenientes de aparelhos periféricos ou pode gerar os seus próprios dados (7). Na Figura 4 é demonstrado o diagrama de blocos do controlador desenvolvido no *LabVIEW*, constando marcações numéricas e respectivas explicações abaixo.

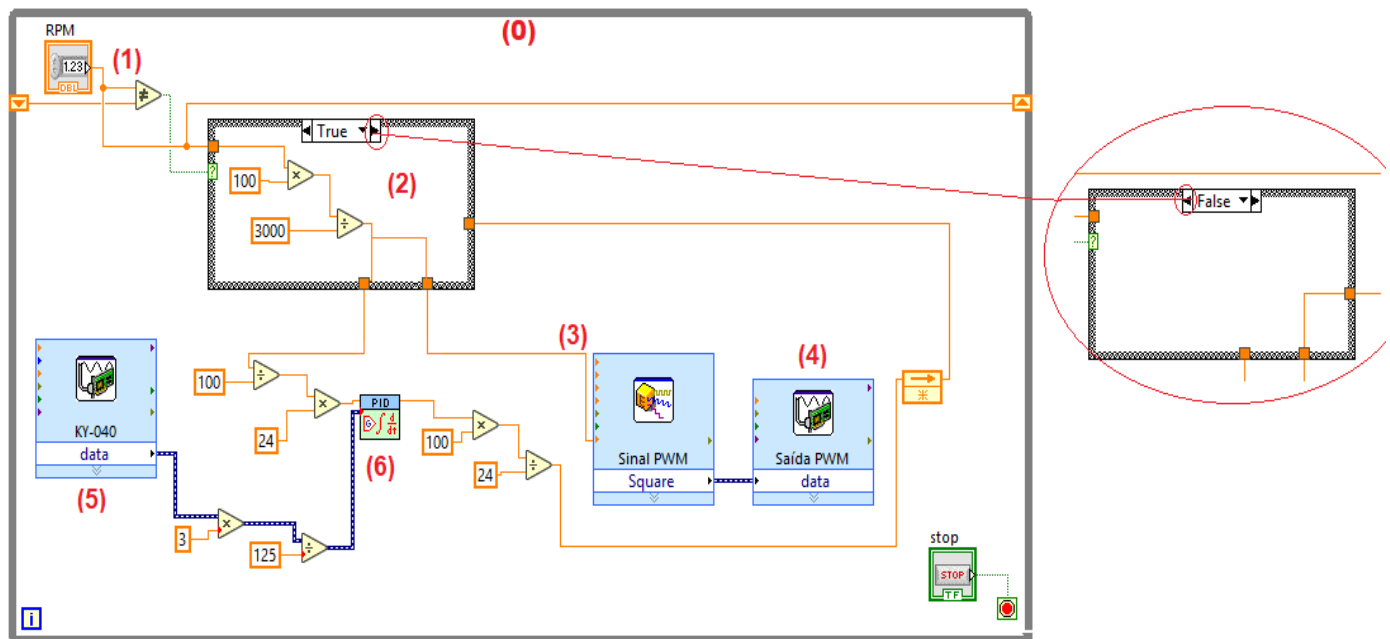


Figura 4. Diagrama de blocos do controlador no *LabVIEW*.

- (0) Estrutura *While*, onde o programa dentro dele se repete até que o Botão “stop” seja apertado e o programa encerrado;
- (1) Bloco de função de controle, onde o valor desejado da variável controlada Velocidade do motor em RPM será ajustado. O motor EC 353518 trabalha com velocidades entre 1250 RPM e 3000 RPM (3);
- (2) Estrutura *Case*, caso o valor da variável controlada Velocidade tenha acabado de ser alterado, o *Case* será *True*, converterá o valor em RPM em uma porcentagem (RPM multiplicado por 100 e dividido por 3000, pois o valor máximo (100%) do motor é 3000, o programa foi construído para ser didático, mantendo redundâncias para ser intuitivo, mas o programa poderia ser direto em RPM dividido por 30). Caso a variável controlada Velocidade não tenha sido alterada recentemente o *Case* será *False* e será usado o valor obtido no (6). Em ambos os casos o valor obtido nos *Cases* será escrito no (3) e usado no (6);

