

Monografia de Graduação

Desenvolvimento da Interface Gráfica para um Simulador Computacional do Sistema de Elevação por Bombeio Centrífugo Submerso

Tiago de Souza Barbosa

Natal, fevereiro de 2009



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE COMPUTAÇÃO E AUTOMAÇÃO
CURSO DE ENGENHARIA DA COMPUTAÇÃO

Desenvolvimento da Interface Gráfica para um Simulador Computacional do Sistema de Elevação por Bombeio Centrífugo Submerso

Tiago de Souza Barbosa

Orientador: Prof. Dr. André Laurindo Maitelli

Trabalho de Conclusão de Curso submetido ao Departamento de Engenharia de Computação e Automação do Centro de Tecnologia da Universidade Federal do Rio Grande do Norte como parte dos requisitos necessários para obtenção do título de Engenheiro da Computação.

Natal, RN, Dezembro de 2008

Resumo

A utilização do sistema de elevação artificial por Bombeio Centrífugo Submerso (BCS) está se expandindo na elevação artificial pela crescente flexibilidade e evolução dos equipamentos disponíveis, mas as dificuldades encontradas na manutenção e instalação dos equipamentos são fatores que oneram a produção de petróleo, principalmente em águas profundas. Com isso, a otimização da produção, via automação, pode ser um passo fundamental para a redução de custos e falhas nos equipamentos. Propõe-se o desenvolvimento de uma ferramenta computacional para simulação do conjunto BCS que seja capaz de representar o comportamento dinâmico deste método buscando proporcionar um maior domínio de conhecimento acerca do mesmo.

Como o simulador proposto tem objetivos didáticos, e busca facilitar o entendimento e visualização dos fenômenos simulados como também seu uso em treinamentos técnicos, o mesmo deve possuir uma interface gráfica amigável permitindo a visualização do comportamento do sistema da melhor forma possível, para isso usando-se de animações tridimensionais, gráficos e etc.

Palavras-chave: Elevação Artificial, Bombeio Centrífugo Submerso, Simulação Computacional, Interface Gráfica, Animações Tridimensionais.

Sumário

Sumário	i
Lista de Figuras	iii
Lista de Símbolos e Abreviaturas	iv
1 Introdução	1
1.1 Bombeio Centrífugo Submerso	3
1.2 Motivação	10
1.3 Objetivos	11
1.4 Organização do trabalho	11
2 Simulação Computacional	12
2.1 Modelagem Matemática	12
2.2 Escalonamento no Tempo	13
2.2.1 Simulação Orientada a Evento	13
2.2.2 Simulação Orientada a Tempo	14
2.3 Vantagens e Desvantagens	15
3 Simulador BCS	16
3.1 A Arquitetura MVC	16
3.1.1 Aplicando o MVC ao Simulador	18
3.2 Padrões de Projetos de Software	20
3.2.1 O Padrão Singleton	22
3.3 Animações Tridimensionais	22
3.3.1 Descritores de Cenas Tridimensionais em Arquivo	23
3.3.2 Otimização de Modelos 3D	26
3.4 A Interface Gráfica do Simulador	27
4 Conclusões	34

Lista de Figuras

1.1	Bomba centrífuga de múltiplos estágios. d) Difusor. e) Impelidor.	3
1.2	Motor elétrico trifásico.	4
1.3	Sêlo mecânico.	5
1.4	Cabos elétricos. a) Cabo chato. b) Cabo redondo.	6
1.5	Variador de Frequência (VFD).	7
1.6	Poço produtor por Bombeio Centrífugo Submerso	9
3.1	Interação entre as camadas MVC.	17
3.2	Divisão das Camadas MVC para o simulador.	19
3.3	Modelagem de uma peça no SolidWorks.	24
3.4	Construção de um modelo 3D no Blender.	25
3.5	Modelos carregados pela Glscene e renderizado com OpenGL.	26
3.6	Optimização de modelo com VizUp Optimize: a) modelo original, modelo com b) 60%, c) 80% e d) 95% de redução	28
3.7	Tela principal do simulador BCS	30
3.8	Tela de Configuração da Bomba	32
3.9	Tela de configuração do Reservatório	33

Lista de Símbolos e Abreviaturas

API:	grau API do fluido
BCP:	Bombeio por Cavidades Progressivas
BCS:	Bombeio Centrífugo Submerso
BM:	Bombeio Mecânico com Hastes
GLC:	Gás Lift Contínuo
GLI:	Gás Lift Intermitente
GoF:	Gang of Four
GRASP:	General Responsibility Assignment Software Patterns
GPL:	GNU General Public License
MVC:	Model - View - Controller
OOPSLA:	Object Oriented Programming, Systems, Languages, and Applications
PL:	Plunger Lift
RGO:	razão gás-óleo
WYSIWYG:	”What You See Is What You Get”

Capítulo 1

Introdução

Segundo [Thomas 2004], o caminho do petróleo, desde as pesquisas para sua descoberta até sua chegada a uma refinaria envolve inúmeros especialistas tais como: geólogos, geofísicos e engenheiros.

Até ser colocada em produção, uma jazida passa por etapas como: perfuração do poço de petróleo, avaliação das rochas perfuradas e estimativas da potencialidade econômica da eventual descoberta, a completação do poço, ou seja, instalação dos equipamentos necessários para colocar o poço em produção, estudos da jazida e projetos de drenagem do petróleo com maior produtividade e menor custo, técnicas de elevação do petróleo do subsolo até a superfície e o tratamento e condicionamento do petróleo produzido para transferência para a refinaria, tratamento dos efluentes e controle do meio ambiente [Almeida 2006].

A fase de extração do petróleo começa após a avaliação da extensão da jazida. Em cada poço é introduzida uma tubulação de aço na superfície até o fundo, chamada de revestimento. Os fluidos que migram da rocha geradora são extraídos através de uma coluna de produção - tubulação de menor diâmetro introduzida no revestimento, enquanto que o controle da vazão espontânea desses fluidos é realizado pela árvore de natal - nome dado ao equipamento composto por um conjunto de válvulas instalado na superfície do poço.

A elevação natural ocorre quando a pressão do reservatório é suficientemente elevada, e os fluidos nele existentes alcançam livremente a superfície. Os poços que produzem desta forma são denominados de poços surgentes.

Quando a pressão do reservatório é relativamente baixa, os fluidos não alcançam a superfície sem que sejam utilizados meios artificiais para elevá-los. O mesmo ocorre no final da vida produtiva por surgência ou quando a vazão do poço está muito abaixo do que poderia produzir, necessitando de uma suplementação da energia natural através de elevação artificial [Thomas 2004].

Equipamentos de elevação artificial de petróleo são aqueles que visam proporcionar e manter a produção de óleo de um poço ou de um conjunto de poços, visando maximizar o retorno financeiro do projeto de produção, durante as diferentes fases que compõem a vida de um poço, do campo ou do reservatório de petróleo [Bezerra 2002].

O propósito da aplicação de métodos de elevação artificial em poços de petróleo é garantir a manutenção do escoamento de óleo da formação (reservatório) para o poço, e deste até a superfície, superando as perdas de carga ao longo da coluna ou tubo de produção [Bezerra 2002].

Alguns dos métodos de elevação apresentaram grande desenvolvimento nas últimas décadas, sendo hoje considerados por especialistas de produção como tecnologias consagradas. Entre estes estão:

- *Gas-Lift* Contínuo e Intermitente (GLC e GLI);
- Bombeio por Cavidades Progressivas (BCP);
- Bombeio Mecânico com Hastes (BM);
- Bombeio Centrífugo Submerso (BCS);
- Plunger Lift (PL).

A avaliação do método de elevação para determinada aplicação representa um passo importante no gerenciamento da produção de petróleo. As diferentes tecnologias existentes apresentam vantagens e desvantagens específicas em termos de projeto e configuração dos equipamentos, custos de aquisição, confiabilidade, procedimentos de operação, intervenção e reparos.

As atividades de análise e seleção de um método de elevação consideram a pesquisa e organização de um amplo número de informações e parâmetros relativos ao reservatório, ao projeto do poço e às características dos fluidos que serão produzidos.

As informações consideradas na avaliação dos métodos de elevação estão presentes na literatura além de considerações de especialistas, técnicos e engenheiros, envolvidos com a produção de poços.

Novas tecnologias de elevação de petróleo vêm sendo desenvolvidas nos últimos anos, ampliando-se assim o leque de opções existentes face às necessidades de produção mais recentes como a ocorrência de reservatórios marítimos localizados sob lâminas d'água cada vez mais profundas, a exploração de óleos pesados, a utilização de projetos de poços multi laterais e etc [Bezerra 2002].

Neste capítulo introdutório apresentaremos as principais características relacionadas ao bombeio centrífugo submerso. No final do capítulo são descritos a motivação e os objetivos do trabalho, além da organização estrutural dos outros capítulos.

1.1 Bombeio Centrífugo Submerso

O sistema BCS trata-se, basicamente, de uma bomba centrífuga de múltiplos estágios, acionada por um motor elétrico, sendo este conjunto motor-bomba fixado na extremidade da coluna do poço. Dessa forma, o conjunto fica submerso no óleo do reservatório e o seu funcionamento cria um incremento de pressão no fundo do poço de modo a se obter a vazão desejada de óleo na superfície.

A bomba centrífuga é representada na Figura 1.1, e consiste de impelidores centrífugos acionados por um eixo dentro de um corpo tubular com difusores. O comprimento normal da bomba é de 30 ft (9 m). Segundo [L.Wilson 1994], uma das principais limitações da bomba BCS é sua inabilidade de operar com eficiência com quantidades significativas de gás. Para minimizar os problemas com gás, é adicionado a admissão da bomba um separador de gás, que busca evitar que o gás vindo do reservatório alcance a bomba centrífuga, quando não, pelos menos quebra as bolhas de gás diminuindo seu tamanho e seu efeito sobre a bomba. A instalação de separadores minimiza o problema da interferência de gás, mas pode significar simultaneamente numa redução na pressão da bomba, vibração, erosão e até perda de eixo, quando em presença de areia.

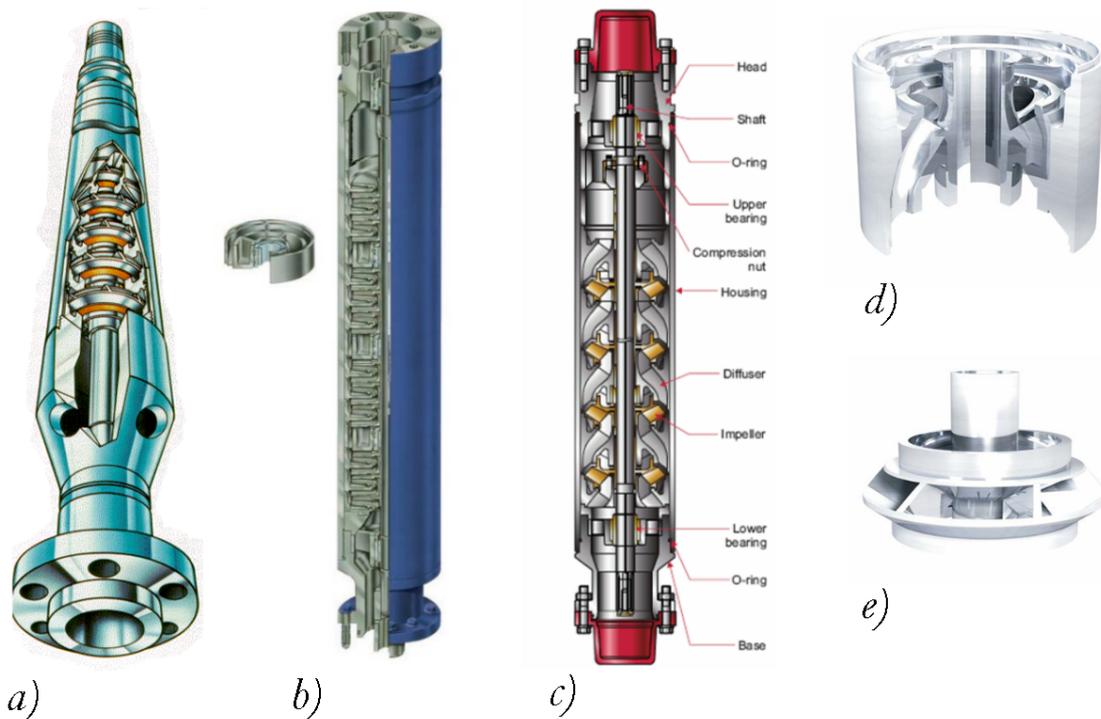


Figura 1.1: Bomba centrífuga de múltiplos estágios. d) Difusor. e) Impelidor.

O motor elétrico é trifásico, como o representado na Figura 1.2, e giram com uma velocidade de 3500 rpm, com corrente de 60 Hz, e é preenchido com óleo mineral refinado que deve prover resistência dielétrica, lubrificação dos mancais, e boa condutividade térmica, transferindo o calor gerado no motor para o anular (espaço entre o conjunto motor-bomba e o revestimento do poço).

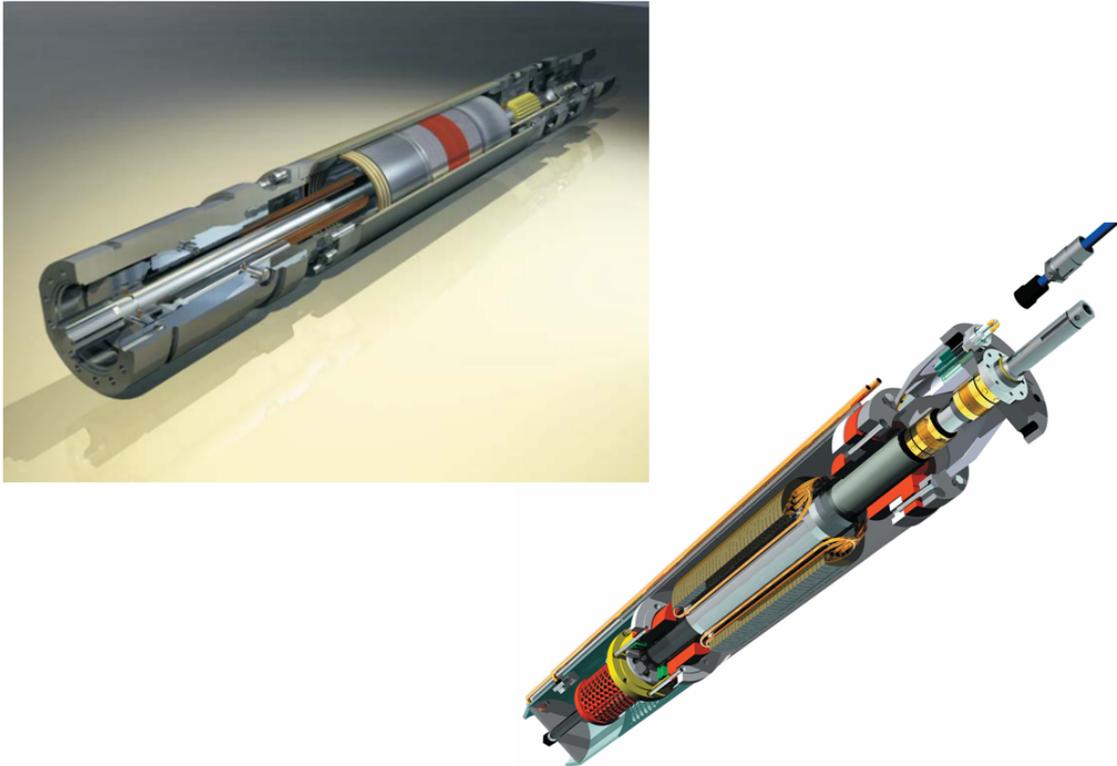


Figura 1.2: Motor elétrico trifásico.

