

Desenvolvimento de Simulador Industrial para Processamento de Gás Natural

Leandro Siqueira Glória
leandrosgloria@hotmail.com
IFF

Adelson Siqueira Carvalho
adelsonsc@gmail.com
IFF

Farlen Souza Grillo
farlensouzagrillo@gmail.com
IFF

Dennis Reis da Silva
dennisreis27@gmail.com
IFF

Resumo: O presente trabalho trata do desenvolvimento de simuladores industriais, especificamente para a indústria de processos contínuos, com foco nos ramos do petróleo e gás. Os simuladores são desenvolvidos a partir da integração entre softwares de supervisão de processos (InTouch®) e de simulação matemática dinâmica (MatLab®/Simulink®). As telas de supervisão são desenvolvidas para facilitar a interface entre o operador do simulador e o modelo matemático do sistema simulado, proporcionando uma interface mais próxima das encontradas na prática profissional, sem perda da fidelidade no comportamento do sistema real. Aspectos relativos à utilização destes simuladores em sala de aula são suscitados bem como as vantagens em se utilizar esta ferramenta na melhoria do processo de ensino-aprendizagem em cursos tecnológicos.

Palavras Chave: Simuladores - Integração - ensino-aprendizagem - Processos industriais -

1 Introdução

No mundo contemporâneo, o uso de simuladores computacionais vem se tornando cada vez mais comum, por meio dele tem-se a oportunidade de simular algo que realmente acontece sem ter o risco de errar como em um sistema real. Eles têm por objetivo a aprendizagem, seja de alunos ou operadores, de forma eficiente e dinâmica onde o usuário poderá ver o que acontece ao se modificar o valor de alguma variável do processo. Como descrito por Pidd (2004), a simulação computacional é o processo de projetar um modelo computacional de um sistema real e conduzir experimentos no computador com este modelo com o propósito de entender seu comportamento e/ou avaliar estratégias para sua operação.

O presente trabalho trata do desenvolvimento de simuladores industriais, especificamente para a indústria de processos contínuos, com foco nos ramos do petróleo e gás natural. Em comum todos estes processos industriais possuem como base para a transformação de matéria-prima em produto, processos fluídico-térmicos, conseqüentemente dinâmicos. Estes simuladores são criados a partir da integração entre softwares de supervisão de processos (InTouch®) e de simulação matemática dinâmica (MatLab®/Simulink®). O MatLab® é um software utilizado para projeto e simulação de sistemas de controle, nele podem ser modelados os sistemas dinâmicos inerentes à aplicação do simulador em desenvolvimento. O InTouch® é um software nativo da área de automação industrial denominado supervisor, nele são montadas as telas de supervisão que normalmente os operadores das plantas industriais operam e estão em constante contato.

As telas de supervisão são o visual de todo o processo, com o intuito de facilitar as ações do operador do simulador. O modelo matemático é o comportamento virtual do sistema físico a ser simulado, assim proporcionando uma interface mais próxima das encontradas na prática profissional, sem perda da fidelidade no comportamento do sistema real.

2 Softwares

O simulador foi desenvolvido com tecnologias conhecidas dos alunos do curso de Engenharia de Controle e Automação.

Com a grande necessidade de se criar simuladores de alto desempenho, ou seja, o que simule o mais próximo do real tem-se a necessidade de softwares com recursos poderosos e com tecnologias conhecidas dos alunos do curso de Engenharia de Controle e Automação. Então foram escolhidos dois softwares que nos possibilitam fazer tais simuladores, o MatLab® e o SCADA InTouch®.

2.1. Software de Simulação Matemática

O MatLab® (MAtrix LABoratory) é um software que tem por finalidade fazer cálculos de matrizes, este software possui um ótimo desempenho voltado para o cálculo numérico. Além disso, também integra análise numérica, cálculo com matrizes, processamento de sinais e construção de gráficos de visualização fácil em que as soluções e os problemas são escritos somente matematicamente, facilitando assim sua utilização, ao contrário da programação tradicional.

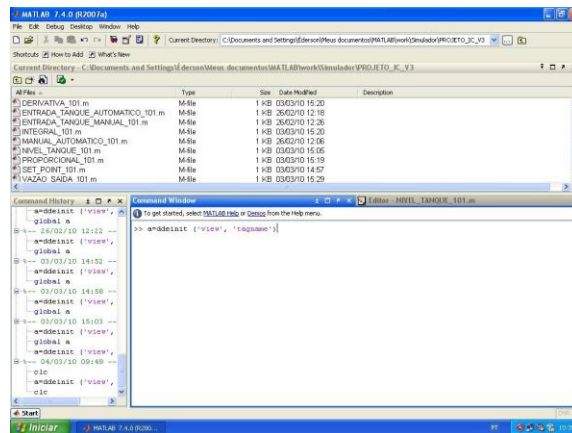


Figura 1: Tela Inicial do MatLab®

Em sua tela principal, temos a janela de comandos, onde será feita a toda configuração de comunicação entre o MatLab® e o SCADA InTouch®. Nesta janela também temos como acessar todos os arquivos do programa, visualizar o histórico de comando e também o editor de comandos.

