



O USO DE UM SISTEMA REMOTO DE COMANDO E MONITORAMENTO DE MOTOR CC ACOPLADO A UM TACO-GERADOR NO ENSINO DE SISTEMAS DE CONTROLE

Janito dos Santos Ramos – janitodossantos23@gmail.com

Centro Federal de Educação Tecnológica de Minas Gerais – Campus III, Faculdade de Engenharia de Controle e Automação
R. José Péres, 558 - Centro
36.700-000 – Leopoldina – Minas Gerais

Paulo Ricardo Santos – pauloricardo.sts87@gmail.com

José Geraldo Ribeiro Junior – jgrjunior@leopoldina.cefetmg.br

Resumo: *As disciplinas de laboratório representam um meio importante para a formação do aluno, permitindo que o mesmo desenvolva habilidades práticas necessárias ao mercado de trabalho, além de facilitar a visualização de conceitos mais complexos e incentivar na busca de novos conhecimentos durante o curso. Este trabalho consiste no desenvolvimento de um sistema de controle e monitoramento remoto de variável de processo do tipo velocidade, aplicado a um motor CC, com objetivo de ser utilizado em disciplinas laboratoriais. O sistema proposto integra um software, confeccionado em linguagem de programação Java para gerenciar e supervisionar o processo, possibilitando realizar o controle automático da velocidade de um motor CC, remotamente, utilizando arquitetura cliente-servidor e protocolos da pilha TCP/IP (Transmission Control Protocol/Internet Protocol). Os resultados são apresentados com intuito de demonstrar o funcionamento do sistema construído e a sua aplicabilidade.*

Palavras-chave: *Controle, Monitoramento, Comunicação, TCP, UDP.*

1. INTRODUÇÃO

A educação em âmbito geral passa por períodos de reformulação, onde o re-pensar da forma de ensino-aprendizagem ganha cada vez mais destaque (FROYD et al., 2012). É preciso usar a criatividade para, além de manter o interesse do aluno, romper barreiras como a dificuldade de se ter acesso a equipamentos caros, principalmente quando se refere ao curso de Engenharia de Controle e Automação. Uma das possibilidades para minimizar estas barreiras é desenvolver equipamentos de baixo custo, úteis para aplicação ou desenvolvimento de práticas pedagógicas (COELHO et al., 2001).

Organização



UDESC
UNIVERSIDADE
DO ESTADO DE
SANTA CATARINA



UNISOCIESC
Educação e Tecnologia

Promoção



ABENGE
Associação Brasileira de Educação em Engenharia



Neste contexto, o presente trabalho tem como propósito apresentar uma solução didática desenvolvida no Laboratório, por meio do desenvolvimento de um sistema de controle e monitoramento de motor CC (Corrente Contínua), utilizando redes de comunicação TCP/IP, com suporte aos protocolos de transporte TCP (*Transmission Control Protocol*) e UDP (*User Datagram Protocol*).

A utilização do motor CC é motivada pelo fato de que, no âmbito da automação, é muito importante e necessária a utilização de plantas industriais para que os alunos tenham contato com situações próximas ao que enfrentarão na vida profissional. O uso dessas plantas possibilita que o aluno acompanhe de perto o funcionamento de sistemas industriais que, com o desenvolvimento da indústria, tornaram-se cada vez mais complexos.

O controle de velocidade de motores CC é de grande relevância prática tanto no meio acadêmico, auxiliando no processo ensino aprendizagem da teoria de controle clássico e moderno, quanto a nível industrial, sendo indicado em processos que necessitam de velocidades precisas ou com baixo grau de variação (ROSA, 2013).

A aplicação do motor CC tem sido amplamente estudada em trabalhos de pesquisa, visando o desenvolvimento de novas tecnologias e implantação de novas técnicas de controle e monitoramento (PATANÉ, 2008). Os experimentos apresentados neste artigo mostram que é possível utilizar de práticas para fazer o monitoramento remoto, possibilitando a realização de práticas que simulam rotinas industriais.

O restante do artigo está organizado da seguinte forma: na Seção 2 é descrita a proposta e feita a modelagem do sistema e na Seção 3 são mostradas os experimentos e resultados. Finalmente, a Seção 4 são apresentadas as considerações finais do trabalho.