- (3) *Daq Assistant* de geração de sinais (*Simulate Signal*, do inglês) para gerar um sinal de Modulação por Largura de Pulso (*PWM*, da sigla em inglês), sendo que *PWM* é uma técnica para obter resultados analógicos por meios digitais. Essa técnica consiste na geração de uma onda quadrada em uma frequência muito alta em que pode ser controlada a porcentagem do tempo em que a onda permanece em nível lógico alto. Esse tempo é chamado de Ciclo de Trabalho (*Duty Cycle*, do inglês) e sua alteração provoca mudança no valor médio da onda, indo desde 0V (0% de *Duty Cycle*) a 5V (100% de *Duty Cycle*). O *Duty Cycle* é a razão do tempo em que o sinal permanece na tensão máxima (5V) sobre o tempo total de oscilação, como está ilustrado na Figura 4.1 e Equação 1 e 2. Na Figura 4.2 é mostrado o *Duty Cycle* de 50 % produzido no *LabVIEW*;

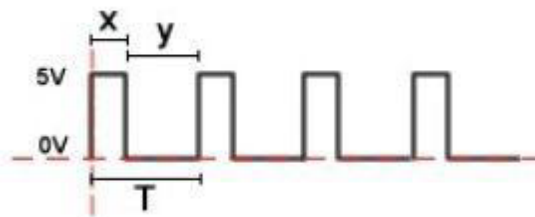


Figura 4.1. *Duty Cycle*, Onde: “x” é o Nível Alto (5V), “y” é o Nível Baixo (0V) e “T” é o Período.

$$\text{Duty Cycle (\%)} = (x/x+y)*100\% = (x/T)*100\% \quad \text{Eq.1}$$

$$\text{Tensão Média} = \text{Tensão máxima} * \text{Duty Cycle (\%)} \quad \text{Eq.2}$$

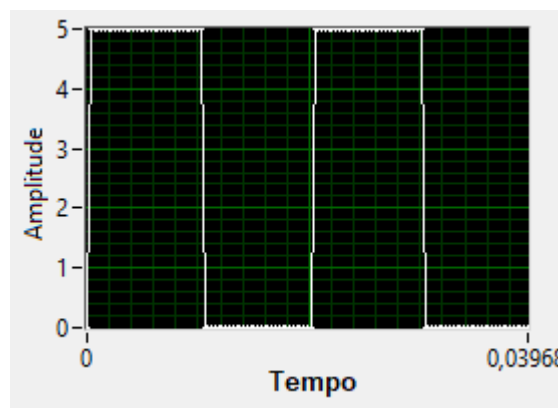


Figura 4.2. *Duty Cycle* de 50% Frequência = 50Hz; Amplitude= 2,5V; Offset = 2,5V; Amostras por segundo = 3125Hz; Números de amostras = 125;

- (4) *Daq Assistant* de Saída Analógica (AO, da sigla em inglês) responsável por escrever o *Duty Cycle* (3) na saída física, sendo essa eletricamente acoplada ao Pino *Enable* do Ponte H L298N, produzindo uma tensão de saída da Ponte H L298N proporcional ao *Duty Cycle* do *PWM* (3);
- (5) *Daq Assistant* de Contador de Entradas do tipo Pulso (*Edge Count*, do inglês) com um *Sample Clock* de 1 segundo, onde os pulsos do KY-040 serão incrementados até completar 1 segundo, depois disso a somatória será multiplicada por 3, obtendo o valor real em RPM da velocidade do motor. O motor EC 353518 varia o RPM conforme a tensão de entrada em uma escala de 1V/125RPM (3). O valor é dividido por 125 para se obter a tensão do RPM e ser usado no (6);
- (6) Bloco de PID, onde ao ser alterado o valor da variável controlada Velocidade (1) convertido em tensão (*Duty Cycle* do *PMW* dividido por 100 e multiplicado por 24, pois a tensão máxima (100%) do motor EC 353518 é 24V, novamente o objetivo é principalmente didático, mas poderia ser simplificado em um bloco de multiplicação por 0,24) é usado como Referência (*Setpoint*, do inglês). O valor de RPM do motor (5) convertido em tensão é utilizado como sinal de entrada realimentado. Após o Valor em tensão ser processado pelo filtro PI o valor é convertido em *Duty Cycle* de *PWM* (Pode ser simplificado em uma multiplicação por 4,16...) e alimenta exclusivamente o *Duty Cycle* do *PMW* (3), o variando afim de induzir a saída em RPM desejada no motor.