Este software (MatLab®) possui algumas extensões (Chamadas de toolboxes ou blocksets), sendo o mais conhecido o Simulink, que é um toolbox com uma interface de diagrama de blocos. O Simulink é uma ferramenta para modelagem, utilizada para a simulação e análise de sistemas cujo comportamento é dinâmico. O Simulink é bastante usado em processamento de sinais e teoria de controle para projeto e simulação multi-domínios.

No Simulink será feita toda construção do comportamento de nosso processo, nele serão inserido controladores, funções do sistema, ganhos, etc. buscando a máxima fidelidade ao sistema real.

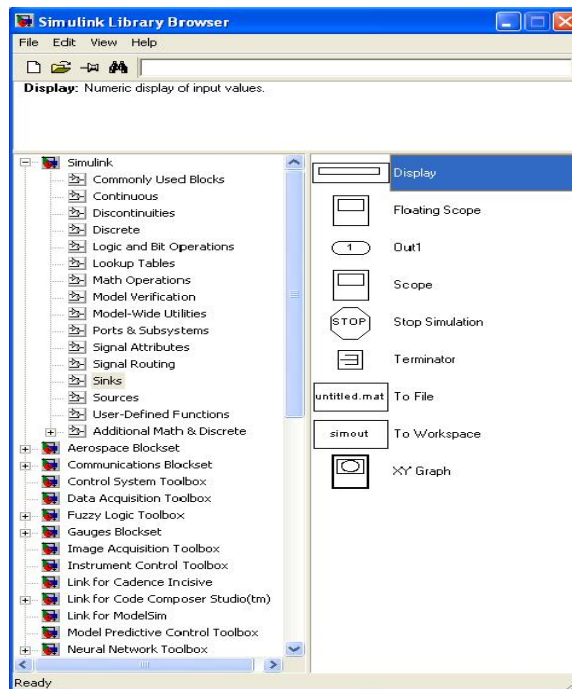


Figura 2: Tela Inicial do Simulink

2.2. SCADA INTOUCH®

O SCADA InTouch® é um software largamente utilizado nas indústrias automatizadas, este software refere-se a controle e aquisição de dados, onde existe um ou mais computadores monitorando e controlando algum processo. O SCADA InTouch® é um software que tem por finalidade mostrar em “tempo real” tudo de importante que acontece em uma planta através de interfaces de alto nível para o operador.

As interfaces devem respeitar conceitos de ergonomia e também um padrão de telas, propiciando ao operador um bom ambiente de trabalho. Este software surgiu para acabar com os painéis de controle, através dele é possível ligar e desligar equipamentos, visualizar a ocorrência alarmes entre outras coisas sem a necessidade de ter um operador no campo.

Este software é formado basicamente por duas janelas: a de criação (WindowMaker) e a de execução (WindowView). O WindowMaker é onde podemos inserir ou criar objetos, configurando o que será feito, como podemos ver abaixo.

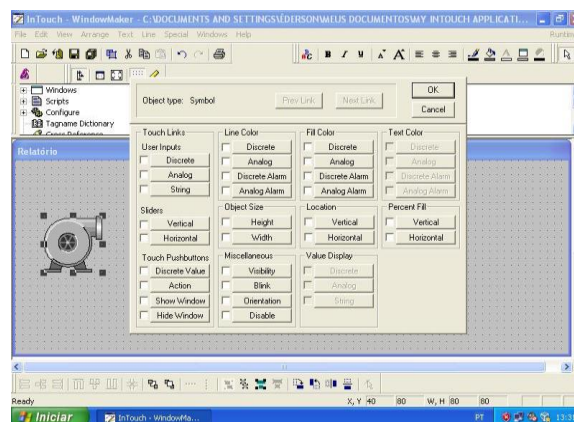


Figura 3: WindowMaker

Depois de construir o objeto desejado, então deve-se executar o *runtime* que está localizado no canto superior direito do WindowMaker. Ao se fazer isso, abrirá o WindowView, nesta janela é que se pode visualizar o comportamento de seus objetos.