2. DESCRIÇÃO DA PROPOSTA E MODELAGEM DO SISTEMA

Esta seção apresenta detalhes da solução proposta, incluindo detalhes do protótipo físico bem como do sistema de comando e monitoramento.

2.1. Protótipo para controle de velocidade

Foram utilizados dois motores CC, de 3000 rpm, um acoplado ao outro, através de seus eixos. Um funciona como motor e o outro como gerador (taco-gerador) de tensão. O motor desempenha o papel de atuador do sistema enquanto o gerador tem a função de informar para o controlador quanto de tensão o motor está gerando na velocidade que ele se encontra, tornando possível a conversão de tensão para velocidade. Sendo assim, a variável de controle do sistema (*setpoint*) será a velocidade. O sistema físico é mostrado na Figura 1.

Organização



UDESC
UNIVERSIDADE
DO ESTADO DE
SANTA CATARINA



UNISOCIESC
Educação e Tecnologia

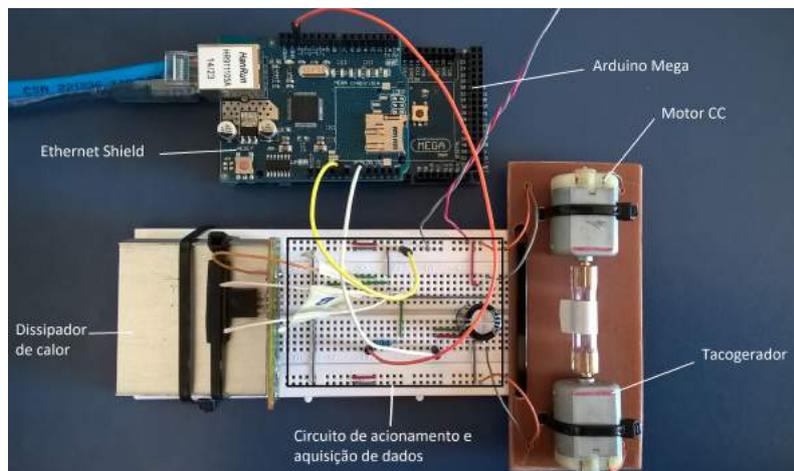
Promoção



ABENGE
Associação Brasileira de Educação em Engenharia



Figura 1 – Sistema físico a ser monitorado.



2.2. Identificação em Malha Aberta do sistema

O motor tem sua velocidade controlada de forma automática pelo sistema. Todo o sistema físico se apresenta como um modelo invariante no tempo, na qual o interesse maior é o comportamento do sistema na saída em relação a entrada. Isso permitiu modelar o sistema físico, aplicando o ensaio de identificação em malha aberta e chegar a função de transferência que rege toda a dinâmica do sistema. A função de transferência é mostrada na Equação 1:

$$G_s = \frac{e^{-\theta s}}{\tau s + 1} = \frac{e^{-0,03s}}{0,2548s + 1} \quad (1)$$

onde θ indica o atraso de transporte (0,03 segundos) e τ é um contante de tempo (0,2548).

Dentro do controlador Arduino Mega existe um algoritmo de controlador PID (Proporcional Integral Derivativo) que possibilita fazer o controle automático da variável do processo. O controlador PID foi projetado utilizando técnicas de controle automático.

2.3. Sistema Remoto de Comando e Monitoramento

O sistema remoto de comando e monitoramento foi desenvolvido utilizando linguagem de programação Java com o ambiente de desenvolvimento integrado (IDE) NET-BEANS (ROCHA et al., 2013). O desenvolvimento do sistema baseou-se na arquitetura cliente-servidor, onde o cliente é o usuário que opera a planta remotamente enquanto o servidor é a parte computacional e física (Figura 1), que possibilita o controle e monitoramento da variável do processo através de envio e recebimento de informações para o controlador Arduino Mega.