PID. O filtro PID (Proporcional, Integral e Derivativo) é uma técnica de controle de processos que une as ações Proporcional, Integral e Derivativa, fazendo assim com que o sinal de erro seja minimizado pela ação proporcional, zerado pela ação integral e obtido com uma velocidade antecipava pela ação derivativa (1).

Conforme Garcia (8), apesar de novas técnicas de controle processos, com utilização de tecnologia digital de ponta, o algoritmo de controle mais utilizado é o PID, apesar deste existir a mais de meio século. Isto se deve a sua fácil implementação tanto analógica quanto digital e à sua versatilidade, proporcionando resultados satisfatórios, mesmo para variações apreciáveis nas características dos processos. Ainda, segundo Garcia (8) o PID apresenta as seguintes vantagens: não requer conhecimento profundo da planta e nem um modelo matemático do processo, embora este seja útil para o projeto do sistema de controle; o PID, é um algoritmo de controle universal baseado na filosofia por realimentação, é versátil e robusto; se as condições de processo mudam a resintonia do controlador usualmente produz controle satisfatório.

Garcia (8) ainda cita as desvantagens do PID: seu desempenho pode não ser satisfatório para processos com a constante de tempo grandes e/ou atrasos longos; se grandes perturbações são frequentes, o processo pode operar continuamente e nunca atingir o estado estacionário desejado. As Figuras 5 e 6 ilustram as equações características PID e a Função de Transferência, respectivamente.

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p T_d \frac{de(t)}{dt}$$

Figura 5. Equações características do PID.

$$\frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i s} + T_d s \right)$$

Figura 6. Função de transferência do PID.

Sintonia do PID. De acordo com Vidal (9), o processo de encontrar os melhores parâmetros de um controlador é chamado de sintonia de malha de controle. Em alguns casos, esses parâmetros podem ser encontrados via tentativa e erro, entretanto cada um dos parâmetros K_p , T_i e T_d pode ter valores que variam desde 0,01 a 100. Aliados aos atrasos inerentes do processo, podem tornar a tarefa demorada e trabalhosa. Em outros casos, é necessário um modelo do sistema razoavelmente preciso para efetuar simulações. Nestes casos, o trabalho e o tempo envolvidos na modelagem e simulação muitas vezes não são justificáveis.

No trabalho utiliza-se o Método da Sensibilidade Limiar de Ziegler-Nichols para ajuste dos parâmetros PID.

1. Neste método faz-se inicialmente as constantes $T_i = \infty$, e $T_d = 0$, trabalhando apenas com a ação de controle proporcional.
2. Experimentalmente vamos aumentar o valor de K_p até que a resposta do sistema apresente uma oscilação sustentada pela primeira vez. A esse valor de K_p estabelece-se a notação de K_{cr} (ganho crítico).
3. O período da senóide encontrada será o nosso P_{cr} (período crítico).
4. Ziegler e Nichols sugeriram escolher valores de K_p , K_i e K_d de acordo com a Tabela1.

Tabela 1. Método da Sensibilidade Limiar de Ziegler-Nichols para ajuste de PID.

Type of Controller	K_p	T_i	T_d
P	$0.5K_{cr}$	∞	0
PI	$0.45K_{cr}$	$\frac{1}{1.2}P_{cr}$	0
PID	$0.6K_{cr}$	$0.5P_{cr}$	$0.125P_{cr}$

Com base nos valores da Tabela 1 os valores PID foram calculados, obtendo $K_p = 0,900$, $T_i = 0.010$ e $T_d = 0$, sendo portando um controlador PI. O *range* de atuação do PI foi ajustado para extremos mínimo e máximo de 0 e 24V, respectivamente. O resultado do controle para o range mínimo e máximo de operação do motor EC 353518, respectivamente 10V (1250 RPM) e 24V (3000 RPM) podem ser observados nas Figura 7, 8 e 9.