Figura 4: Bomba Desligada

Ao abrir a tela, a bomba se encontra em estado normal, ou seja, desligada. Após acionar o botão de liga, o estado lógico é modificado e então a bomba é acionada como pode ser visualizado na figura 5.



Figura 5: Bomba Ligada Após o Acionamento

3 Comunicando os Softwares

Utilizando três comandos básicos (comunicação, leitura e escrita), integram-se os softwares. Para isto foi utilizado o protocolo DDE (Dynamic Date Exchange), presente no Sistema Operacional Windows, que é um protocolo composto por um conjunto de mensagens e conceitos e que permite a troca de dados entre softwares que aceitam o mesmo protocolo. O DDE envia mensagens entre aplicações que partilham dados e usa memória partilhada para efetuar a troca de dados entre aplicações.

O primeiro passo será executar no MatLab® o comando para estabelecer a comunicação, esse comando será:

$a = ddeinit$ ('nome da aplicação', 'nome do tópico')

que será configurado da seguinte forma:

$a = ddeinit$ ('view', 'tagname')

onde a é a variável que armazena a requisição por conexão (afirmativa ou negativa), $view$ é o nome da aplicação servidora e $tagname$ se refere à sessão da aplicação servidora que será acessada.

Ao se efetuar esse comando, devemos ver qual a resposta do MatLab® a respeito da conexão, se a resposta for diferente de "0", significa que a conexão virtual foi estabelecida com sucesso.

Até agora, somente a parte de conexão virtual entre os softwares foram efetuadas, sendo necessário agora à configuração para a troca de dados, onde temos a configuração destinada a apenas leitura e outra destinada à escrita dos dados na aplicação servidora.

3.1. OPERAÇÕES DE LEITURA E ESCRITA

Para ler uma variável na aplicação servidora, é necessário que a configuração seja feita da seguinte forma:

$b = ddereq$ ('canal de conexão', 'nome da variável na aplicação servidora')

Onde b é a variável que armazena a resposta à requisição por valor da variável controlada, $canal$ é o nome da variável que recebeu o retorno do pedido de conexão e $nome da variável$ se refere a variável controlada que está sendo medida e disponibilizada no processo.

Abaixo temos um exemplo de leitura de variável utilizada em nosso simulador, onde a variável da aplicação servidora é a vazão de entrada de um tanque.

$b = ddereq$ ('a', 'VAZAO_ENTRADA')

Feito isso, o MatLab® será capaz de ler as variáveis na aplicação servidora podendo agora este valor interagir no modelo matemático.

Para escrever em uma variável na aplicação servidora, é necessário que a configuração seja feita da seguinte forma:

$c = ddpoke$ ('canal de conexão', 'nome da variável na aplicação servidora', 'valor')

Assim como na leitura, o canal e nome da variável terão a mesma função e valor é o valor que será escrito na variável manipulado na sessão da aplicação servidora.

Abaixo temos um exemplo de escrita de variável utilizada em nosso simulador, onde a variável da aplicação servidora é a vazão de saída de um tanque.

$c = ddepoke('a', 'SAIDA_TANQUE_101', 'd')$

Ao fazer a configuração, o MatLab® será capaz de escrever valores na aplicação servidora, porém temos que tomar o cuidado para não usar variáveis iguais, pois isso irá gerar um conflito em nosso sistema.

3.2. SIMULINK® PARA COMPOSIÇÃO DO MODELO

Com os três comandos básicos (comunicação entre softwares, leitura e escrita) é possível realizar o ciclo básico de um sistema de controle no qual a função dos sensores é realizada pelo comando `ddereq` e a função dos atuadores realizada pelo comando `ddepoke`.

Os comandos `ddereq` e `ddepoke` devem ser utilizados dentro de blocos MatLab® Function, exibido na figura a seguir, para que possam ser chamados a cada repetição no Simulink®.

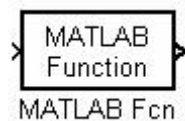


Figura 6: Bloco MatLab® Function

As figuras abaixo a seguir mostram a configuração de leitura em uma MatLab® Function, onde o primeiro passo é realizar a configuração que se deseja e antes dela por a palavra “function”, ao salvar o arquivo, para padronizar, utilizamos o nome dado a function.

Figura 7: Exemplo de configuração para leitura da variável do InTouch®

Na parte inferior, vemos que o nosso arquivo foi salvo como “ENTRADATANQUE101”, esse é o nome de arquivo que deverá ser configurado nos parâmetros da MatLab® function no Simulink®.