Para evitar a contaminação do óleo mineral com o óleo de produção, é adicionado entre o motor e o rotor da bomba um sistema de selagem mecânica. O selo mecânico, como o mostrado na Figura 1.3, possibilita algumas funções importantes no BCS como suportar o eixo da bomba e cargas hidráulicas, e age como uma barreira de proteção do motor separando o óleo do motor dos fluidos produzidos.

Em poços a altas profundidades, a temperatura no fundo do poço torna-se elevada e aumenta-se, então, a ocorrência de superaquecimento do motor, deve se ter a preocupação com a disposição dos componentes para que o óleo de produção sendo deslocado passe lateralmente a superfície do conjunto motor-selo resfriando-o. Normalmente os componentes BCS são colocados um pouco acima do canhoneado de tal forma que o óleo de

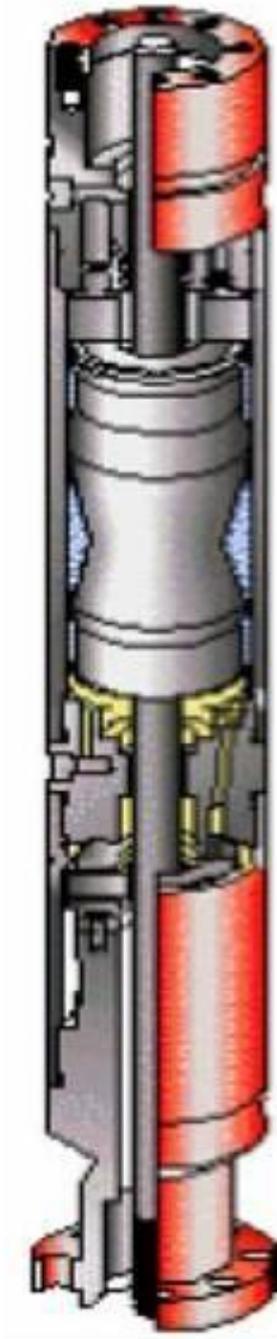


Figura 1.3: Sêlo mecânico.

produção saindo do reservatório passe, obrigatoriamente, lateralmente ao motor.

A energia elétrica é levada da superfície até o motor por cabos elétricos que passam lateralmente aos outros componentes do sistema, por isso alguns componentes, como o tubo de produção, devem ter seus diâmetros externos pequenos para permitir a passagem do cabo pelo anular. Outra estratégia adotada é o uso de dois tipos de cabos elétricos, um redondo para os locais de maior espaço anular e um chato para as regiões de menor espaço. A Figura 1.4 mostra os dois modelos de cabos usados.



Figura 1.4: Cabos elétricos. a) Cabo chato. b) Cabo redondo.

Ao longo do tubo de produção, uma válvula de retenção é instalada visando manter os fluidos dentro da bomba quando fora de operação. O retorno de fluidos gera rotação reversa do motor, o que pode causar aprisionamento do mesmo quando da nova partida, sendo esta uma das causas de queima do motor, do cabo elétrico, da caixa de controle na superfície e de fusíveis [Bezerra 2002].

As falhas mais comuns verificadas no BCS têm sido de natureza elétrica. Segundo [Kunkel 2000] estas representam cerca de 67% dos casos em geral, especialmente ocorrências no cabo elétrico, principalmente nas operações de partida e de parada de equipa-

mento, quando a corrente no cabo pode atingir valores de 5 a 7 vezes maiores do que a corrente normal de operação. Com o intuito de amenizar os problemas de partida, usa-se, então, um variador de frequência (VFD) que possibilita que se obtenha inicialmente uma partida lenta e um aumento gradual da frequência até a frequência da rede. O uso de VFDs possibilita também um maior controle da velocidade do motor e, conseqüentemente, da vazão do poço. Na Figura 1.5 temos a imagem de um VFD da Weatherford.



Figura 1.5: Variador de Frequência (VFD).

Melhorias tecnológicas têm permitido a menor ocorrência de falhas no BCS. Parte destas melhorias são o melhor controle de sólidos, isolamento dos transformadores, melhor especificação dos cabos elétricos, a introdução de sensores de fundo do poço, motores

para altas temperaturas e estudos para aumentar a capacidade da BCS para operar com sólidos. Quanto aos sensores, eles poderão medir os efeitos de gás livre através da bomba, além de estudar o efeito da viscosidade sobre o desempenho da bomba. Para temperaturas até 400 (graus Celcius), pesquisas estão em desenvolvimento para utilização de motores com injeção de óleo a partir da superfície, visando facilitar a refrigeração do mesmo. Em anos mais recentes, uma grande ênfase tem sido dada no sentido de ampliar a durabilidade do BCS, e aumentar a capacidade do mesmo para operar com escoamento multifásicos [Bezerra 2002].

De acordo com [Thomas 2004], a utilização do BCS está se expandindo na elevação artificial pela crescente flexibilidade e evolução dos equipamentos disponíveis. [Allis & Capps 1984] cita a utilização de BCS em poços desviados com alta vazão de fluidos motivada por sua grande flexibilidade de produção. [Clegg 1988] trata o BCS como um método de elevação comumente especificado para altas vazões de fluido. O BCS tem sido utilizado tanto em aplicações onshore como offshore, em condições adversas de temperatura, em poços com presença de fluidos viscosos e ambientes gaseificados.

Como se pode notar ante a descrição acima e a representação mostrada na Figura 1.6, um sistema BCS é formado por vários componentes arranjados logicamente na forma de um sistema em série. Geralmente, pode-se subdividir o sistema BCS de um poço submarino em dois conjuntos de equipamentos: os de subsuperfície, situados no interior do poço, e os de superfície, localizados na plataforma de produção. Esta mesma subdivisão pode ser usada para um poço terrestre, sendo que neste caso, os equipamentos de superfície localizam-se na área de produção da instalação.

1. Equipamentos de Subsuperfície são:

- Bomba centrífuga de múltiplos estágios, onde cada estágio consiste de um impelidor (girante) e um difusor (estacionário);
- Separador ou intake - evita a sucção de gás livre pela bomba;
- Motor elétrico;
- Protetor ou Selo - equaliza a pressão interna do motor com a pressão do fluido produzido pelo poço;
- Cabo elétrico.

2. Equipamentos de Superfície são:

- Quadros de comando;
- Transformador;
- Cabeça de produção;
- Caixa de ventilação;

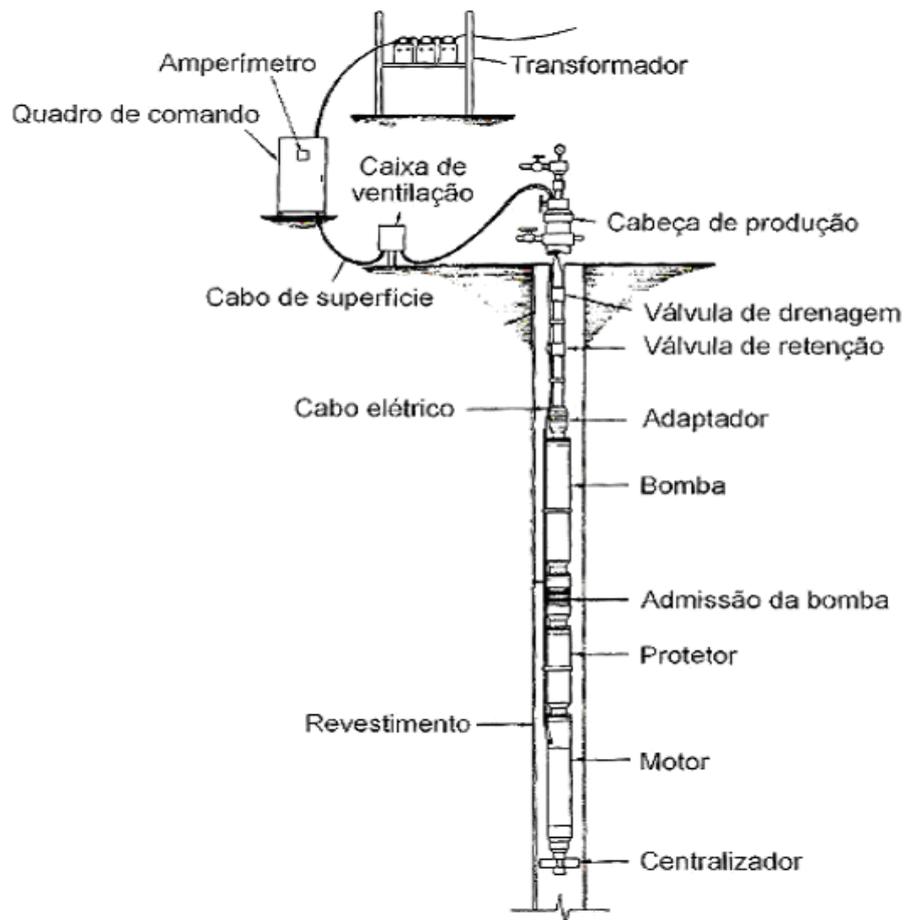


Figura 1.6: Poço produtor por Bombeio Centrífugo Submerso

- Válvula de retenção;
- Válvula de drenagem ou de alívio;
- Sensores de pressão e temperatura de fundo.

1.2 Motivação

O BCS requer projeto de poço adequado, bom treinamento do pessoal operacional e de manutenção, seleção cuidadosa do equipamento, fabricantes confiáveis, instalação correta, suprimento de energia confiável, condições estáveis de operação. Como todo tipo de bomba, o BCS também é afetado significativamente pela presença de areia e gás livre [Bezerra 2002].

É necessário que haja um bom conjunto de dados relativos ao poço afim de promover um bom dimensionamento e seleção do tipo de equipamento a ser instalado. Uma BCS mal dimensionada acarretará menor vida útil do equipamento.

A vazão produzida pelo poço equipado com o BCS depende de características do reservatório, de propriedades dos fluidos, da geometria interna da bomba, da rotação do motor e das pressões de sucção e descarga, sendo esta ajustável pela modificação da abertura do beam. Pode-se variar a vazão produzida, dentro de certos limites, ajustando-se a abertura do beam ou variando-se a rotação da bomba centrífuga, via ajuste da frequência de alimentação elétrica do motor. Há interesse em que a vazão obtida seja a ótima, do ponto de vista técnico-econômico, o que implica, muitas vezes em ajustes no beam instalado na superfície ou ajustes na frequência do inversor que alimenta o motor elétrico.

Atualmente estes ajustes são realizados pelo operador com base na leitura de diversos parâmetros operacionais, dentre os quais a pressão de fundo. Diante dessa realidade, pesquisas tem sido realizadas buscando implementar um controle automático eficaz do método de elevação BCS, de modo a otimizar a vazão produzida e proteger o equipamento contra falhas precoces devido a operação inadequada.

Objetivando minimizar as dificuldades, anteriormente citadas, com o sistema BCS, cresce a necessidade da utilização de ferramentas que auxiliem no entendimento do sistema, no projeto de poço, na validação de algoritmos controladores e etc. e para tal propósito uma ferramenta poderosa é um simulador computacional.

Um simulador computacional, em geral, é uma ferramenta de grande potencial, tanto de caráter didático, auxiliando na capacitação técnica dos operadores do sistema, quanto para auxílio na engenharia de produção, permitindo realizar uma análise mais detalhada da dinâmica operacional dos poços de petróleo equipado com o método de elevação.

1.3 Objetivos

Este trabalho visa apresentar o desenvolvimento da interface gráfica de um simulador computacional para o conjunto BCS. Este simulador será capaz de representar o comportamento dinâmico deste método, considerando o modelo da fonte e da transmissão de energia elétrica para o motor, o modelo do motor elétrico BCS (incluído o cálculo térmico), simulações do escoamento e também disponibilizará animações tridimensionais de cada subsistema BCS (motor, bomba, selo, entre outros), visando facilitar a visualização e compreensão do processo.

O simulador propõe uma melhoria no monitoramento de poços petrolíferos através de um melhor conhecimento dos fenômenos físicos pertinentes ao processo, podendo prevenir falhas, facilitando tomadas de decisão numa menor parcela de tempo. Assim também, permitindo controlar o aumento da produção de óleo, através de ajustes de condições operacionais (frequência, pressão na cabeça do poço, pontos de operações, etc.).

1.4 Organização do trabalho

Este trabalho está organizado em 4 capítulos. O Capítulo 2, traz uma breve revisão bibliográfica sobre simulação computacional. Os resultados preliminares da proposta do simulador são encontrados no Capítulo 3. Neste mesmo capítulo são destacadas algumas questões relacionadas a engenharia de software necessárias ao desenvolvimento do simulador, como também os resultados obtidos e atividades realizadas.

Capítulo 2

Simulação Computacional

Já nos primórdios, sabia-se que simular uma realidade era uma forma de ampliar o conhecimento e avançar no aprimoramento dos produtos.

A técnica de Simulação Computacional consiste em estabelecer um modelo capaz de descrever ou representar o problema real a ser submetido a manipulação *experimental* em um computador. Em outras palavras, Simulação Computacional consiste em conduzir *experimentos* em um computador, envolvendo relações de conteúdo lógico e matemático, necessários à descrição do comportamento e da estrutura de um problema real, em períodos de tempo bem definidos [Abreu & de Assis Rangel 1999].