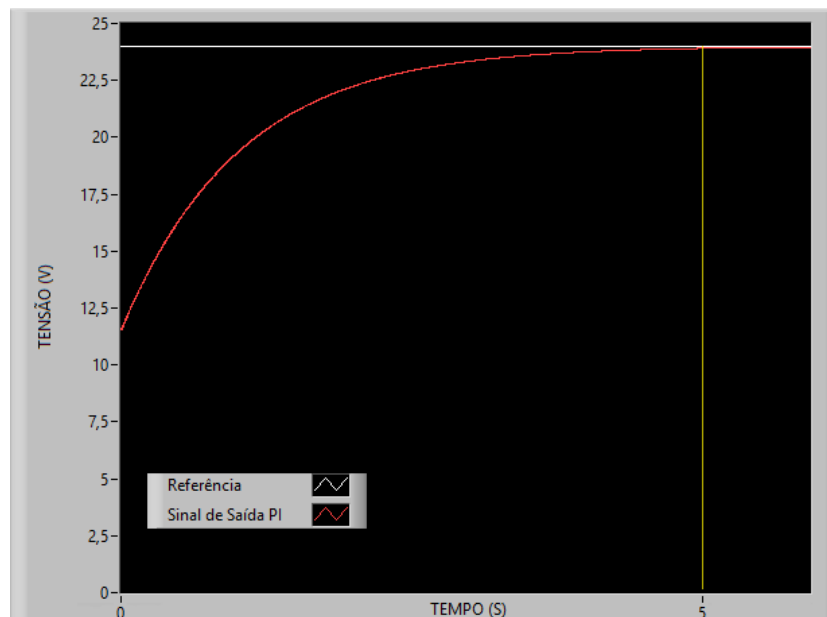


Figura 7. Controle PI com Referência (*Setpoint*) de 24V.

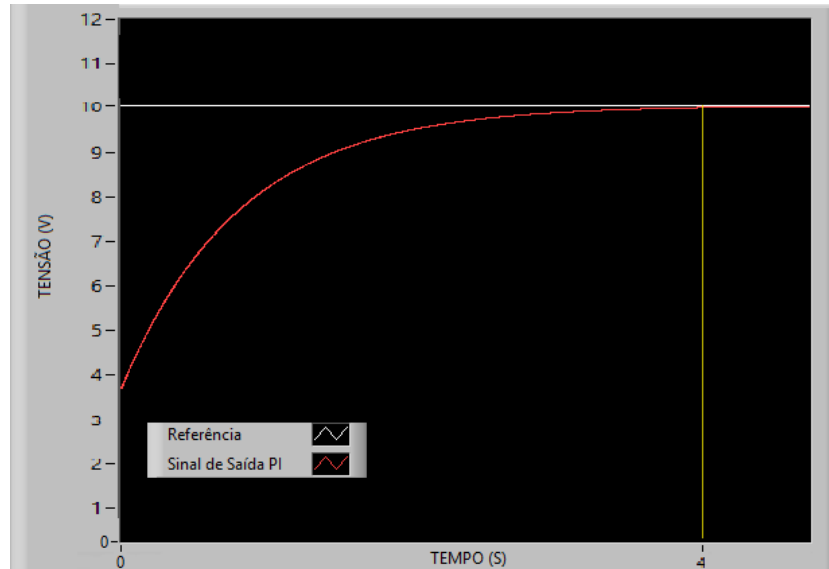


Figura 8. Controle PI para Referência (*Setpoint*) de 10V.

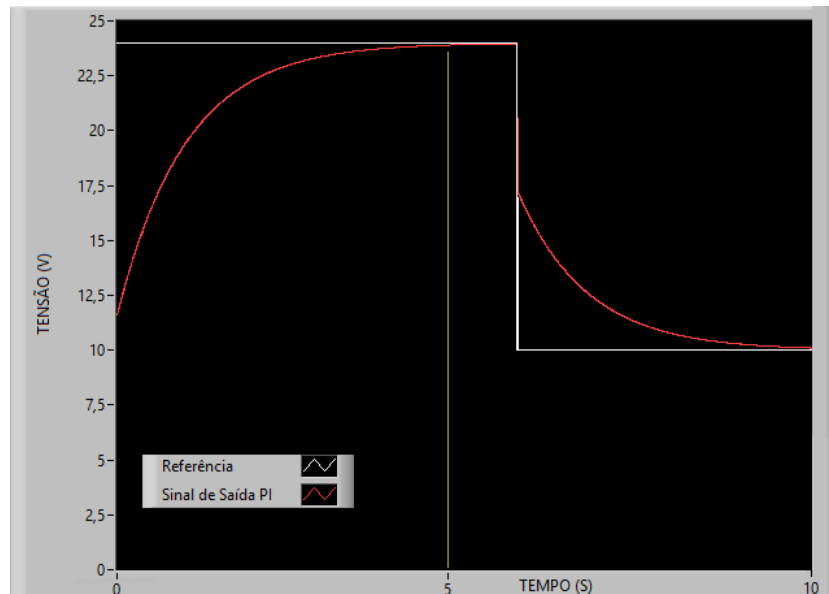


Figura 9. Controle PI variando a Referência (*Setpoint*) de 24V para 10V.