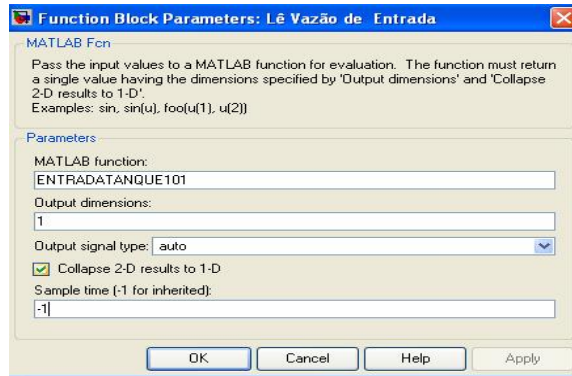


Figura 8: Parâmetro de configuração do bloco no Simulink

Ao colocar o nome da MatLab® Function, esta já estará fazendo o que foi programado nela, onde no caso da figura acima está lendo a vazão de entrada do sistema.

Para fazer à escrita, devemos seguir os mesmos passos utilizados para leitura, porém com algumas mudanças na estrutura no Simulink como pode ser visto nas figuras abaixo.



Figura 9: Exemplo de configuração para escrita da variável do InTouch®

Para a configuração de escrita, além de inserir o nome da MatLab® Function, devemos acrescentar as entradas vindas do mux como podemos ver a seguir, para um caso de duas entradas.

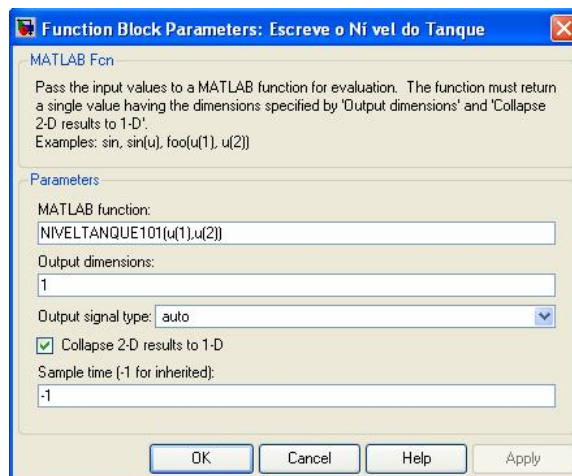


Figura 10: Parâmetro de configuração do bloco no Simulink

Obs.: Para o funcionamento, deve-se atentar ao fato da necessidade da variável de conexão (nesse caso 'a') estar sempre na entrada do sistema.

4 PROTÓTIPO

A partir do conhecimento das tecnologias inicia-se a fase de construção do simulador. Este tem o propósito de virtualizar parte de uma unidade de processamento de gás natural. Este possui subsistemas, são equipamentos como separador de trifásico e torre desbutanizadora. Cada subsistema se compõe de diferentes quantidades e tipos de malhas de controle.

Sendo Assim a criação do simulador será dividida em partes gráficas (telas de supervisão) e partes dos modelos matemáticos de comportamento dos sistemas (MATLAB®/ Simulink).

Nas telas foram inseridos os equipamentos e suas respectivas malhas de controle. Para controlar os subsistemas, são utilizadas válvulas de controle, onde estas possuem janelas de ajustes de proporcional, integral e derivativo (PID), para visualizar o comportamento das variáveis no sistema é posto também telas de gráfico de tendência. Para uma melhor visualização dividiu-se em partes a tela com os equipamentos.