Neste sentido, a Simulação visa a compreender o funcionamento, prever o desempenho, o comportamento, dominar e controlar a evolução do sistema real, através da avaliação de estratégias a serem adotadas dentro das restrições impostas pela realidade. A simulação é obtida através do uso de modelos, que descrevem de forma segura e confiável esse problema real.

Um modelo é uma abstração ou uma aproximação, que é usada para entender a realidade. Nele, são ressaltados ou destacados os pontos essenciais do problema que se deseja analisar através da simulação computacional.

2.1 Modelagem Matemática

Há muito tempo as pessoas têm usado modelos. Uma descrição escrita de uma batalha, uma réplica de um prédio, o uso de símbolos para representar dinheiro, números e relações matemáticas são todos exemplos de modelos.

Modelos podem ser utilizados na arquitetura através de maquetes e plantas; na geografia através dos mapas; na aviação e exploração espacial, através dos modelos em escala e o efeito dos simuladores de voo. Na ciência, em geral, os modelos são de natureza matemática, estatística e/ou computacional, constituídos com o uso de equações e relações e

usados para simulação em computador.

Para a realização de uma modelagem é preciso inicialmente conhecer a natureza do problema, tais como: características físicas, químicas, biológicas, etc. Após esse processo, faz-se um modelo matemático que represente a tradução do problema real [Abreu & de Assis Rangel 1999].

Depois de modelado o problema através das técnicas matemáticas e/ou computacionais, os resultados devem ser investigados para ver a correspondência com o problema original. A esta fase de análise de resultados chamamos de *validação de modelos*.

Após ultrapassada a fase de modelagem, tendo-se confiança no modelo desenvolvido, ou seja, o modelo desenvolvido realmente traduz a realidade da melhor forma possível, podem simular situações mais complexas do problema em estudo.

2.2 Escalonamento no Tempo

Geralmente o tempo de simulação computacional de um evento é menor que o tempo do evento no mundo real. Muitos fenômenos de processos reais podem durar dias, meses ou anos para ocorrerem, porém, com o advento da simulação computacional, este tempo pode ser reduzido para frações de tempo muito pequenas. Este fenômeno ocorre, por exemplo, com os métodos de elevação artificial de petróleo, que em geral possuem uma dinâmica lenta. Com isto, situações que poderiam ser impraticáveis de se analisar, devido ao tempo, podem, através de simulação, ser estudadas de forma robusta e por repetidas vezes [Nascimento 2005].

Os fatos acima citados fazem do escalonamento temporal uma das maiores vantagens das simulações computacionais. Nas simulações computacionais o tempo pode ser controlado, comprimido, expandido. Permitindo reproduzir fenômenos de maneira lenta ou acelerada. Por esses fatos é que a representação do tempo se torna essencial para simular fenômenos reais com flexibilidade sem perder a sua credibilidade.

O ritmo de uma simulação pode depender dos próprios eventos, o que caracteriza uma simulação orientada a eventos, ou dos incrementos fixos de tempo gerenciados por um relógio.

2.2.1 Simulação Orientada a Evento

A simulação orientada a evento utiliza uma lista onde os eventos são armazenados por ordem cronológica de acontecimento. Os eventos vão sendo disparados no meio simulado de forma sequencial, fazendo com que a representação do tempo seja apenas em função do

atributo tempo de ocorrência dos eventos. Caso estes eventos não possuam uma variável explícita de tempo de acontecimento, eles devem ser organizados simplesmente por suas ordens de acontecimentos.

A cada evento disparado, o relógio é incrementado com o atributo tempo do evento. O evento, após ser disparado, pode criar conseqüências ou alterações no estado do sistema, necessitando que outros eventos precisem ser disparados logo em seguida, ou numa ordem diferente da usual. Um método para reordenar a lista de eventos deve ser ativado cada vez que o sistema de simulação solicitar. O avanço do tempo em simulações orientadas a evento, dar-se de forma discreta, o que permite que a evolução do tempo seja dada aos pulsos, e não de uma forma contínua [Pinheiro et al. 2008].

2.2.2 Simulação Orientada a Tempo

Na simulação orientada a tempo, o relógio conta o tempo continuamente, sem saltos, onde o evento será executado, somente, quando a sua hora estiver de acordo com a do relógio. Já que não há saltos, é necessário uma taxa de aceleração para aumentar a velocidade do relógio, diminuindo o pulso do relógio, Essa alta freqüência diminui o tamanho do pulso, obrigando que tarefas, antes realizadas em um único pulso, passem a usar vários, retardando o efeito de simulação.

A outra forma de ditar o ritmo de uma simulação, é através de um relógio. Neste caso, o relógio vai ser o componente ativo na tomada de decisão de qual evento será ativado e, principalmente, no controle do tempo entre esses eventos. Uma simulação orientada a tempo é capaz de retornar os estados do sistema em qualquer instante de tempo, visto que, o tempo, agora corre continuamente, sem saltos. O tempo é incrementado de unidade por unidade, sendo esta estabelecida de acordo com a exigência da simulação. Essa unidade, que irá ser incrementada unidade por unidade, deve ser compatível ou proporcional com os atributos de tempo das ocorrências dos eventos e que seja a menor unidade do relógio.

Quanto maior a taxa de aceleração aplicada a esse relógio, menor o seu pulso e com isso mais rápido será a simulação. Um relógio usando uma altíssima taxa de aceleração tem um pulso muito pequeno e com isso consegue realizar muitos pulsos em um único segundo real. A necessidade do relógio de gerenciar essa grande quantidade de pulsos detém grande parte do processamento, o que acarreta uma lentidão para as outras partes envolvidas na simulação, como, por exemplo, o tempo de respostas dos agentes [Pinheiro et al. 2008].

2.3 Vantagens e Desvantagens

Simular um sistema real, talvez seja a melhor forma de prever um comportamento futuro, sem a necessidade de testar ou expor o sistema real a experimentos. Deste modo, com o avanço da tecnologia de sistemas e com a evolução das linguagens de programação, é possível aplicar a técnica da simulação em todas as áreas de conhecimento como, por exemplo, em sistemas de atendimentos bancários, análise de desempenho em uma linha de produção e no auxílio na gerência de terminais de transportes. Enfim, a todas as áreas as quais sejam necessário um auxílio para tomadas de decisões de forma clara e condizentes com o mundo real.

Pela simulação computacional é possível saber, com antecedência, os pontos fracos do projeto, e corrigi-los antes mesmo de começar a fabricar o produto. Atualmente, graças ao aumento na capacidade de processamento dos computadores, as simulações têm ficado mais completas e complexas, mesmo para materiais não-lineares

Segundo [Nascimento 2005], embora os simuladores possuam inúmeras vantagens, existem também algumas limitações, as quais não estão relacionadas ao modelo imposto no simulador, e sim a expectativas geradas sobre o simulador. As limitações podem ser:

- uma simulação não pode dar resultados precisos quando os dados de entrada foram imprecisos;
- a simulação não pode dar respostas fáceis a problemas complexos;
- a simulação não pode solucionar problemas por si só.

O termo utilizado na literatura para os resultados imprecisos de uma simulação, devido a parâmetros de entrada incorretos, é *garbage in, garbage out* (lixo entrando, lixo saindo), ou seja, não adianta o modelo estar perfeitamente correto se os dados de entrada não forem confiáveis. Por isto, muitos simuladores hoje tentam realizar rotinas de verificação e consistência de dados para exporem possíveis erros ao usuário.

Capítulo 3

Simulador BCS

Com o intuito de acelerar o desenvolvimento do software simulador, duas frentes de trabalhos foram criadas: uma buscando modelar matematicamente o sistema BCS e seus componentes, e outra objetivando desenvolver a interface gráfica com o usuário e suas animações tridimensionais. A proposta do presente trabalho é o desenvolvimento da interface gráfica do simulador computacional para o sistema BCS. Os modelos matemáticos para o simulador computacional são encontrados nos trabalhos de [Batista et al. 2008] e [Batista 2008].

O núcleo simulador e a interface gráfica com o usuário se relacionam fortemente e modificações em um deles, normalmente, acarretariam a necessidade de se fazer modificações no outro. Se o software não fosse desenvolvido para suportar essas frequentes alterações, essas mudanças, mesmo que pequenas, se tornariam cada vez mais dispendiosas. Como medida visando garantir o desenvolvimento do simulador de forma rápida, com alta qualidade e flexibilidade para as alterações de requisitos (facilidade de manutenção), foram adotados alguns padrões de projeto e arquitetura de software.

3.1 A Arquitetura MVC

Com o aumento da complexidade das aplicações desenvolvidas torna-se fundamental a separação entre os dados (Model) e o layout (View). Desta forma, alterações feitas no layout não afetam a manipulação de dados, e estes poderão ser reorganizados sem alterar o layout.

O model-view-controller resolve este problema através da separação das tarefas de acesso aos dados e lógica de negócio, lógica de apresentação e de interação com o utilizador, introduzindo um componente entre os dois: o Controller. MVC é usado em padrões de projeto de software, mas MVC abrange mais da arquitetura de uma aplicação do que é típico para um padrão de projeto.

Várias arquiteturas de software dividem a aplicação em desenvolvimento em camadas separadas. O padrão MVC também o faz, e em três camadas a camada de apresentação ou Model, view e o controller, como mostrado na Figura 3.1.

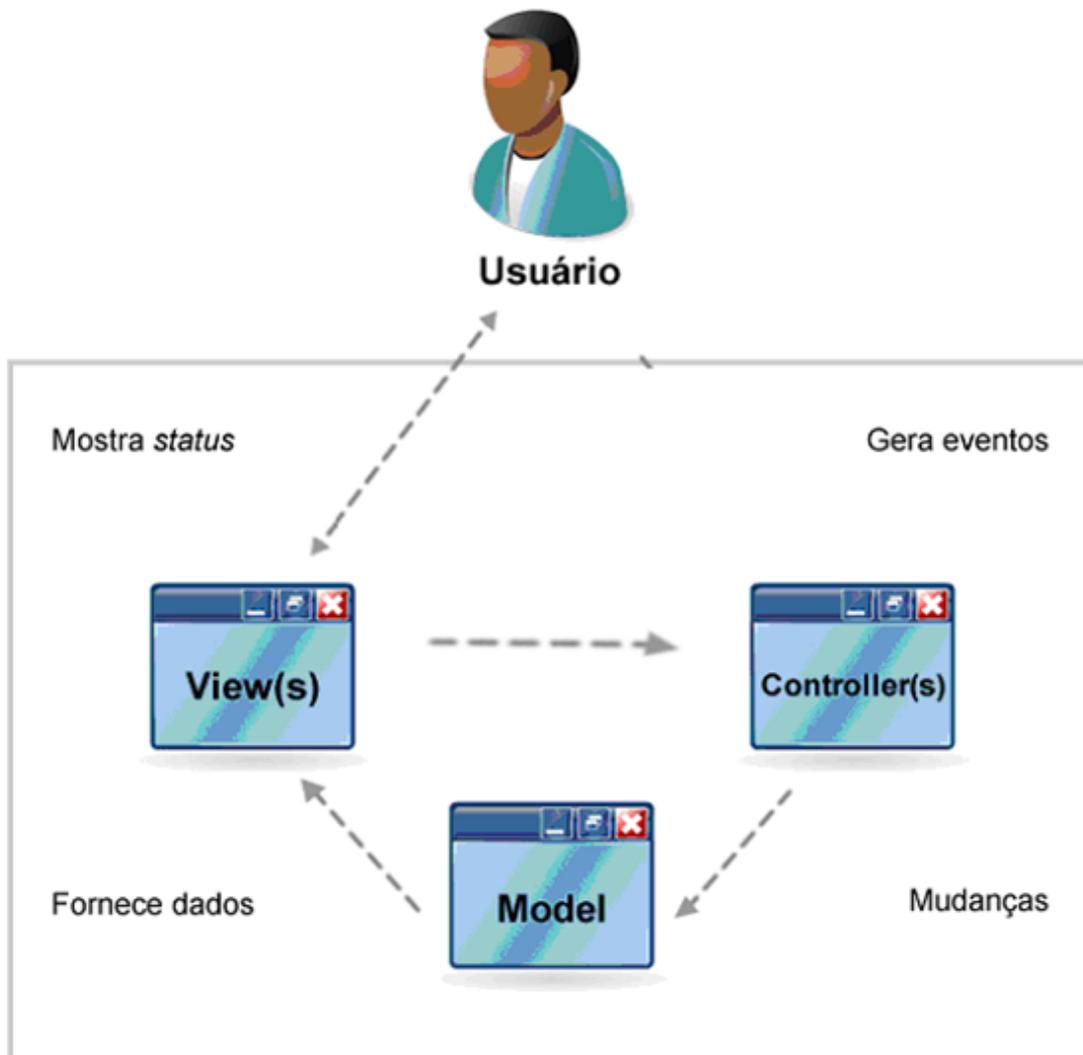


Figura 3.1: Interação entre as camadas MVC.

Model : Representa o "domínio" específico da informação em que a aplicação opera. Por exemplo, aluno, professor e turma fazem parte do domínio de um sistema acadêmico. É comum haver confusão pensando que Model é um outro nome para a camada de domínio. A Lógica de domínio adiciona sentido à dados crus (por exemplo, calcular se hoje é aniversário do usuário, ou calcular o total de impostos e fretes sobre um determinado carrinho de compras). Muitas aplicações usam um mecanismo de armazenamento persistente

(como banco de dados) para armazenar dados. MVC não cita especificamente a camada para acesso aos dados, porque subentende-se que estes métodos estariam encapsulados pelo Model.

View : "Renderiza" o model em uma forma específica para a interação, geralmente uma interface de usuário.

Controller : Processa e responde a eventos, geralmente ações do usuário, e pode invocar alterações no Model.

MVC é muito visto também em aplicações para Web, onde a View é geralmente a página HTML, e o código que gera os dados dinâmicos para dentro do HTML é o Controller. E, por fim, o Model é representado pelo conteúdo de fato, geralmente armazenado em bancos de dados ou arquivos XML.

Ainda que existam diferentes formas de MVC, o controle de fluxo geralmente funciona como segue:

- O usuário interage com a interface de alguma forma (por exemplo, o usuário aperta um botão)
- O Controller manipula o evento da interface do usuário através de uma rotina pré-escrita.
- O Controller acessa o Model, possivelmente atualizando-o de uma maneira apropriada, baseado na interação do usuário (por exemplo, atualizando os dados de cadastro do usuário).
- Algumas implementações de View utilizam o Model para gerar uma interface apropriada (por exemplo, mostrando na tela os dados que foram alterados juntamente com uma confirmação). O View obtém seus próprios dados do Model. O Model não toma conhecimento direto da View.
- A interface do usuário espera por próximas interações, que iniciarão o ciclo novamente.