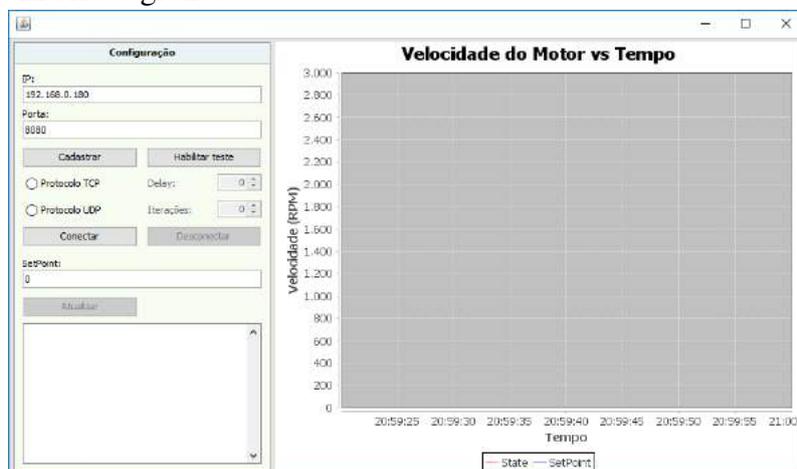
O sistema consiste de uma interface gráfica (Figura 2), para o cliente, com comandos que possibilita realizar controles sobre o motor, do tipo: ligar e desligar a planta, modificar a velocidade do motor e acompanhar em tempo real a resposta do sistema.

Com a finalidade de realizar testes de eficiência na comunicação entre cliente e servidor, foi implementado o suporte a dois protocolos de transporte da arquitetura TCP/IP, o



TCP e o UDP. Com isso é possível, não só trabalhar em práticas relacionadas à disciplina de controle, mas também em testes que envolvam disciplinas como redes de computadores ou sistemas distribuídos.

Figura 2 – Interface gráfica.



Para implementar as rotinas do servidor, desenvolvido para operar dentro do controlador em questão, utilizou-se a linguagem de programação C/C++. A interface do programa do cliente apresenta flexibilidade para escolha do tipo de protocolo a ser utilizado (TCP ou UDP). Portanto, a comunicação entre cliente e servidor utilizará o protocolo escolhido pelo operador do sistema.

O sistema apresenta também uma ferramenta que possibilita realizar testes rápidos de desempenho, de acordo com o protocolo de comunicação escolhido, entre o cliente e o servidor. Com isso é possível fazer comparações, de forma quantitativa, entre os protocolos de comunicação.

2.4. Execução da Ação de Controle no Protótipo

Para que o usuário opere o sistema remoto de comando e monitoramento com sucesso, é preciso que o cliente e servidor estejam conectados e configurados de forma correta. A interface do cliente apresenta dois campos fundamentais, que devem ser preenchidos, são eles: o campo do endereço IP (*Internet Protocol*) e a porta. Este canal de comunicação definido entre cliente e servidor possibilita que os dados sejam transferidos entre um dispositivo e outro, utilizando uma rede de comunicação, seja local, ou uma rede como a Internet.

Com o endereço do servidor definido, deve-se cadastrar, no cliente, o endereço IP e a porta, além de definir o protocolo de transporte utilizado TCP ou UDP. Para isso, a interface apresenta a opção de clicar e depois, basta clicar em CONECTAR para começar a comunicação entre o cliente e o servidor.

Os protocolos TCP e UDP estão na camada de transporte do modelo OSI (*Open System Interconnection*). O TCP é o protocolo responsável por prover confiabilidade na transmissão. O TCP fornece um serviço orientado a conexão, confiável e *full-duplex* para os serviços de aplicação. O UDP é o protocolo não confiável (não provendo controle



de fluxo e erro) e não é orientado a conexão. Entretanto, o UDP é mais simples e possui cabeçalho menor, gerando um sobrecarga menor. O UDP é mais recomendado para aplicações em que o desempenho é mais importante que confiabilidade (KUROSE; ROSS, 2010).

No campo SETPOINT é possível entrar com um valor entre 0 a 3000, correspondendo a velocidade mínima e máxima em RPM, respectivamente, do motor CC. Clicando em ATUALIZAR, o valor será setado dentro do controlador do sistema, realizando então, o controle automático da variável do processo. Caso queira desligar o motor, basta setar o valor 0 (zero) ou simplesmente DESCONECTAR. A reação do processo pode ser acompanhada, em tempo real, pelo gráfico Velocidade do motor por Tempo.

O Sistema apresenta também a ferramenta que permite escolher o número de iterações e atraso (delay). Desta forma é possível fazer testes rápidos, como citado na Seção 2.3. Para utilizá-lo basta clicar em HABILITAR TESTE. Além disso, toda interface e instalação é intuitiva e de fácil entendimento. O tratamento de possíveis erros de comunicação, por exemplo, foram considerados no desenvolvimento do projeto.