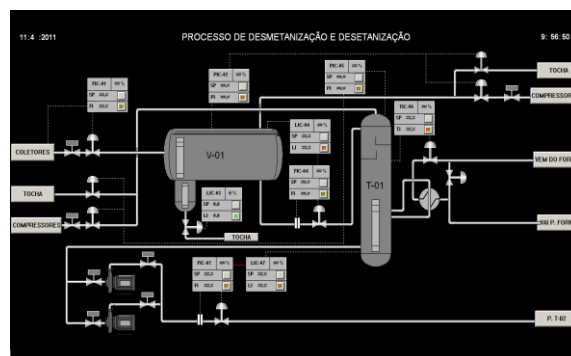


Figura 11: Tela completa do supervisório

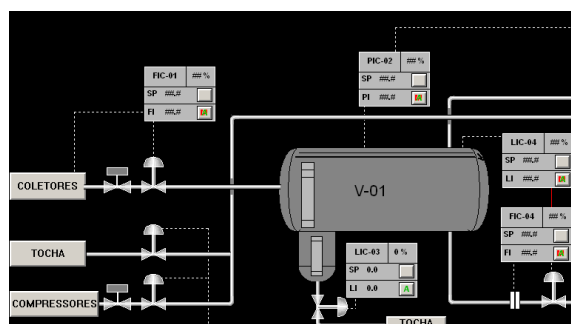


Figura 12: Somente o Separador trifásico

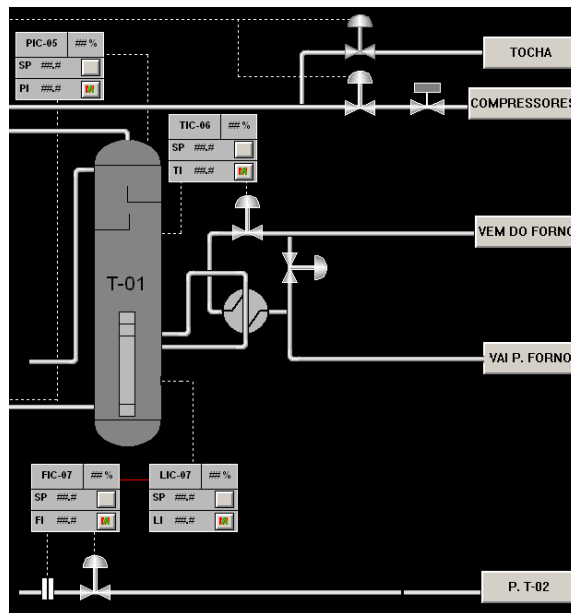


Figura 13: Somente o Desbutanizador

Finalizada a parte gráfica do simulador foi iniciada a pesquisa sobre o comportamento do sistema para o desenvolvimento do modelo matemático pelo qual os processos responderam, para isto é utilizado o MATLAB®/Simulink.

O processo de criação do modelo consiste de inserir em conjunto dos blocos de leitura e escrita o bloco PID e um bloco onde possui a função de transferência da malha de controle. Além de inserir um bloco para comando de manual/automático (M/A) ao sistema.

Cada malha de controle possui uma função de transferência, PID e comando de M/A, com isto, esta segunda parte do projeto ainda encontra-se em fase de desenvolvimento pelo qual já foi feita os modelos matemáticos dos sistemas, mas não as adequações de funcionamento para um sistema próximo ao real.

Ao se integrar os sistemas deve-se atentar em construir uma tabela com todos tagnames utilizados assim como seus respectivos nomes de variáveis cadastradas no MatLab®, pois se por algum motivo ser inserido dois nomes de variáveis iguais, ocorrerá conflito de informações.

5 Aplicação do Protótipo

Com o protótipo construído, chega-se a fase de utilização, sendo um deles a apresentação em uma aula do curso de Engenharia de Controle e Automação no Instituto Federal Fluminense, o IFF.

Durante a aula pode-se perceber o grande interesse dos alunos da Engenharia. Eles questionaram as variáveis controladas, quais serem manipuladas, de que forma serão controladas, que instrumentos podem ser utilizados para medição e atuação do processo.

Que se pode observar a aplicação dos conhecimentos adquiridos no curso de Engenharia de Controle e Automação do IFF.

Em seguida a apresentação, um formulário com afirmativas seguido de valores correspondentes a níveis de concordância e discordância relacionando ao simulador. Abaixo são apresentados gráficos que representam cada afirmativa com os níveis e a quantidade de pessoas que marcaram cada nível.