3.1.1 Aplicando o MVC ao Simulador

O software foi desenvolvido sobre a plataforma Builder C++. A biblioteca de componentes do Builder se utiliza de uma variação do mvc, onde o controller é implementado nas funções de captura de eventos da view (formulários).

A Figura mostra como foi dividida as classes entre as camadas.

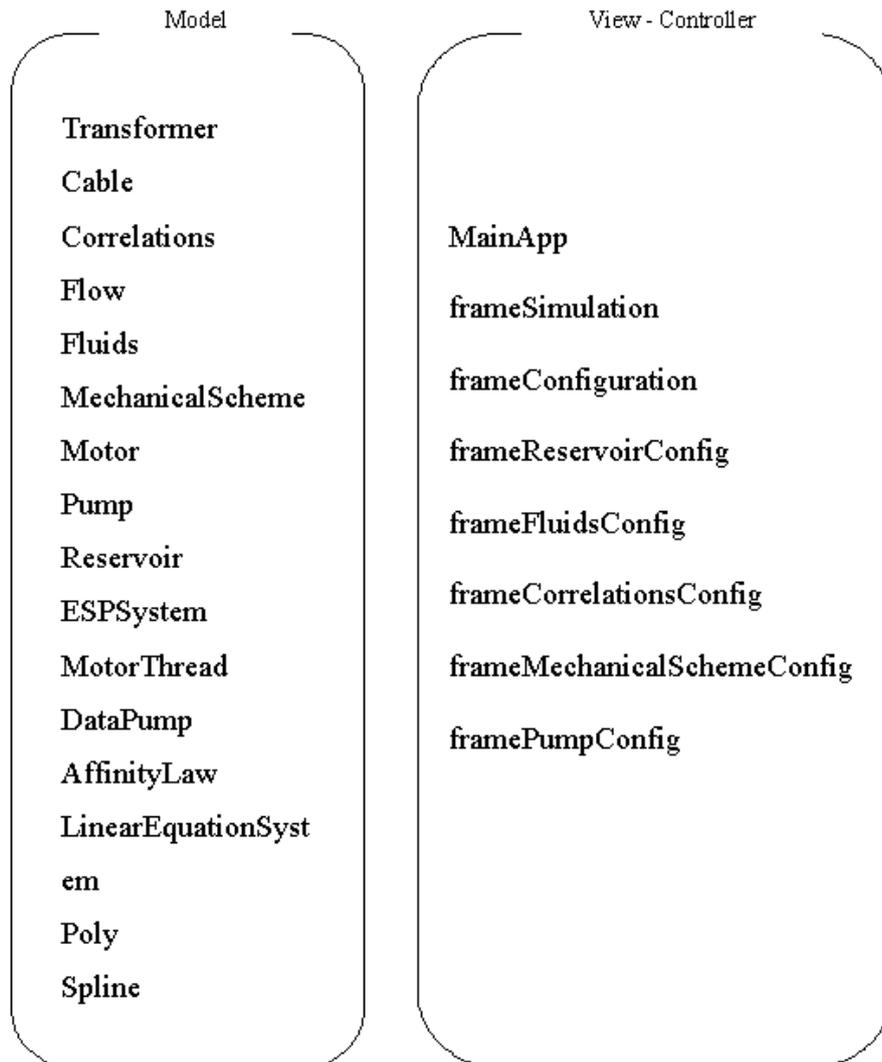


Figura 3.2: Divisão das Camadas MVC para o simulador.

3.2 Padrões de Projetos de Software

Os padrões de projeto de software ou padrões de desenho de software, também muito conhecido pelo termo original em inglês: Design Patterns, descrevem soluções para problemas recorrentes no desenvolvimento de sistemas de software orientados a objetos. Um padrão de projeto estabelece um nome e define o problema, a solução, quando aplicar esta solução e suas conseqüências.

Os padrões de projeto visam facilitar a reutilização de soluções de desenho - isto é, soluções na fase de projeto do software, sem considerar reutilização de código. Também acarretam um vocabulário comum de desenho, facilitando comunicação, documentação e aprendizado dos sistemas de software.

O conceito de padrão de projeto foi criado na década de 70 pelo arquiteto Christopher Alexander. Em seus livros *Notes on the Synthesis of Form*, *The Timeless Way of Building* e *A Pattern Language*, ele estabelece que um padrão deve ter, idealmente, as seguintes características:

- encapsulamento; um padrão encapsula um problema/solução bem definido. Ele deve ser independente, específico e formulado de maneira a ficar claro onde ele se aplica.
- generalidade; todo padrão deve permitir a construção de outras realizações a partir deste padrão.
- equilíbrio; quando um padrão é utilizado em uma aplicação, o equilíbrio dá a razão, relacionada com cada uma das restrições envolvidas, para cada passo do projeto. Uma análise racional que envolva uma abstração de dados empíricos, uma observação da aplicação de padrões em artefatos tradicionais, uma série convincente de exemplos e uma análise de soluções ruins ou fracassadas pode ser a forma de encontrar este equilíbrio.
- abstração; os padrões representam abstrações da experiência empírica ou do conhecimento cotidiano.
- abertura; um padrão deve permitir a sua extensão para níveis mais baixos de detalhe.
- combinatoriedade; os padrões são relacionados hierarquicamente. Padrões de alto nível podem ser compostos ou relacionados com padrões que endereçam problemas de nível mais baixo.

Além da definição das características de um padrão, Alexander definiu o formato que a descrição de um padrão deve ter. Ele estabeleceu que um padrão deve ser descrito em cinco partes:

- nome; uma descrição da solução, mais do que do problema ou do contexto.
- exemplo; uma ou mais figuras, diagramas ou descrições que ilustrem um protótipo de aplicação.
- contexto; a descrição das situações sob as quais o padrão se aplica.
- problema; uma descrição das forças e restrições envolvidos e como elas interagem.
- solução; relacionamentos estáticos e regras dinâmicas descrevendo como construir artefatos de acordo com o padrão, freqüentemente citando variações e formas de ajustar a solução segundo as circunstâncias. Inclui referências a outras soluções e o relacionamento com outros padrões de nível mais baixo ou mais alto.

Em 1987, a partir dos conceitos criados por Alexander, os programadores Kent Beck e Ward Cunningham propuseram os primeiros padrões de projeto para a área da ciência da computação. Em um trabalho para a conferência OOPSLA (Object Oriented Programming, Systems, Languages, and Applications), eles apresentaram alguns padrões para a construção de janelas na linguagem Smalltalk. Nos anos seguintes Beck, Cunningham e outros seguiram com o desenvolvimento desta idéias.

O movimento ao redor de padrões de projeto ganhou popularidade com o livro *Design Patterns: Elements of Reusable Object-Oriented Software*, publicado em 1995. Os autores desse livro são Erich Gamma, Richard Helm, Ralph Johnson e John Vlissides, conhecidos como a "Gangue dos Quatro (Gang of Four)" ou simplesmente "GoF". Posteriormente, vários outros livros do estilo foram publicados, como *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development*, que introduziu um conjunto de padrões conhecidos como GRASP (General Responsibility Assignment Software Patterns).

Os padrões "GoF" são organizados em famílias de padrões: de criação, estruturais e comportamentais. Os padrões de criação são relacionados à criação de objetos, os estruturais tratam das associações entre classes e objetos e os comportamentais das interações e divisões de responsabilidades entre as classes ou objetos. Exemplos de padrões de criação são: Abstract Factory, Builder, Factory Method, Prototype e Singleton. Exemplos de padrões estruturais são: Adapter, Bridge, Composite, Decorator, Facade, Flyweight e Proxy. Exemplos de padrões comportamentais são: Chain of Responsibility, Command, Interpreter, Iterator, Mediator, Memento, Observer, State, Strategy, Template Method e Visitor.

3.2.1 O Padrão Singleton

Muitos projetos necessitam que algumas classes tenham apenas uma instância. Por exemplo, em uma aplicação que precisa de uma infraestrutura de log de dados, pode-se implementar uma classe cujo único objeto instanciado fosse responsável pelo log em toda a aplicação. Para tanto esse objeto teria que ser acessível em qualquer parte da aplicação, uma forma de garantir o acesso global é o uso de uma variável global, porém a variável global não garante a unicidade.

O Singleton é um padrão de projeto que visa garantir a existência de apenas uma instância de uma classe, mantendo um ponto global de acesso ao seu objeto. A solução proposta pelo padrão Singleton é tornar a própria classe responsável pela sua única instância. A classe que implementa o padrão Singleton garante o acesso à sua instância e ainda intercepta as requisições para criação de novos objetos, garantindo que nenhuma outra instância seja criada.

O software simulador quando executando a simulação necessitará do uso de três threads de execução: uma para a interface gráfica, uma para a simulação do sistema BCS como um todo e uma para o motor, pois o tempo de transitório do mesmo é bem menor que o transitório do restante do sistema.

Nessas três threads haverá a necessidade de acesso as classes Pump, Motor, Flow e Fluids, e como as mesmas também se caracterizam por possuírem apenas uma instância para a aplicação as mesmas foram implementadas como Singletons.

3.3 Animações Tridimensionais

Uma das características fundamentais e de grande relevância em um simulador que tenha fortes objetivos didáticos, e que busca facilitar o entendimento e visualização dos fenômenos simulados como também seu uso em treinamentos técnicos, é a utilização de animações para apresentar o comportamento dinâmico das principais partes de um sistema de bombeio. Estas animações propiciam ao simulador características únicas, devido ao nível de interação usuário/animação, que permite ao usuário a manipulação das animações, facilitando a visão e compreensão dos fenômenos.

As animações do simulador serão criadas através da biblioteca gráfica OpenGL. Esta biblioteca é em essência uma biblioteca de modelagem extremamente portátil e muito rápida. Através de sua utilização pode-se criar gráficos 3D com qualidade próxima de um raytracer. OpenGL utiliza algoritmos desenvolvidos e otimizados pela Silicon Graphics, Inc. uma conhecida empresa mundial, líder em computação gráfica e animação [Wright

et al. 2005].

Sem dúvida a computação gráfica apresentada em três dimensões é a melhor forma de percepção para a visão humana. Neste contexto, as animações criadas em OpenGL apresentam ambientes tridimensionais criados com ótima qualidade. Uma das grandes vantagens de se conciliar à animação com a simulação do processo é poder observar fenômenos que ocorrem durante a dinâmica do mesmo e a utilização em aplicações didáticas.

3.3.1 Descritores de Cenas Tridimensionais em Arquivo

A biblioteca OpenGL oferece as primitivas básicas para a modelagem de cenas tridimensionais e renderização das mesmas. Essas primitivas permitem a criação de pontos, linhas e polígonos. Para a criação de cenas complexas o uso de apenas essas primitivas sem nenhuma automação pode torna a atividade de construção de cenas exaustivas, com o intuito de resolver esse problema são usados então os descritores de modelos tridimensionais armazenados em arquivo.

Para utilização dessa idéia no desenvolvimento das animações é necessário a utilização de um software que permita a construção, visualização, e armazenamento em arquivo dos cenas tridimensionais, e um software ou biblioteca de funções (Loader) que possa ler a cena do arquivo e utilizar as primitivas oferecidas pelo OpenGL para renderizar e animar a mesma cena.

Vários formatos de descritores tridimensionais são conhecidos na literatura como o OBJ (Wavefront OBJ Format), o 3DS (3DStudio 3DS File Format) desenvolvido pela Autodesk, VRML e outros mais.

Objetivando a construção das cenas e objetos tridimensionais vários software estão disponíveis tanto comerciais quanto livres (OpenSource), habilitados para manipularem tanto os formatos acima citados como outros mais.

Entre os softwares comerciais podemos citar o Autodesk 3DS Max[®], Autodesk Maya, LightWave 3D, Cinema 4D, AutoCAD, SolidWorks e outros mais. Para construção dos modelos 3D, usados na implementação do simulador, foi usado o SolidWorks pelo simples fato de termos uma cópia disponível e algum tempo de experiência no uso do mesmo. A Figura 3.3 mostra uma modelagem 3D de uma peça no SolidWorks.

Entre os softwares de distribuição gratuita destaca-se o Blender desenvolvido pela Blender Foundation e sua comunidade de colaboradores via web. O Blender é um software livre distribuído sob a licença GPL (GNU General Public License). A Figura 3.4 mostra a construção de um modelo 3D no Blender.

Também são difundidos softwares e bibliotecas de funções, tanto proprietárias como

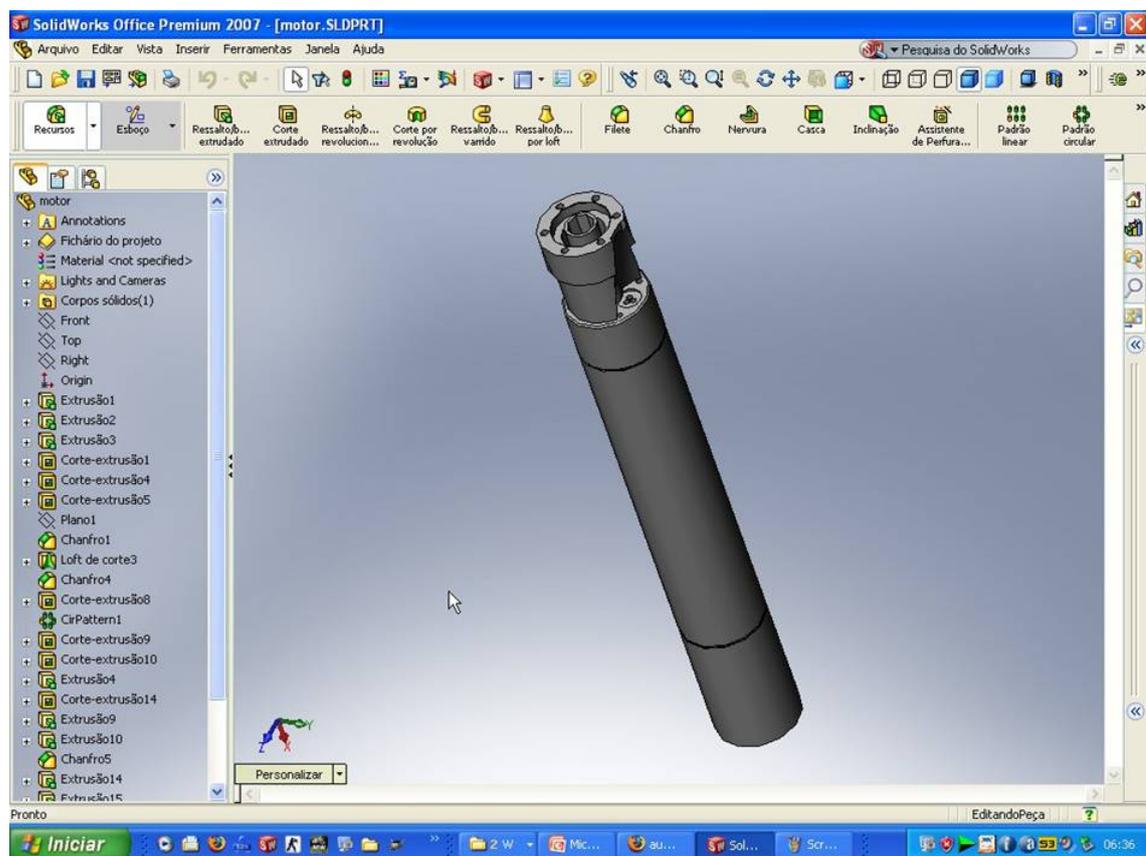


Figura 3.3: Modelagem de uma peça no SolidWorks.