3. EXPERIMENTOS E RESULTADOS

Foram realizados diversos testes com o sistema desenvolvido com a finalidade de avaliar, de forma quantitativa e qualitativa, o desempenho dos protocolos de comunicação TCP e UDP, e compará-los. A seguir, serão descritos os testes aplicados e posteriormente apresentados quadros de comparações e conclusões.

Para a realização, da maioria dos testes, foi utilizado um analisador de protocolos de rede, software Wireshark (SANDERS, 2007). Em todos os testes o Wireshark estava configurado para filtrar informações específicas, providas da porta utilizada e do endereço IP do cliente (origem) e endereço IP do servidor (destino), evitando assim a contabilização de possíveis informações externas ao experimento.

3.1. Descrição dos testes empregados em cada protocolo

Teste com apenas uma iteração

O objetivo desde teste é estabelecer a quantidade de pacotes que cada protocolo envia em apenas uma iteração (comunicação entre cliente-servidor) e estratificar essa quantidade de pacotes por tipos. Com este teste, obtém-se respostas óbvias, que caracterizam muito bem a forma de comunicação de ambos os protocolos.

A aplicação consiste em configurar o sistema remoto, através da opção “Habilitar Testes”, na interface do cliente, e estabelecer zero de atraso e um número de iterações igual à 1. Ao passar pela interface de rede, os pacotes são capturados e identificados pelo Wireshark.

Teste com interrupção de conexão forçada do cliente com o servidor

O objetivo é avaliar o comportamento do sistema quando ocorre uma interrupção na conexão entre cliente e servidor, de forma forçada ou quando surgem distúrbios externos. Foi observado se houve perdas de pacotes e, no caso positivo, verificar até quanto o sistema foi prejudicado.



Teste com interrupção de conexão forçada e Reestabelecimento da Conexão

Levando em conta o teste anterior, observa-se neste o comportamento do sistema quando a conexão é reestabelecida. Observou-se a eficiência, por partes dos protocolos, em continuar (ou não) a transmissão e recepção de dados, entre o cliente e o servidor, logo após o religamento.

Teste com rede estressada

Utilizando o Wireshark, verificou-se como o sistema se comporta, em termos de comunicação, quando o usuário realiza várias mudanças sucessivas, em curtos intervalos de tempo, no *setpoint* do processo. Foi verificado se houve perdas de pacotes ou atrasos na transmissão.

Testes em três cenários de redes diferentes

O objetivo é definir quanto tempo leva cada protocolo para trocar informações entre o cliente e o servidor, dentro de um número de 1000 iterações, em cenários diferentes de redes, com aplicação de *delay* de 0, 50 e 100 milissegundos. Para isso é preciso que o usuário habilite os testes na interface do cliente e estabeleça o número de iterações igual a 1000, mudando apenas o valor dos atrasos para cada teste.

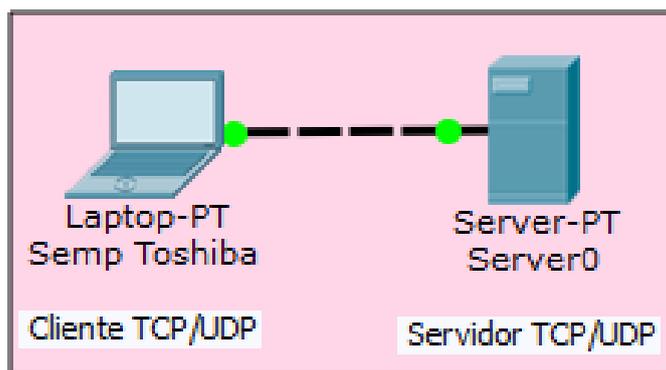
Utilizando o Wireshark, foi possível analisar a quantidade de pacotes enviados e perdidos, em cenários diferentes. Foi possível também calcular a vazão (de acordo com o cenário em questão).

3.2. Descrição dos cenários de rede

Cliente conectado diretamente com o servidor através de um cabo de rede

Neste cenário (Figura 3), o cliente é ligado diretamente ao servidor (módulo Ethernet Shield), sem a utilização de roteadores.