Conclusão. Para testes preliminares, a saída PWM (4) foi acoplada a um osciloscópio onde foi constatado uma precisão na reprodução do *Duty Cycle* do PWM na saída PWM (4). A leitura da Velocidade do motor foi satisfatória e o controle PI da Velocidade apresentou erro em regime estacionário menor que 2%. Os valores de trabalho e amostragem do *Duty Cycle* do PWM (3) foram obtidos empiricamente e futuros trabalhos correlacionarão diferentes sinais PWM e a variação da tensão na Ponte H L298N para motores de diferentes RPM.



Referências

- (1) Dorf RC, Bishop RH. Sistemas de controle modernos. LTC. 2009.
- (2) <http://www.ni.com/pt-br/shop/labview.html> [Internet]. Brasil: LabVIEW National Instruments, Inc.; Acesso em Março de 2018.
- (3) Ensatek “Datasheet EC 45 Flat”, 2015. Maxon. Suíça.
Disponível em: <http://www.ensatek.com.tr/image/urun/ecflat/15EC45FLAT_30_B.pdf>. Acesso em Março de 2018.
- (4) SparkFun “Datasheet L298N”, 2016. STMicroelectronics. Suíça.
Disponível em: < https://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf>. Acesso em Março de 2018.
- (5) Sensor Kit “Datasheet KY-024”, 2016. Keyes. EUA.
Disponível em:
<http://sensorkit.en.joyit.net/index.php?title=KY024_Linear_magnetic_Hall_Sensor>. Acesso em Março de 2018.
- (6) Henry’s Berch “Datasheet KY-040”, 2017. Keyes. EUA.
Disponível em: < <http://eeshop.unl.edu/pdf/KEYES%20Rotary%20encoder%20module%20KY-040.pdf>>. Acesso em Março de 2018.
- (7) Costa, C. “Introdução ao LabVIEW”, 2013. National Instruments. EUA.
Disponível em: <[http://professorcesarcosta.com.br/upload/imagens_upload/introducao_ao_labview_em_6_horas%20\(2\).pdf](http://professorcesarcosta.com.br/upload/imagens_upload/introducao_ao_labview_em_6_horas%20(2).pdf)>. Acesso em Março de 2018.
- (8) Garcia C. Controle de processos industriais, Apostila do curso de pós graduação em Instrumentação, Automação e Controle, IMT – Instituto Mauá de Tecnologia, Escola de Engenharia Mauá, 2003.
- (9) VIDAL LC. Implementação e Comparação de Técnicas de Controle em Plantas Simuladas - Universidade de Taubaté – SP, 2012.



Abstract. This work presents the development of a DC motor speed controller (RPM) using Control techniques and LabVIEW® functions (National Instruments, USA). A closed-loop system enables an efficient response of the controller to disturbances of the external signals, the controller monitors and executes real-time reaction actions to the physical data previously converted into digital or analog electrical pulses by means of transducers, keeping stabilized controlled variable independent of the external environment in the specified conditions, keeping homogeneity inherent in industrial applications, as well as guarding the system from possible technical problems. The controller was developed by means of graphic language in LabVIEW, the actuator is an H-bridge module L298N (STMicroelectronics, Switzerland) that varies the output voltage proportionally to the percentage of Duty Cycle PWM of Enable's Input, have been coupled electrically the L298N to an EC motor 353518 (Maxon, Switzerland) which varies the speed (RPM) linearly to the voltage supplied by he and the transducer is a KY-040 rotary encoder sensor (Keyes, USA), which converts the motor rotor rotation into digital electrical pulses for the count RPM. The motor velocity reading was satisfactory and the PI speed control showed a steady state error of less than 2%.

Key-words. Controller, DC Motor, Controller PI, LabVIEW.