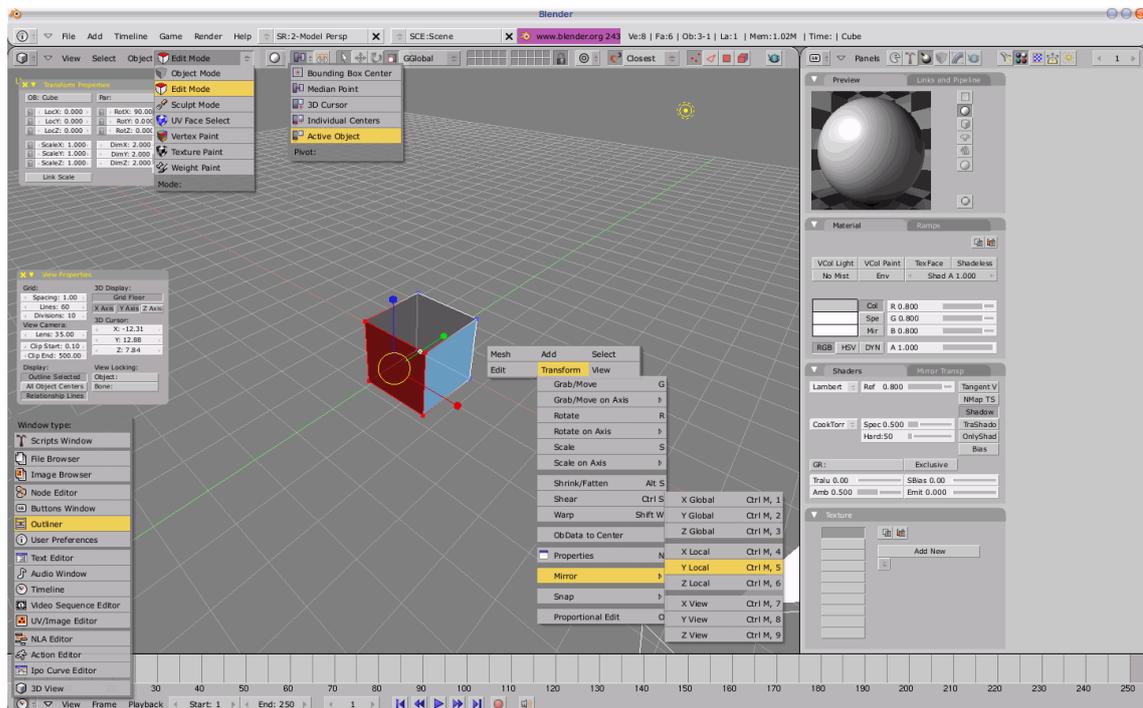


Figura 3.4: Construção de um modelo 3D no Blender.

também livres, que cumprem as funções de um Loader para esses formatos.

Um dos problemas encontrados na escolha da biblioteca a ser usada pelo projeto residia em sua incompatibilidade com a plataforma de desenvolvimento em uso (o Builder C++). Das bibliotecas livres de componentes, a que melhor mostrou compatibilidade com a plataforma Builder C++ foi a biblioteca de componentes GLscene.

A GLScene é uma biblioteca Open Source, distribuída sob a Mozilla Public License, fortemente baseada em OpenGL 3D que fornece elementos visuais e objetos que permitem a descrição e renderização de cenas tridimensionais de maneira fácil e eficiente.

Desenvolvida inicialmente para a plataforma Delphi (Borland), foi facilmente portada para FreePascal, Builder C++ (Borland) e Kylix (Borland). A GLScene tem se mostrado bem completa aos propósitos para os quais foi desenvolvida e só tem aumentado a número de usuários que a usam para o desenvolvimento de animações e jogos 3D.

A GLScene permite aos programadores criar objetos 3D em OpenGL com a visualização do mesmo em tempo de concepção (tradicional modo WYSIWYG), também oferecendo uma vasta gama de objetos e controles VCL adicionais para ajudar o programador a construir poderosas aplicações 3D para Delphi, Builder e Kylix. Ela também oferece suporte a opções variadas de renderização, ao uso de texturas com leitura a partir de arquivo de imagem, ao uso de áudio, e, de muito interesse aos objetivos do projeto, a constru-

ção de objetos a partir de modelos em arquivos nos formatos 3DS (3DStudio), MD2, MD3 (Quake2/3, animação), OBJ (WaveFront), SMD (Half-Life), NMF (Normal Mapper File), GTS (GNU Triangulated Surface) e muitos outros. As figuras abaixo mostram alguns subsistemas BCS já modelados e renderizados com o uso da GLScene.

A Figura 3.5 mostra alguns modelos (um estágio da bomba BCS, o sistema BCS completo e a arvore de natal de um poço) carregado via Glscene e renderizados com OpenGL,

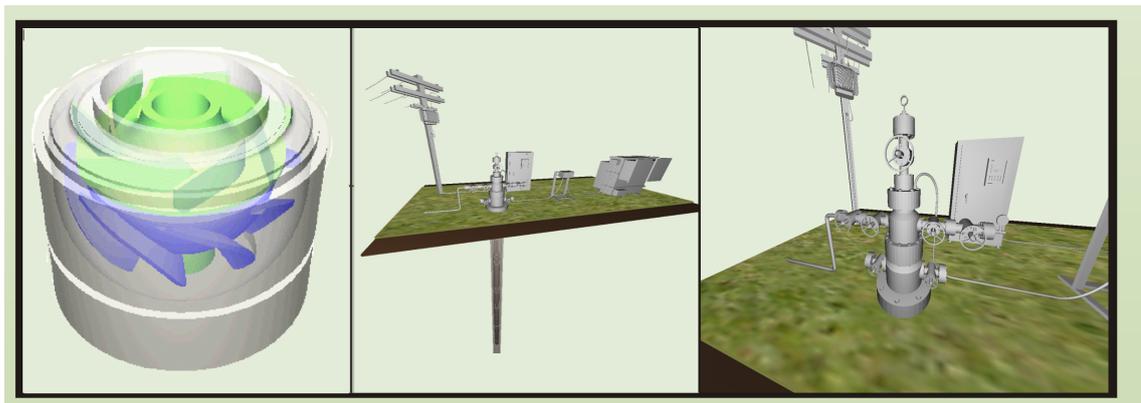


Figura 3.5: Modelos carregados pela Glscene e renderizado com OpenGL.

3.3.2 Otimização de Modelos 3D

Projetos com mundos 3D sempre envolvem limitações de hardware, software. É comum dá-se tanta ênfase a máxima qualidade da cena que o projeto finda sendo algo tão "pesado" que poucas máquinas conseguem executar, por isso deve sempre existir um equilíbrio entre a qualidade e a performance, de maneira tal que se possa garantir um bom resultado e uma boa visibilidade.

Dadas as limitações de hardware, um dos requisitos fundamentais para a criação de modelos 3D que vão ser utilizados em ambientes gerados em tempo real é, sem dúvida, a otimização poligonal.

Existem diferentes ferramentas para a otimização de modelos poligonais, algumas já vem inclusas por padrão em ambientes CAD, outras são projetadas de um modo independente.

Ferramentas dedicadas a criação de modelos para renderização em tempo real como o Autodesk® Maya®, mais recentemente, começaram adicionar em sua caixa de ferramentas algumas ferramentas de otimização poligonal.

Buscando-se otimizar os modelos construídos para as animações do simulador computacional, fez-se um estudo de algumas ferramentas comerciais e livres de otimização de modelos 3D. Das ferramentas testadas duas mostraram resultados interessantes, uma delas é o VizUp Optimize®, a outra o Polygon Cruncher®.

Polygon Cruncher® é um plugin para o 3D Photo Browser®, o 3DS Max®, e Lightwave® / Inspire 3D®. Esse software reduz o número de polígonos de objetos 3D sem mudar muito sua aparência. O software mantém todos os detalhes em alta resolução e otimização, mantendo também as informações de textura, vértice e tangente. Polygon Cruncher trabalha com algoritmos excepcionais para reduzir o seu modelo 3D em até 75%, sem a perda de detalhes. Sua interface é intuitiva e o seu modo de trabalhar é muito rápido, suportando vários modos de visualização.

VizUp Optimize®, da VizUp Inc., é um programa especialmente criado para importar e otimizar modelos 3D. Organizado e de fácil manejo, o ambiente principal do programa se limita a visualizar as opções e funções necessárias para a otimização dos modelos em maior ou menor escala. Inclui doze níveis de otimização, visor para trocar entre o modelo original e o otimizado e opções para ativar/desativar a visualização de vértices, fachadas, planos, suavizado e/ou texturas.

A Figura 3.6 mostra os resultados de otimização poligonal obtida pelo VizUp Optimize®. Para a maioria dos modelos otimizados o software alcançou uma taxa próxima a 50% de redução sem grandes perdas de qualidade. As perdas na qualidade só tornavam-se notáveis para reduções a taxas superiores a 70%. O Polygon Cruncher obteve desempenho muito próximo ao do VizUp Optimize, e ao contrário deste último sua versão de avaliação permite que sejam salvos os modelos otimizados.

3.4 A Interface Gráfica do Simulador

Um dos fatores mais importantes para desenvolver com sucesso uma aplicação é o projeto de interface com o usuário. A facilidade de uso é considerada pela engenharia de software um dos atributos essenciais dos produtos de software

No desenvolvimento de um projeto de software que envolve a interação homem-computador, o projeto de interface com o usuário é fundamental para o sucesso do sistema [Sommerville 2003]. Segundo [Pressman 2006], a interface com o usuário pode ser considerada o elemento mais importante de um sistema ou produto baseado em computador.

Uma interface bem projetada deve ser a mais amigável possível, possibilitando o usuário a extrair todo o poder computacional de uma aplicação e utilizá-la de forma confortável proporcionando uma interação homem-computador transparente. Entretanto, uma inter-

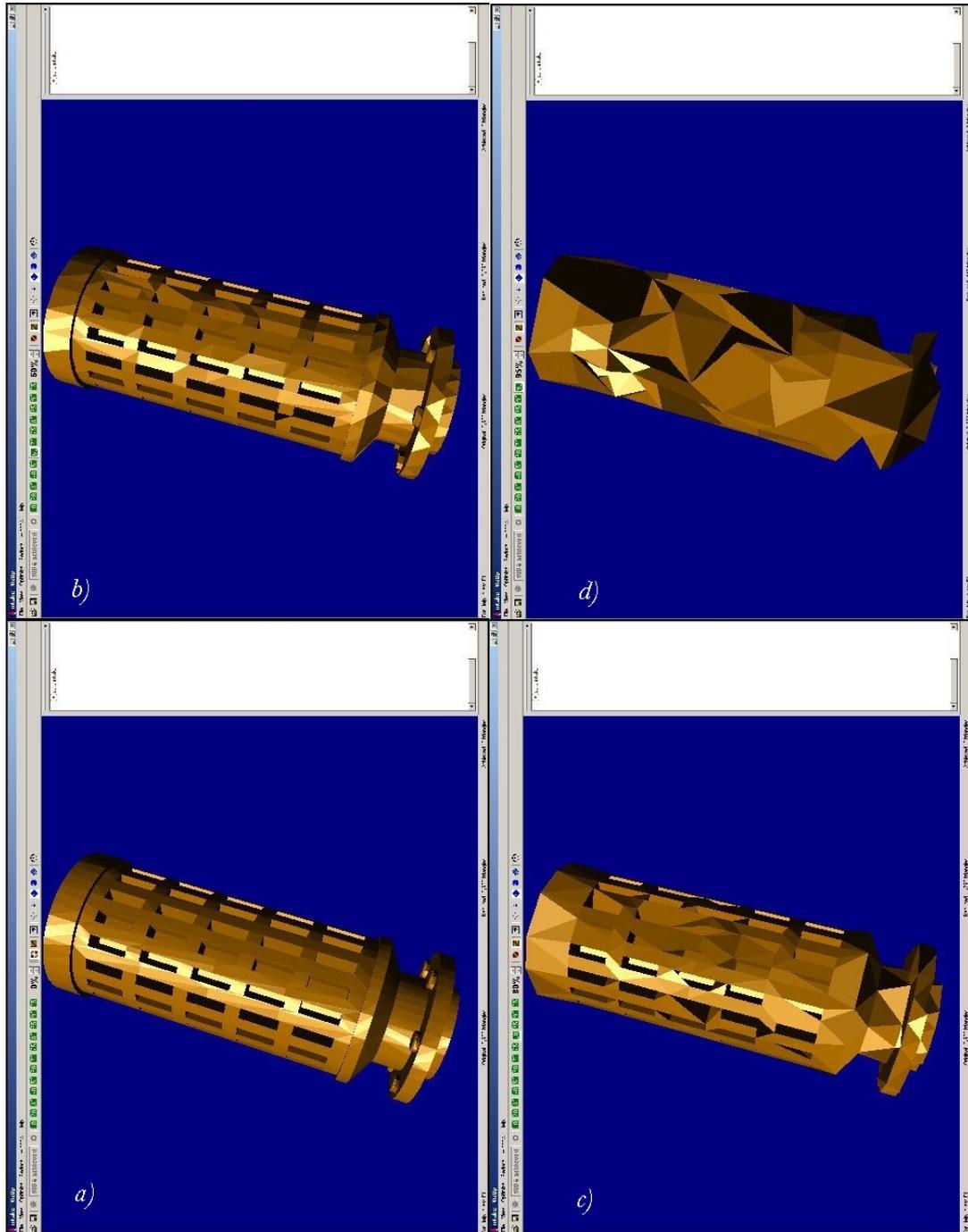


Figura 3.6: Otimização de modelo com VizUp Optimize: a) modelo original, modelo com b) 60%, c) 80% e d) 95% de redução

face mal projetada, pode se transformar em um ponto decisivo na rejeição de um sistema, independentemente de sua funcionalidade, podendo provocar ainda, a falha de uma aplicação que tenha sido bem projetada e desenvolvida [Souza & Costa 2006].