Figura 3 – Cenário 1 - Cliente ligado ao Servidor via cabo.



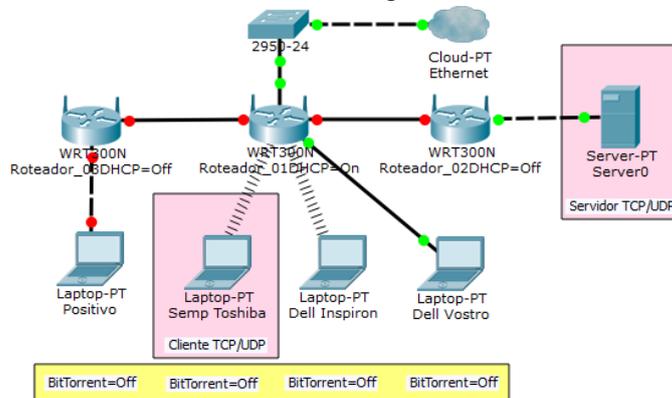
Rede doméstica, com Internet, com pouco tráfego na rede

Este cenário (Figura 4) apresenta quatro computadores, conectados em uma rede privada com três roteadores. Um dos roteadores está ligado no servidor, utilizando uma porta Lan, enquanto o cliente se encontra conectado, via wireless, em outro roteador. Os quatro



computadores encontram-se conectados na Internet gerando pouco tráfego na rede, sem realizações de download via Torrent.

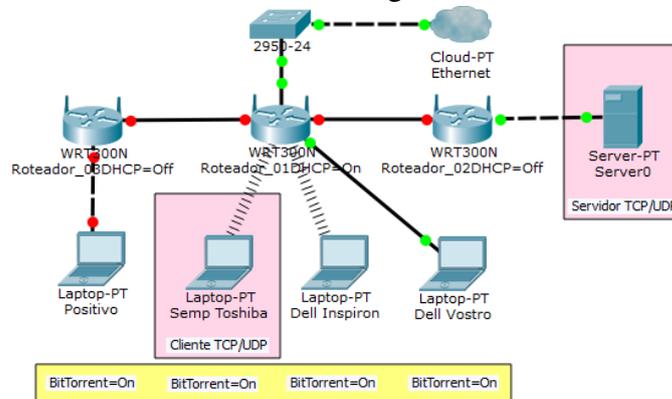
Figura 4 – Cenário 2 - Cliente sem torrent ligado ao servidor via wireless.



Rede doméstica, com Internet, com muito tráfego na rede

Este cenário (Figura 5) é o mesmo descrito anteriormente porém os quatro computadores encontram-se conectados na Internet gerando muito tráfego na rede, com realizações de downloads via Torrent.

Figura 5 – Cenário 3 - Cliente com torrent ligado ao servidor via wireless.



3.3. Resultados obtidos

Os resultados dos testes são apresentados a seguir. Em seguida, são discutidos alguns pontos importantes. Para os testes de 1 a 4, foi utilizado o cenário 2, representado pela Figura 4.

Teste com apenas uma iteração

Neste teste, realizando apenas uma iteração do monitoramento da velocidade do motor CC, observou-se que para o protocolo UDP, em uma iteração, são enviados dois pacotes, referentes ao envio da requisição do cliente para o servidor e a resposta deste para



aquele com os dados a serem mostrados. Já no Protocolo TCP, conforme apresentado na Seção 2.4, uma vez que é preciso estabelecer a conexão e confirmar do recebimento das mensagens, de modo a garantir a confiabilidade, em uma iteração o TCP gera onze pacotes.

Teste com interrupção de conexão forçada do Cliente com o Servidor

Neste teste, para o protocolo TCP, quando feita a interrupção de conexão forçada, ocorreu o envio de vários pacotes do tipo [TCP Retransmission], que são enviados tentando reenviar o último pacote a ser enviado. Como pode ser visto, o TCP insiste em enviar um pacote cujo equipamento de destino não envia uma mensagem de confirmação de recebimento. Entretanto, esta insistência possui um tempo limite, que quando estourado o força a encerrar a conexão. Para o protocolo UDP, observou-se que mesmo com a interrupção forçada do servidor, o cliente continuou enviando os pacotes e estes foram perdidos pois, houve uma interrupção da troca de mensagens entre ambos. A mesma situação aconteceu quando feita a interrupção forçada do cliente.