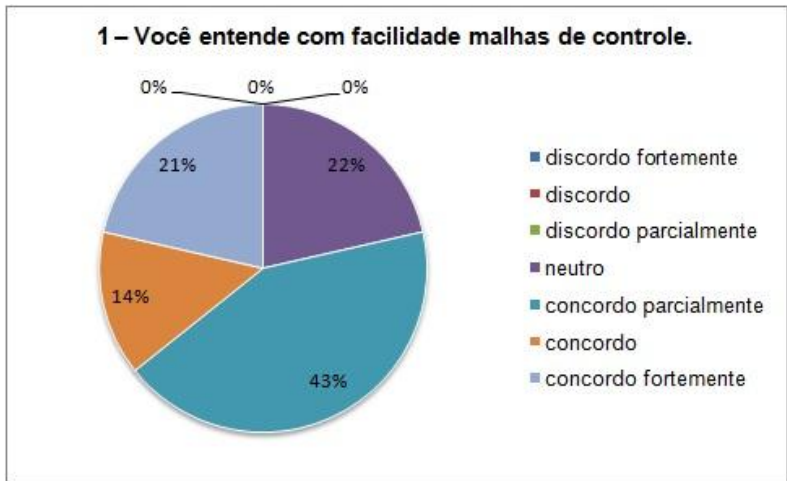


Gráfico 1: Gráfico referente a afirmação um

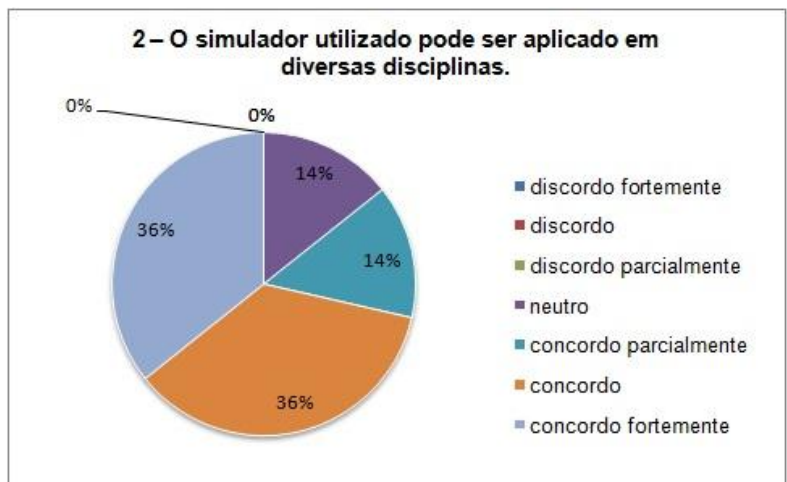


Gráfico 2: Gráfico referente a afirmação dois

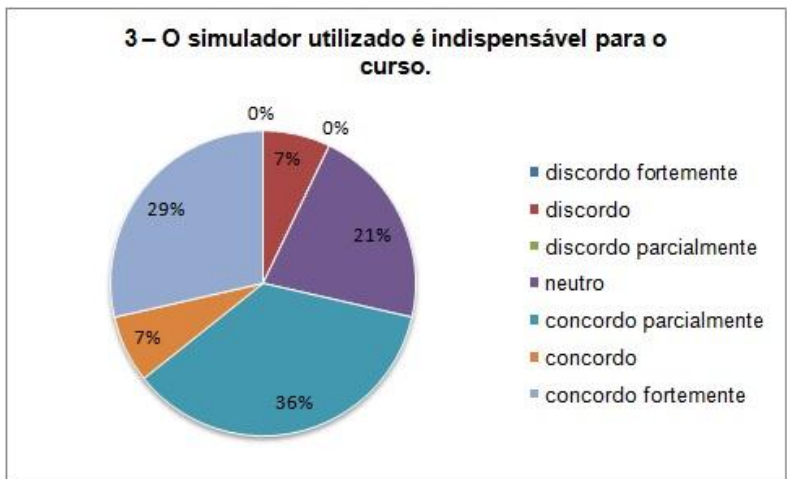


Gráfico 3: Gráfico referente a afirmação três

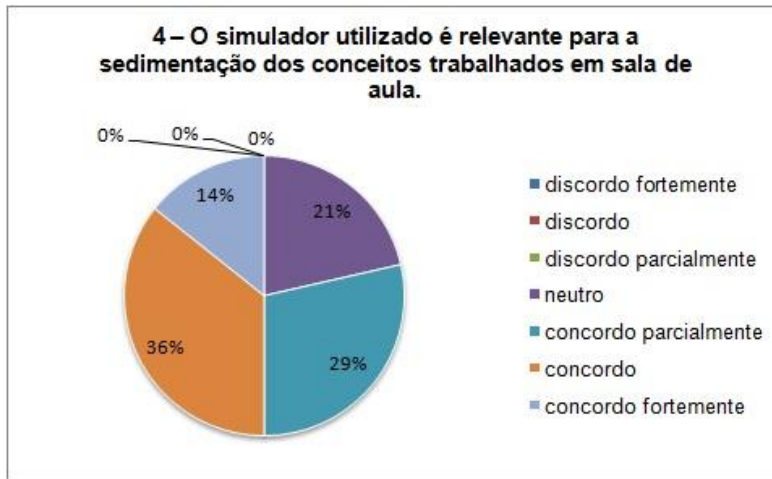


Gráfico 4: Gráfico referente a afirmação quatro

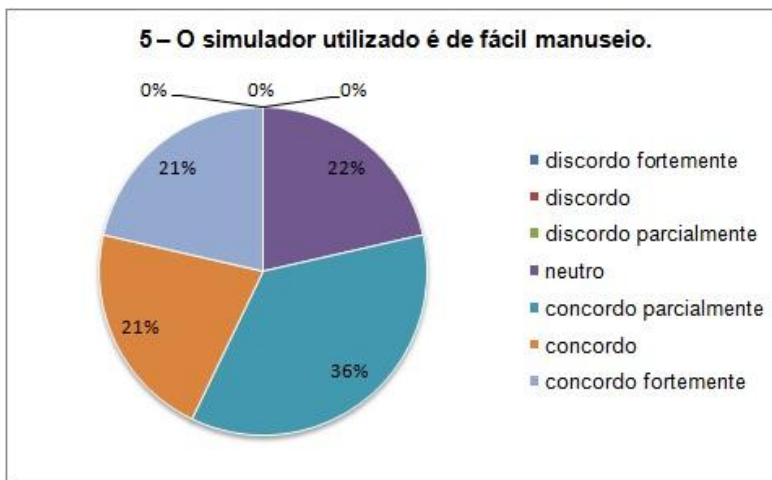


Gráfico 5: Gráfico referente a afirmação cinco

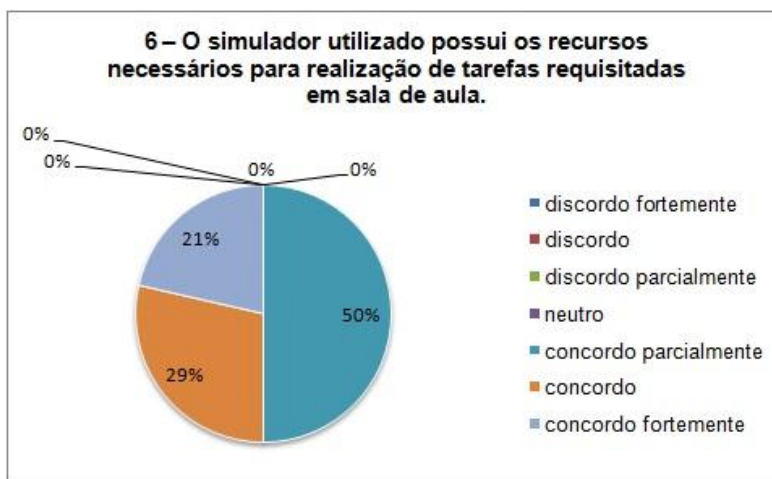


Gráfico 6: Gráfico referente a afirmação seis

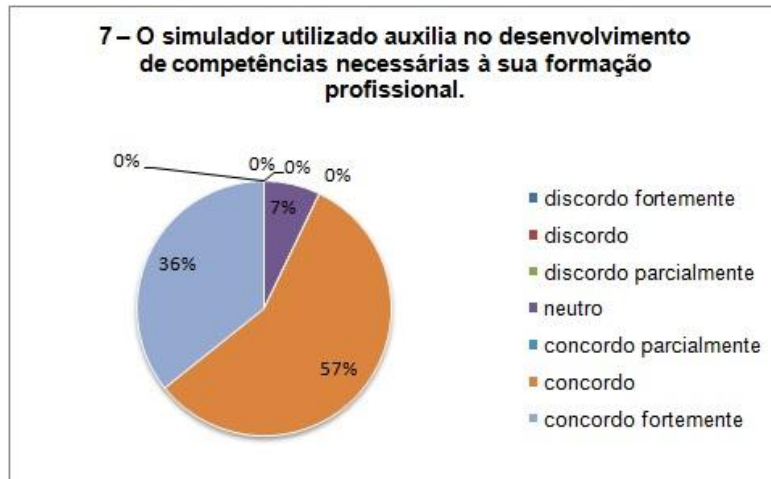


Gráfico 7: Gráfico referente a afirmação sete

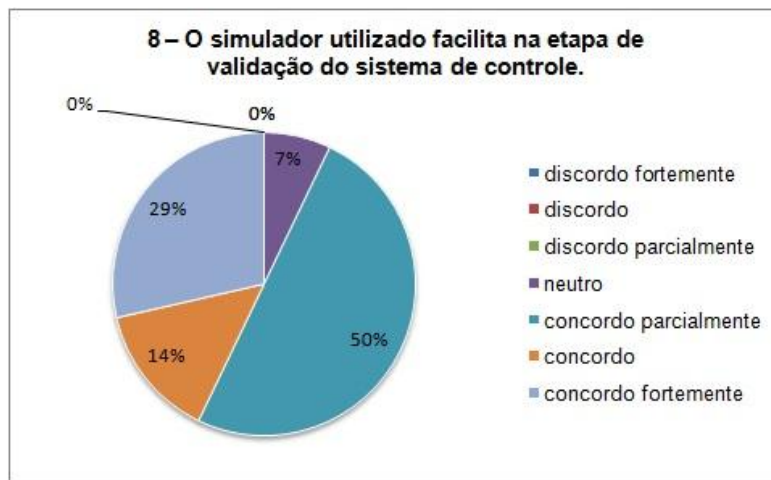


Gráfico 8: Gráfico referente a afirmação oito

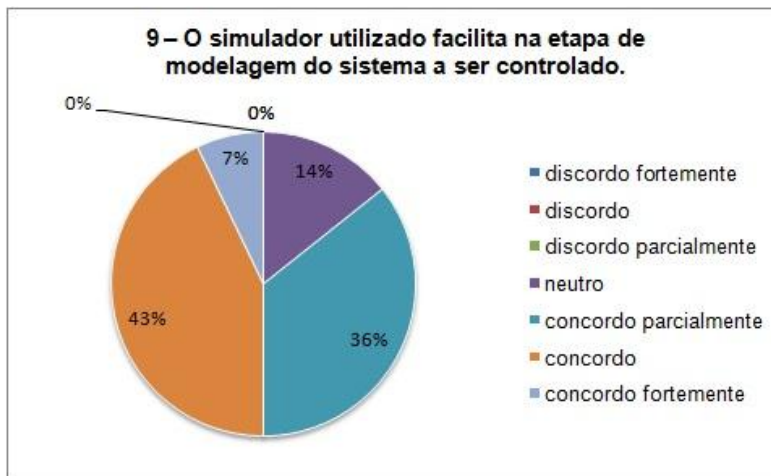


Gráfico 9: Gráfico referente a afirmação nove

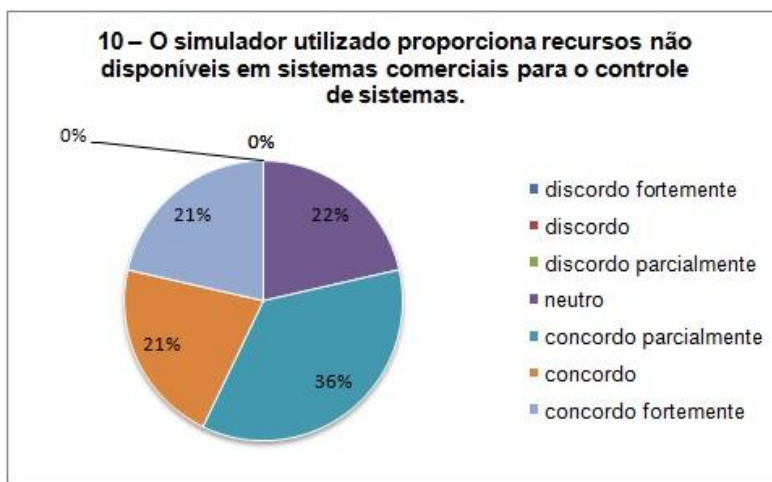