Em qualquer interface, gráfica ou não, o utilizador valoriza a facilidade com que consegue executar as tarefas pretendidas e a comodidade ao executá-las. Por isso, as questões chave na implementação de uma interface são a sua finalidade e os seus utilizadores finais. As escolhas e decisões a tomar ao longo do processo de criação da interface devem ser feitas com base na compreensão dos utilizadores. Isso envolve ter em consideração aquilo em que as pessoas são *boas* ou *menos boas*, considerar o que poderá ajudar os utilizadores no modo como executam as suas tarefas, meditar no que poderá dar qualidade às utilizações, ouvir o que as pessoas querem e envolvê-las no processo de criação, e adotar técnicas testadas pelos utilizadores ao longo de todo o desenvolvimento da interface.

A interface gráfica do simulador, correspondente a camada View da arquitetura MVC, foi construída com o uso dos componentes da plataforma Builder C++. A tela principal é mostrada na Figura 3.7, nela podemos perceber a barra de ferramentas principal e carregado o frame de simulação, o qual mostra as animações tridimensionais e, quando a simulação estiver em execução, os resultados da simulação.

Buscando-se evitar o acúmulo de janelas sobrepostas durante a entrada de dados e execução das simulações, deu-se preferência pelo uso de frames onde os mesmos são sempre carregados no corpo da tela principal mantendo sempre o mesmo padrão de iteração com o usuário.

O frame de simulação é dividido em duas partes. Em uma delas pode-se visualizar as animações tridimensionais tanto do sistema completo como dos subsistemas em particular (motor, bomba, selo protetor, separador centrífugo) bastando para isso acionar o devido botão localizado na parte superior da área da animação. Na outra metade do frame de simulação pode-se visualizar os dados de saída da simulação, novamente com a possibilidade de se selecionar entre dados de motor, bomba, e etc. através dos botões localizados na parte superior da mesma área.

Para se configurar o sistema, realizar a entrada de dados da simulação, basta acionar o botão de configuração na barra de ferramentas principal, que carregará o frame de configuração no corpo da tela principal.

Buscando-se organizar e facilitar a entrada de dados, os mesmos foram reagrupados em frames separados seguindo o seguinte esquema:

- Propriedades dos fluidos: Grau API, RGO, gravidades específicas dos fluidos e etc;

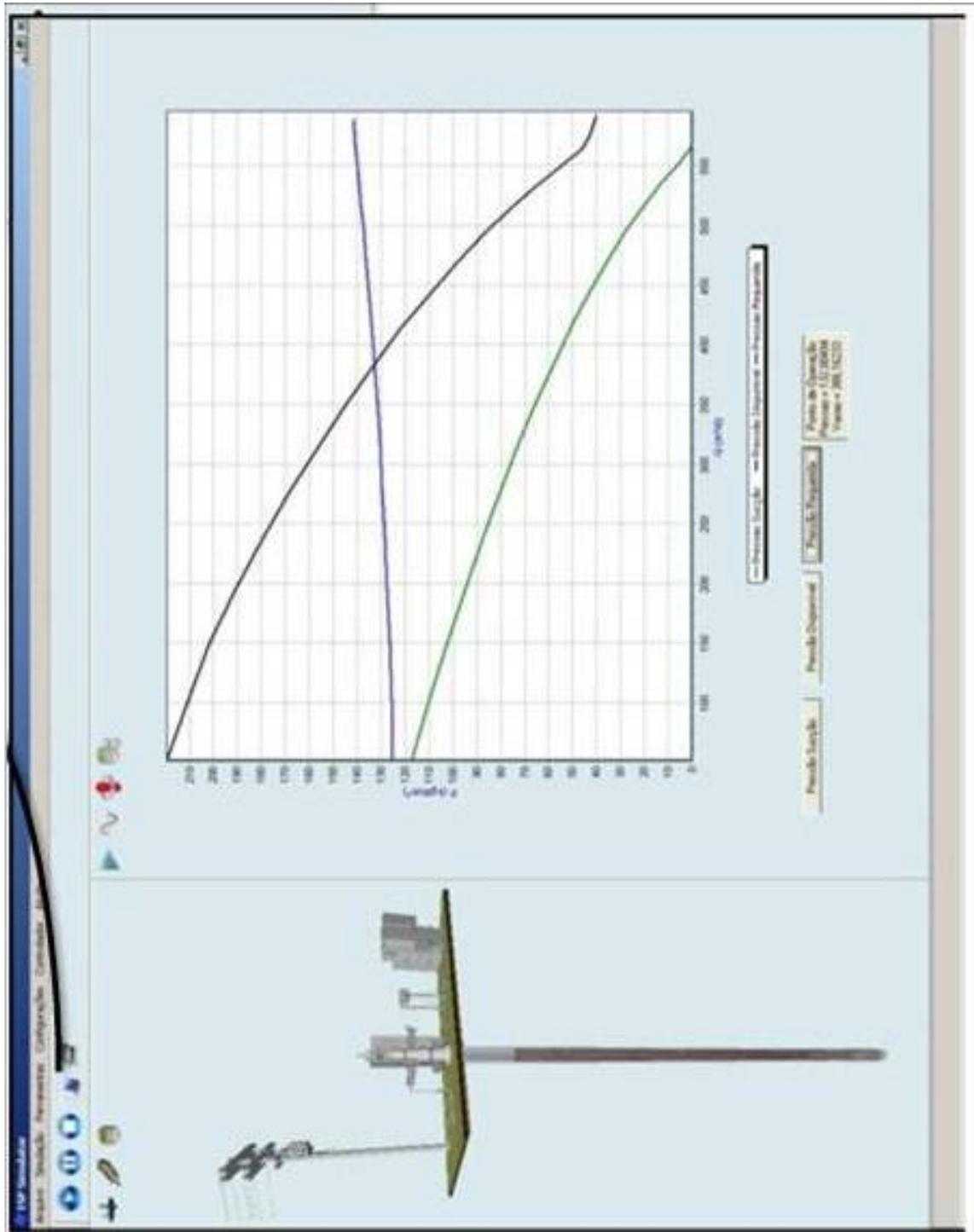


Figura 3.7: Tela principal do simulador BCS

- Esquema mecânico: profundidade da bomba, profundidade do canhoneado, ângulo do poço, rugosidade da tubulação, diâmetro da coluna de produção e do revestimento;
- Reservatório : pressão estática, pressão e vazão de teste;
- Correlações: opções de escolha de correlações multifásicas;
- Configuração da bomba: opções de escolha de fabricante e modelos de bombas centrífugas.

A Figura 3.8, mostra a opção de configuração da bomba onde se pode escolher entre alguns modelos presentes no banco de dados da aplicação. Escolhido um modelo, pode-se visualizar sua respectiva curva e/ou selecionar esse modelo para a simulação.

A Figura 3.9, mostra a opção de configuração do reservatório onde se pode definir as características do reservatório e visualizar a IPR do reservatório definido.

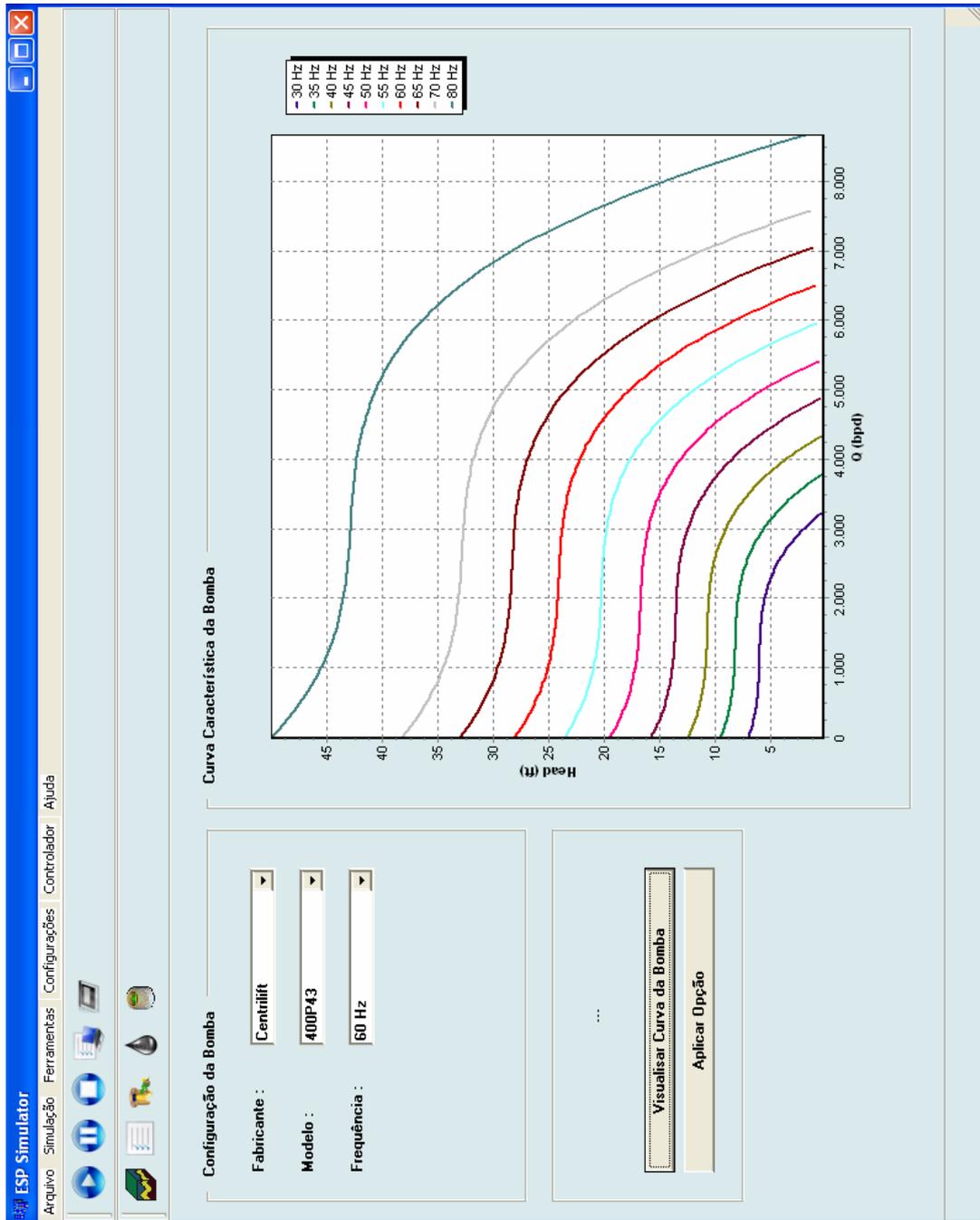


Figura 3.8: Tela de Configuração da Bomba

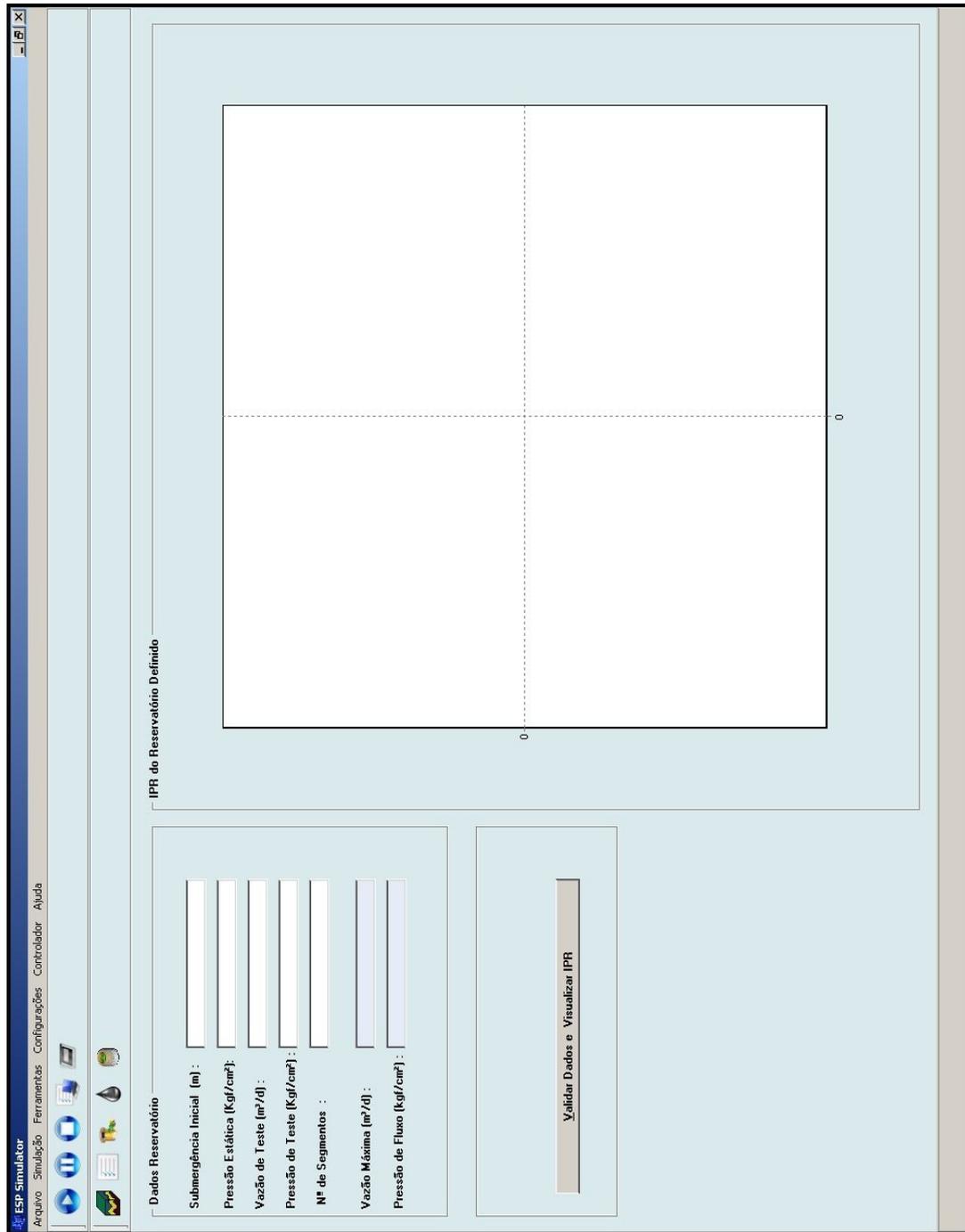


Figura 3.9: Tela de configuração do Reservatório

Capítulo 4

Conclusões

Neste trabalho é apresentado o desenvolvimento da interface gráfica de um simulador para o método de elevação de petróleo bombeio centrífugo submerso, cujo objetivo é apresentar, de forma clara e simples, os fenômenos de um sistema de BCS, viabilizando uma melhor compreensão dos fenômenos e facilitando o estudo do método.