Teste com interrupção de conexão forçada e Reestabelecimento da Conexão

Neste teste, para o protocolo TCP observou que quando feita a interrupção de conexão forçada, houve envio do pacote do tipo [TCP Retransmission]. Entretanto, quando a conexão foi reestabelecida, antes que o tempo limite de interrupção fosse estourado, observou que houve novamente a comunicação normalmente entre cliente e servidor. Para o protocolo UDP, o sistema apresentou o mesmo comportamento do teste anterior.

Teste com rede estressada

Observou-se neste teste que foram enviados vários pacotes indica que houve uma retransmissão desnecessária. Foram identificados também, o envio de outros tipos de pacote que indicam: falsa confirmação por parte do receptor da mensagem, confirmação de recebimento vazia e retransmissão da confirmação de recebimento decorrente do reenvio de um pacote. É importante salientar que essas mensagens indicam que houve uma série de distúrbios na comunicação entre cliente e servidor e não perdas de pacotes. Para o protocolo UDP a rede manteve-se em funcionamento, sem que houvesse o registro de pacotes de indicação de falha.

Testes em três cenários de redes diferentes

Os resultados destes testes estão apresentados nas Tabela 1, 2 e 3 para os cenários 1, 2 e 3 respectivamente.

Organização



UDESC
UNIVERSIDADE
DO ESTADO DE
SANTA CATARINA



Promoção





Tabela 1 – Resultados do cenário 1.

	Pacotes enviados	pacotes recebidos	tempo de transmissão	Vazão
Teste com delay 0 ms, 1000 iterações e cenário 1				
TCP	6011	5008	19,02s	142 kb/s
UDP	1000	1000	10,57s	36 kb/s
Teste com delay 50 ms, 1000 iterações e cenário 1				
TCP	6151	5012	63,20s	44 kb/s
UDP	1000	1000	58,77s	6394 b/s
Teste com delay 100 ms, 1000 iterações e cenário 1				
TCP	7008	5008	111,95s	28 kb/s
UDP	1000	1000	107,24s	3501 b/s

Tabela 2 – Resultados do cenário 2.

	Pacotes enviados	pacotes recebidos	tempo de transmissão	Vazão
Teste com delay 0 ms, 1000 iterações e cenário 2				
TCP	5959	5010	27,72s	99 kb/s
UDP	1000	1000	12,51s	30 kb/s
Teste com delay 50 ms, 1000 iterações e cenário 2				
TCP	6104	5008	69,45s	40 kb/s
UDP	1000	1000	60,56s	6210 b/s
Teste com delay 100 ms, 1000 iterações e cenário 2				
TCP	6335	5008	119,06s	24 kb/s
UDP	1000	1000	101,11s	3647 b/s

Tabela 3 – Resultados do cenário 3.

	Pacotes enviados	pacotes recebidos	tempo de transmissão	Vazão
Teste com delay 0 ms, 1000 iterações e cenário 3				
TCP	6418	5665	214,67s	13 kb/s
UDP	1000	1000	11,28s	3392 b/s
Teste com delay 50 ms, 1000 iterações e cenário 3				
TCP	6111	5170	220,82s	12 kb/s
UDP	1000	1000	67,52s	5571 b/s
Teste com delay 100 ms, 1000 iterações e cenário 3				
TCP	6369	5074	163,9s	17 kb/s
UDP	1000	1000	108,5s	3465 b/s

Analisando as tabelas, observa-se que para o protocolo TCP houve perdas de pacotes significativas nos três cenários. Já para o protocolo UDP, verifica-se que não houve perdas de pacotes, em nenhum dos cenários. Em relação ao tempo do envio e recebimento de pacotes, observa-se que TCP demora mais tempo para enviar os pacotes. Verifica-se também que quanto menor for o *delay*, maior é o tempo necessário para a transmissões de pacotes e maior é a vazão. Finalmente, o sistema se mostrou eficiente para o suporte no processo de ensino-aprendizagem de Controle, Redes de Computadores e Sistemas



Distribuídos.