Gráfico 8: Gráfico referente a afirmação oito

Os gráficos indicam que há concordância de grande parte do grupo com relação às afirmativas mesmo variando de parcialmente a fortemente. Mostrando a aplicação e criação destes tipos de sistemas é de grande valia ao curso.

6 Conclusão

O simulador apresentado neste trabalho para se mostrou eficiente e atendeu aos objetivos propostos.

Aspectos relativos ao uso destes sistemas são de caráter importante no mercado acadêmico, à criação de novas tecnologias e o conhecimento de processos industriais. Criou-se um arcabouço de conhecimentos teóricos e práticos pelos quais pode-se estar utilizando como ferramenta na melhoria do processo de ensino-aprendizagem em cursos tecnológicos.

Referências Bibliográficas

CAVALINI JR, A.A. GONSALEZ, C.G. FRANCO, V.R. *Introdução ao MatLab®*. São Paulo: 2008. Disponível em: <HTTP://www.dem.feis.unesp.br/caos/down/a_MatLab@.pdf> acessado em 24/06/2010.

OGATA, Katsuhiko. Engenharia de Controle Moderno. 4.ed. Prentice Hall – Br, 2003.800p.

VAZ, C. E. M.; MAIA, J. L. P.; SANTOS, W. G. Tecnologia da Indústria do Gás Natural. 1. ed. São Paulo, 2008.415p.

PIDD, M. *Computer Simulation in Management Science*. 5. Ed. Lancaster, United Kingdom: Editora John Wiley Prof, 2004. 328p.

SCHNAID, F., TIMM, M. I. e ZARO, M. A. Considerações sobre uso de modelo construtivista no ensino de Engenharia - disciplina de projeto, com graduandos e mestrados. In: Revista Novas Tecnologias na Educação-RENOTE, Porto Alegre, UFRGS, v. 1, n. 1, 2003.

VIANNA, W.S. *Sistema SCADA Supervisorio*. Rio de Janeiro: 2008. Disponível em: <HTTP://www.professor.cefetcampos.br/professores/wvianna/scada/supervisorio_scada.pdf/view> acessado em 24/06/2010.