O simulador tem a capacidade de representar a dinâmica de um sistema e a possibilidade de mudanças nos parâmetros em tempo de simulação podendo, assim, garantir a realização de diversos testes, que lembram ao usuário situações vivenciadas em campo. Esta característica torna o simulador uma ferramenta capaz de prover soluções na análise e projeto de sistemas de BCS.

Com o objetivo de acelerar a construção do software simulador, buscou-se desenvolver as duas partes principais do software, o núcleo simulador e a interface gráfica com usuário, paralelamente, como isso poderia levar a alguns problemas de desenvolvimento e manutenção, foram adotados alguns padrões de projetos e arquitetura de software.

A adoção desses padrões ajudou a amenizar os riscos citados, fornecendo-nos também estratégias para resolver outros problemas.

Para criação das animações tridimensionais, surgiu a necessidade de se utilizar um construtor de cenas 3D e um "Loader" para as mesmas, foi adotado, então, o SolidWorks para a construção das cenas e a Glscene para o carregamento e renderização das mesmas em tempo de execução. A GLScene mostrou-se uma boa ferramenta para construção de animações 3D.

Referências Bibliográficas

- Abreu, Adriana Meireles M. & João José de Assis Rangel (1999), 'Simulação computacional: Uma abordagem introdutória', *UENF - Vértices* **2**(1).
- Allis, D. H. & W. M. Capps (1984), 'Submersible pumping - long beach unit of east willmington field: A 17-year review', *JPT - Journal of Petroleum Technology* pp. 1321–1325.
- Almeida, Leonardo J. (2006), Sistema de representação e avaliação de metodos de elevação artificial de petróleo, Dissertação de mestrado, Universidade Federal da Bahia, Salvador, BA.
- Batista, Evelyne S. (2008), Desenvolvimento de uma ferramenta computacional para aplicação no método de elevação por bombeio centrífugo submerso, Dissertação de mestrado, Universidade Federal do Rio Grande do Norte, Natal, RN.
- Batista, Evelyne S., Rutácio O. Costa, André Laurindo Maitelli & Tiago S. Barbosa (2008), 'Simulação computacional para poços de petróleo com método de elevação artificial por bombeio centrífugo submerso', *Rio Oil & Gas Expo and Conference* .
- Bezerra, Murilo Valença (2002), Avaliação de métodos de elevação artificial de petróleo utilizando conjuntos nebulosos, Dissertação de mestrado, Universidade Estadual de Campinas, Campinas, SP.
- Clegg, J. D. (1988), 'High-rate artificial lift', *JPT - Journal of Petroleum Technology* pp. 277–282.
- Kunkel, Bill (2000), 'Downhole pumps deliver broad gains', *Hart's E&P* pp. 71–80.
- L.Wilson, B. (1994), 'Esp gas separator's affect on run life', *SPE Progam Commitee* **1**(SPE 28526).
- Nascimento, João Maria A. (2005), Simulador computacional para poços de petróleo com método de elevação artificial por bombeio mecânico, Dissertação de mestrado, Universidade Federal do Rio Grande do Norte, Natal, RN.

- Pinheiro, Paulo G., Adriano A. N. de Melo, José Eurico de V. Filho & Vasco Furtado (2008), 'A representação do tempo em simulações computacionais', *Universidade de Fortaleza - UNIFOR*.
- Pressman, Roger (2006), *Engenharia de Software*, 6ª edição, McGraw-Hill, São Paulo, SP.
- Sommerville, Ian (2003), *Engenharia de Software*, 6ª edição, Addison-Wesley, São Paulo, SP.
- Souza, Luciano S. & Ivanir Costa (2006), 'Fatores impactantes nos projetos de interface com o usuário de software de dispositivos móveis', *XXVI ENEGEP*.
- Thomas, José Eduardo (2004), *Fundamentos de Engenharia de Petróleo*, 2ª edição, Interciência, Rio de Janeiro, RJ.
- Wright, Richard S., Benjamin Lipchak & Nicholas Haemel (2005), *OpenGL Super Bible*, 4ª edição, Pearson Education, Inc., Boston, MA.

ANEXO
PUBLICAÇÕES

Certificada

Rio Oil & Gas Expo and Conference 2008

Certifica que,

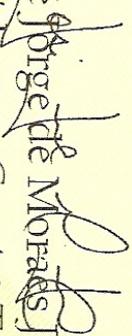
Evellayne da Silva Batista, Tiago de Sousa Barbosa, André Laurindo Maitelli,

Rutácio de Oliveira Costa

Contribuíram com a apresentação do trabalho

“IBP2654_08 - Simulação computacional para poços de petróleo com método de elevação artificial por bombeio centrífugo submerso”

Rio de Janeiro, 18 de Setembro de 2008.



José Jorge de Moraes Júnior
Presidente – Comitê Técnico



IBP2654_08

SIMULAÇÃO COMPUTACIONAL PARA POÇOS DE PETRÓLEO COM MÉTODO DE ELEVAÇÃO ARTIFICIAL POR BOMBEIO CENTRÍFUGO SUBMERSO

Evellyne da Silva Batista¹, Rutácio de Oliveira Costa², André Laurindo Maitelli³, Tiago de Souza Barbosa⁴

Copyright 2008, Instituto Brasileiro de Petróleo, Gás e Biocombustíveis - IBP

Este Trabalho Técnico foi preparado para apresentação na *Rio Oil & Gas Expo and Conference 2008*, realizada no período de 15 a 18 de setembro de 2008, no Rio de Janeiro. Este Trabalho Técnico foi selecionado para apresentação pelo Comitê Técnico do evento, seguindo as informações contidas na sinopse submetida pelo(s) autor(es). O conteúdo do Trabalho Técnico, como apresentado, não foi revisado pelo IBP. Os organizadores não irão traduzir ou corrigir os textos recebidos. O material conforme, apresentado, não necessariamente reflete as opiniões do Instituto Brasileiro de Petróleo, Gás e Biocombustíveis, seus Associados e Representantes. É de conhecimento e aprovação do(s) autor(es) que este Trabalho Técnico seja publicado nos Anais da *Rio Oil & Gas Expo and Conference 2008*.

Resumo

O bombeio centrífugo submerso (BCS) tem se mostrado uma das soluções mais adequadas como método de elevação. Tanto em aplicações *onshore* como *offshore*, em condições adversas de temperatura, fluidos viscosos e ambientes gaseificados. As dificuldades encontradas na manutenção e instalação de equipamentos são fatores que oneram a produção de petróleo em águas profundas, com isso, a otimização da produção, via automação, pode ser um passo fundamental para a redução de custos e falhas nos equipamentos de subsuperfície.

Este trabalho visa apresentar o desenvolvimento de um simulador computacional para o conjunto BCS. Este simulador será capaz de representar o comportamento dinâmico deste método, considerando o modelo da fonte e da transmissão de energia elétrica para o motor, o modelo do motor elétrico BCS (incluindo o cálculo térmico), simulações do escoamento e também disponibilizará animações tridimensionais de cada subsistema BCS (motor, bomba, selo, entre outros), visando facilitar a visualização e compreensão do processo.

O simulador propõe uma melhoria no monitoramento de poços petrolíferos através de um melhor conhecimento dos fenômenos físicos pertinentes ao processo, podendo prevenir falhas, facilitando tomadas de decisão numa menor parcela de tempo. Assim também, permitindo controlar o aumento da produção de óleo, através de ajustes de condições operacionais (frequência, pressão na cabeça do poço, pontos de operações, etc).

Abstract

The electrical submersible pump (ESP) is an appropriate solution to lifting methods. This approach is indicated to onshore and offshore applications with adverse temperature conditions, viscous fluid and gasified environments. Installation and maintenance problems in equipments raise the cost of oil production in deep water, thus, the production optimization through automation can be essential to minimize cost and failure in the subsurface equipment.

This paper shows the computational simulator development for the ESP set. This simulator will be able to represent the dynamic behavior of this method, considering the model of the source and electric energy transmission for the engine, the electric engine model ESP (including the thermal calculation), flow simulations and three-dimensional animations for each ESP subsystem (engine, pump, seal, and so on), to facilitate the visualization and understanding of the process.

The simulator proposes an improvement in oil wells monitoring through a better knowledge of relevant physical phenomena of the process, it can prevent failures, facilitating decision-making in a smaller portion of time. Similarly, allowing to control the increased oil production through adjustments of operating conditions (frequency, head well pressure, operations points, etc.).

¹ Engenheira da Computação – Universidade Federal do Rio Grande do Norte

² Mestre, Engenheiro de Petróleo – PETROBRAS UN-RNCE

³ Doutor, Engenheiro Eletricista – Universidade Federal do Rio Grande do Norte

⁴ Aluno de IC, Engenharia da Computação – Universidade Federal do Rio Grande do Norte

1. Introdução

Segundo Rangel (1999), a técnica de simulação computacional consiste em estabelecer um modelo capaz de descrever ou representar o problema real a ser submetido à manipulação “experimental” em um computador. Em outras palavras, simulação computacional consiste em conduzir “experimentos” em um computador, envolvendo relações de conteúdo lógico e matemático, necessários à descrição do comportamento e da estrutura de um problema real, em períodos de tempo bem definidos.

Através de simuladores é possível a visualização das principais etapas de um processo. Assim, a análise de sistemas em ambientes computacionalmente simulados pode cooperar na melhoria de processos e no emprego de novas técnicas, sem possíveis desgastes do sistema real. Neste contexto, está sendo desenvolvido um simulador para poços de petróleo com o método de elevação artificial BCS, de forma que, seja possível analisar e melhor compreender o processo real.

Neste tipo de bombeio, ilustrado na figura 1, a energia é transmitida para o fundo do poço através de um cabo elétrico que alimenta um motor de subsuperfície, o qual está diretamente conectado a bomba centrífuga. A bomba, por sua vez, transmite essa energia para o fluido sob a forma de pressão, elevando-o para a superfície.

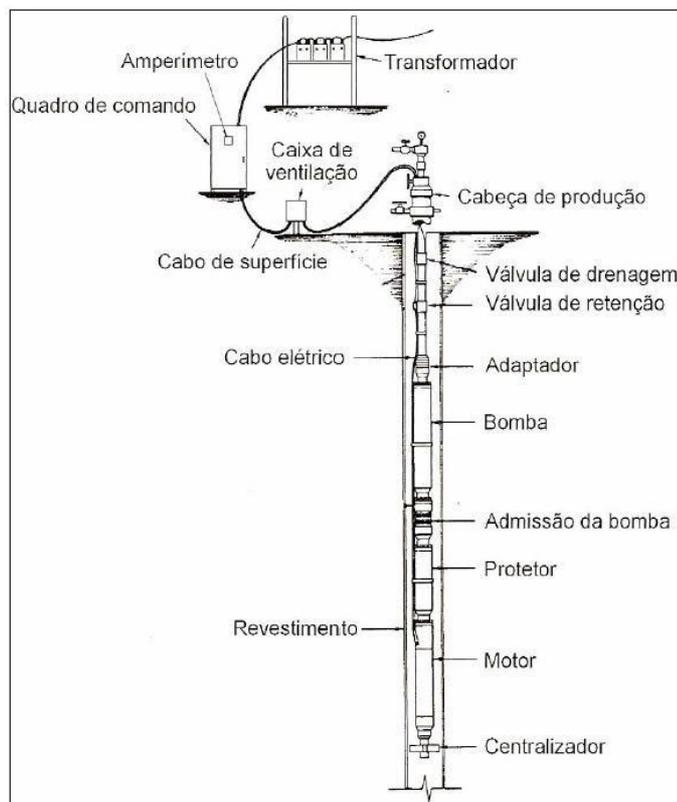


Figura 1. Poço produtor por bombeio centrífugo submerso.

O transformador fornece a tensão necessária para que o motor opere. A queda de tensão e as perdas elétricas no cabo serão consideradas através de um modelo simplificado, como apresentado por French (1991).

2. Modelo do motor elétrico BCS

Os motores elétricos utilizados para impulsionar as bombas centrífugas são normalmente de indução, trifásicos, com dois pólos. Neste trabalho, considera-se o motor acionado por variador de frequência (*VFD*). De acordo com Laughy (2007), o circuito equivalente pode ser representado pela figura 2.

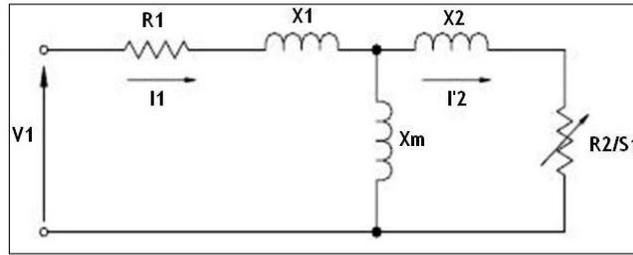


Figura 2. Circuito equivalente do motor de indução.

Onde:

V_1 : tensão por fase no terminal do motor com frequência de 50/60 Hz;

R_1 , X_1 : Resistência e indutância do estator;

R_2 , X_2 : Resistência e indutância do rotor;

X_m : Indutância magnética;

I_1 : Corrente do estator;

I'_2 : Corrente do rotor referindo-se ao lado do estator;

S_1 : Escorregamento.