4. CONSIDERAÇÕES FINAIS

Este artigo propõe o uso de um sistema remoto de comando e monitoramento de motor C, acoplado a um taco-gerador, no ensino de sistemas de controle. O objetivo é apresentar uma alternativa à dificuldade de acessar laboratórios específicos para práticas envolvidas à automação. O artigo apresenta ainda vários testes de desempenho que, além de mostrarem ser possível fazer as práticas à distância, é possível utilizar o sistema proposto para o estudo em disciplinas como redes de computadores e sistemas distribuídos.

Os testes de comunicação utilizando os protocolos UDP e TCP buscaram compreender as diferenças de desempenho entre os dois protocolos em três configurações e tempos de atraso distintos entre envio e recebimento de pacotes por parte do cliente.

Enfim, para o controle de velocidade de Motor CC, o protocolo UDP tem melhor aplicação por ser mais rápido. Apesar de não haver confirmação da recepção dos pacotes, o UDP tem alta velocidade e, por isso, tem maior aplicabilidade nesse sistema, pois qualquer dado retransmitido já estará obsoleto.

Em relação ao aprendizado, este trabalho agregou grande gama de conhecimento por estar em condições parecidas ao ambiente industrial, além de possibilitar a interdisciplinaridade, incluindo vários campos de conhecimento em engenharia.

4.1. Agradecimento

Os autores agradecem ao MEC/SEsu, FNDE, CAPES, CNPq, FAPEMIG, Fundação CEFETMINAS e CEFET-MG pelo apoio ao desenvolvimento desse trabalho.

REFERÊNCIAS BIBLIOGRÁFICAS

COELHO et al. Projeto motor taco-gerador. *Anais: XXIX – Congresso Brasileiro de Ensino de Engenharia*, Porto Alegre: PUC-RS, 2001.

FROYD et al. Five major shifts in 100 years of engineering education. *Proceedings of IEEE*, p. 1344–1360, 2012.

KUROSE, J.; ROSS, K. *Rede de computadores e a internet - Uma abordagem top-down*. 5 edição. ed. São Paulo: [s.n.], 2010.

PATANÉ, E. *Implementação de controle de velocidade em malha fechada para motores de corrente contínua utilizando sistema de aquisição de dados*. Dissertação (Mestrado) — Escola de Engenharia Mauá do Centro Universitário do Instituto Mauá de tecnologia, São Caetano do Sul, 2008.

ROCHA, F. et al. Sistema automatizado utilizando tecnologias de baixo custo para controle e monitoramento de irrigações no vale do açu. *IX – Congresso de Iniciação Científica do IFRN*, Rio Grande do Norte, p. 495–502, 2013.

ROSA, E. M. e. R. B. A. Controle de velocidade de motor cc e taco-gerador. *Revista da Vinci*, Chapecó – Santa Catarina – Brasil, 2013.

Organização



UDESC
UNIVERSIDADE
DO ESTADO DE
SANTA CATARINA



UNISOCIESC
Educação e Tecnologia

Promoção



ABENGE
Associação Brasileira de Educação em Engenharia

Joinville/SC – 26 a 29 de Setembro de 2017
UDESC/UNISOCIESC
“Inovação no Ensino/Aprendizagem em Engenharia”



COBENGE 2017
XLV CONGRESSO BRASILEIRO DE EDUCAÇÃO EM ENGENHARIA

SANDERS, C. Practical packet analysis: using wireshark to solve real-world network problem. San Francisco: No Starch Press, 2007.

Abstract: *The laboratory disciplines represent an important means for the formation of the student, allowing him to develop practical skills necessary for the labor market, besides facilitating the visualization of more complex concepts and encouraging in the search of new knowledge during the course. This work consists in the development of a system of control and remote monitoring of variable of process type of speed, applied to a motor DC, aiming to be used in laboratory disciplines. The proposed system integrates a software, made in Java programming language, to manage and supervise the process, allowing automatic control of the speed of a DC motor, remotely, using client-server architecture and TCP/IP ((Transmission Control Protocol/Internet Protocol)) protocol stack. The results are presented in order to demonstrate the operation of the system and its applicability.*

Palavras-chave: *Control, Monitoring, Communication, TCP, UDP.*

Organização



UDESC
UNIVERSIDADE
DO ESTADO DE
SANTA CATARINA



UNISOCIESC
Educação e Tecnologia

Promoção



ABENGE
Associação Brasileira de Educação em Engenharia