O VFD permite que o motor opere em um nível de torque constante até a frequência nominal, ou seja, a tensão e a frequência variam correspondentemente conforme a equação 1. Após a frequência nominal ele opera com potência constante.

$$\frac{\text{voltage}}{\text{frequencia}} = \frac{\text{Volts}}{\text{Hz}} = \text{constante} \quad (1)$$

As resistências R_1 e R_2 são valores constantes e independentes da frequência. Entretanto, os valores das indutâncias X_1 , X_2 e X_m são funções da frequência e podem ser calculadas da seguinte maneira:

$$X_1 = 2\pi f_1 L_1 \quad (2)$$

$$X_2 = 2\pi f_1 L_2 \quad (3)$$

$$X_3 = 2\pi f_1 L_m \quad (4)$$

Onde L_1 , L_2 e L_m são as indutâncias de dispersão do estator, do rotor referindo-se ao estator e indutância magnética, respectivamente. Assim, as equações de tensão e corrente de acionamento do motor podem ser calculadas da seguinte maneira:

$$V_{st-VFD} = \frac{V_1}{f_{sys}} \times f_1 \quad (5)$$

$$I_{st-VFD} = \frac{\left(\frac{V_1}{f_{sys}} \times f_1 \right)}{\left(R_1 + j2\pi f_1 L_1 \right) + \frac{\left(R_2 + j2\pi f_1 L_2 \right) \left(j2\pi f_1 L_m \right)}{R_2 + j2\pi f_1 \left(L_2 + L_m \right)}} \quad (6)$$

A equação do torque de partida é em função da tensão (equação 7) ou da corrente de acionamento (equação 8) do motor, de modo que, pode ser calculado como segue abaixo:

$$T_{st} = \frac{3pR_2}{4\pi f_1} \left| \frac{jX_m}{(R_1R_2 - X_1X_2 - X_m(X_1 + X_2)) + j(R_1X_2 + R_2X_1 + X_m(R_1 + R_2))} \right|^2 \times \left(\frac{V_1}{f_{sys}} \times f_1 \right)^2 \quad (7)$$

$$T_{st} = \frac{3pR_2}{4\pi f_1} \left| \frac{jX_m}{R_2 + j(X_2 + X_m)} \right|^2 \times I_{st-VFD}^2 = \frac{3pR_2}{4\pi f_1} \left| \frac{j2\pi f_1 L_m}{R_2 + j2\pi f_1 (L_2 + L_m)} \right|^2 \times I_{st-VFD}^2 \quad (8)$$

2.1. Cálculo térmico

Esta etapa visa calcular o efeito térmico sob o motor, para que este simulador seja capaz de identificar a temperatura de funcionamento, com isso pode-se prevenir uma possível queima do motor. Uma vez elevada à temperatura além dos limites permitidos, pode-se provocar a inutilização do componente. Para isso, foi estudado o comportamento de uma convecção forçada no interior de tubos.

2.1.1. Convecção forçada no interior de tubos e dutos

Para calcular a temperatura do motor utilizou-se a abordagem de Kreith (1997). Para isso foi necessário identificar o calor transmitido através da convecção exigindo o conhecimento do coeficiente de transmissão de calor entre a parede do conduto e o fluido que escoar em seu interior. Uma vez conhecido o coeficiente de transmissão de calor para uma dada forma geométrica e as condições especificadas de escoamento, diferença de temperatura, pode ser calculado o calor transmitido por unidade de tempo, através da equação abaixo:

$$q_c = \overline{h}_c A (T_{Superfície} - T_{Fluido}) \quad (9)$$

Onde: q_c : calor transmitido por unidade de tempo por convecção (Kcal/h)

\overline{h}_c : coeficiente médio de transmissão de calor por convecção (Kcal/h m² °C)

A : área de transmissão de calor (m²)

O \overline{h}_c , na realidade, é uma função complexa do escoamento do fluido, das propriedades térmicas do meio fluido e da geometria do sistema. Seu valor numérico não é, em geral, uniforme sobre a superfície e depende também do local onde a temperatura do fluido é medida. Este coeficiente pode ser calculado a partir do número de *Nusselt* (\overline{N}_u), que é uma quantidade adimensional que, na prática, é uma medida conveniente do \overline{h}_c , porque uma vez conhecido seu valor, o coeficiente de transmissão de calor por convecção pode ser calculado pela relação:

$$\overline{N}_u = \frac{\overline{h}_c D_h}{k} \quad (10)$$

Onde k é a condutibilidade térmica do fluido e para o caso de escoamento em tubos o comprimento significativo no número de *Nusselt* é o diâmetro hidráulico (D_h). A partir de uma análise dimensional, o número de *Nusselt* obteve uma relação funcional entre *Reynolds* (R_e) e *Prandtl* (P_r). O número de *Prandtl* é uma combinação de propriedades físicas que relaciona a distribuição de temperatura com a distribuição de velocidade, este é definido na equação seguinte:

$$P_r = \frac{c_p \mu}{k} \quad (11)$$

Tal que: c_p : calor específico (J/Kg°C)

μ : viscosidade absoluta (Kg/ms)

O número de *Reynolds* provoca efeitos na transmissão de calor e na perda de carga no monofásico, pois para um dado fluido, o número de *Nusselt* depende das condições do escoamento, que podem ser caracterizadas por R_e (definido na equação 12).

$$R_e = \frac{\rho v D_h}{\mu} \quad (12)$$

Onde: v : velocidade de escoamento do fluido (m/s)

ρ : densidade (kg/m³)

Para realizar o cálculo térmico para o bombeio centrífugo submerso, fez-se um estudo sobre o escoamento turbulento. Para isso, foi necessário fazer uma analogia entre a transmissão de calor e a quantidade de movimento. É neste contexto que o número de *Stanton* (S_t) é observado.

$$S_t = \frac{N_u}{R_e P_r} = 0.023 R_e^{-0.2} \quad (13)$$

A partir da análise, constatou-se que se P_r for admitido unitário, pode se utilizar as seguintes equações:

$$N_u = 0.023 R_e^{0.8} \text{ e } \overline{h_c} = 0.023 V^{0.8} D^{-0.2} k \left(\frac{\mu}{\rho} \right)^{-0.8} \quad (14)$$

Para fluidos que têm P_r no intervalo de 0,5 a 100 recomenda-se, com base em dados experimentais, que S_t seja multiplicado por $P_r^{2/3}$, ou seja,

$$S_t P_r^{2/3} = j = 0.023 R_e^{-0.2} = \frac{f}{8} \quad (15)$$

O termo $S_t P_r^{2/3}$ é usualmente chamado de fator j de *calburn*. Assim, para efetuar o cálculo do coeficiente de transmissão de calor, obedecemos à seguinte equação, onde $G = \rho v$:

$$\frac{\overline{h_c}}{C_p G} = 0.023 \left(\frac{\mu_f}{D_H G} \right)^{0.2} P_r^{-2/3} \quad (16)$$

Ao encontrar o coeficiente de transmissão de calor ($\overline{h_c}$), substitui na equação 9. Encontra-se o calor transmitido (q_c). Considerando para o regime permanente a temperatura do fluido constante, faz-se o equilíbrio entre $q_g = q_c$, tal que, pode ser representado pela equação 17:

$$iR^2 = \overline{h_c} A (T_{Motor} - T_{Fluido}) \quad (17)$$

Dessa forma, encontra-se a temperatura do motor.

3. Simulação do escoamento

Na simulação do escoamento, o simulador faz uso de uma biblioteca de ligação dinâmica (*dll*), com o objetivo de calcular os valores de pressão, temperatura, regime de fluxo e propriedades dos fluidos em tubulações verticais, horizontais ou inclinadas e o ponto de operação, que determina a carga e a vazão da bomba para um dado sistema. Uma ilustração deste processo pode ser representada na figura 3. Onde P_{wf} representa a pressão do fundo em fluxo, P_{cab} a pressão na cabeça do poço, $P_{sucção}$ representa a pressão de sucção da bomba e $P_{disponível}$ define pressão disponível na descarga da bomba.

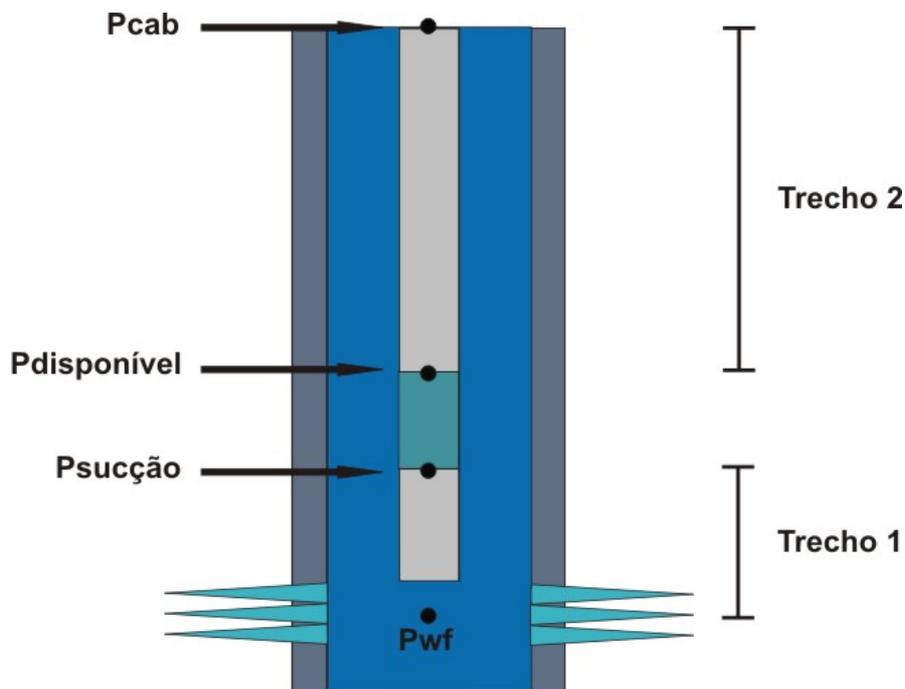


Figura 3. Modelo do processo de escoamento.

Inicialmente em regime permanente, calculou-se a curva de *IPR* (*Inflow Performance Relationship*), de acordo com Lee (1988), na profundidade do canhoneado, utilizando o modelo de *Vogel* descrito na equação abaixo:

$$\frac{Q}{Q_{max}} = 1 - 0,2 \left(\frac{P}{P_e} \right) - 0,8 \left(\frac{P}{P_e} \right)^2 \quad (18)$$

Onde Q , P e P_e são respectivamente vazão, pressão de fluxo no fundo do poço e pressão estática, respectivamente. Através de correlações multifásicas apresentadas por Shoham (2006), por exemplo, *Beggs-Brill*, *Orkiszewski*, *Hagedorn-Brown*, *Duns & Ros*, entre outras, calculou-se a curva de pressão disponível na sucção da

bomba versus vazão no trecho 1 da figura 3. Assim como, o comportamento do reservatório, tipos de regime, pressões e temperaturas em vários pontos do trecho 1. Essas correlações dependem de variáveis do sistema. Desta forma, dividimos o sistema de entrada nos seguintes grupos de parâmetros: propriedades dos fluidos, dados de coluna de produção e dados gerais do reservatório.

Após encontrar a curva de sucção, encontrou-se a curva disponível na descarga da bomba e a curva requerida pelo sistema. Para encontrar a curva disponível, determinou-se o ΔP entre a sucção e a descarga através da curva característica da bomba fornecida pelo fabricante. Assim, a curva disponível será a da pressão de sucção somada ao ΔP . Para determinar a curva requerida pelo sistema calcula-se a pressão requerida para o escoamento multifásico no trecho 2 da figura 3, de maneira que, o ponto de operação ilustrado na figura 4 é o ponto de interseção entre a curva disponível na descarga da bomba e a curva requerida pelo sistema.

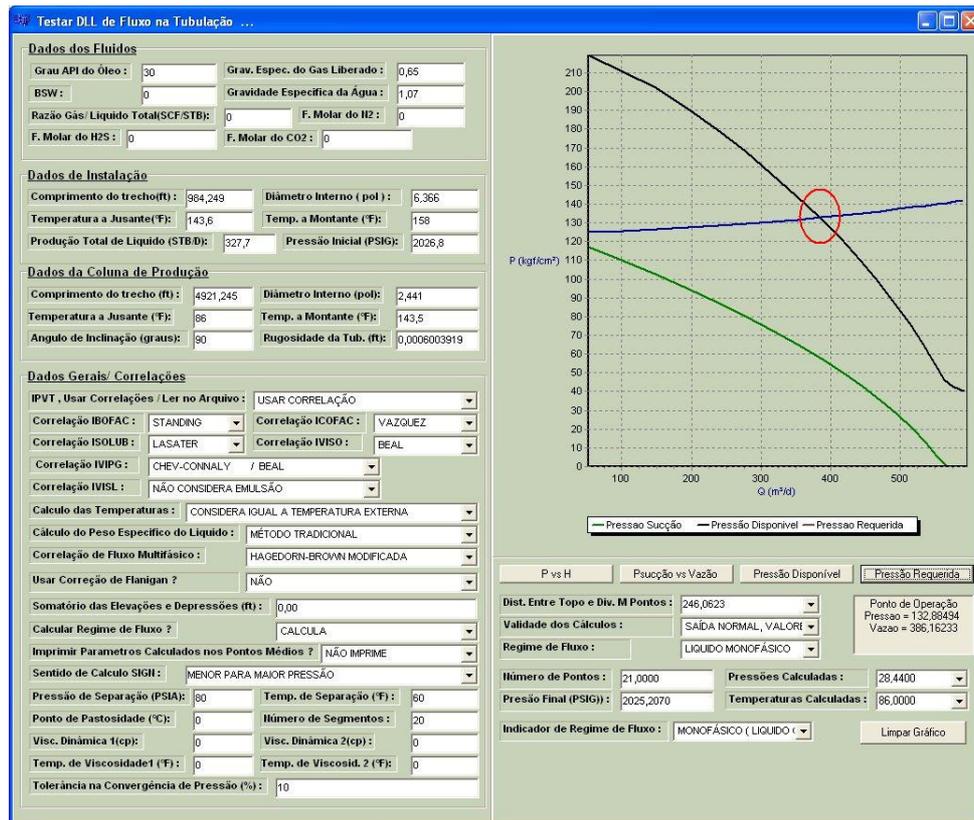


Figura 4. Tela da simulação do escoamento – Ponto de operação.

Deve ser levado em consideração para a simulação do escoamento o comportamento da bomba quando bombeando fluidos gaseificados e fluidos viscosos, pois irá atingir a velocidade da bomba. Estas mudanças são previsíveis através das leis de afinidade.

4. Animações tridimensionais

A interface gráfica de interação com o usuário está sendo desenvolvida com base na biblioteca de componentes do Builder C++. O interesse deste simulador é tornar a interface bastante interativa com o usuário, para que o mesmo possa visualizar tanto o processo como um todo, como cada componente do conjunto BCS, podendo compreender e visualizar o sistema de diversos ângulos. Na figura 5, pode-se observar uma visão externa de alguns componentes do simulador.

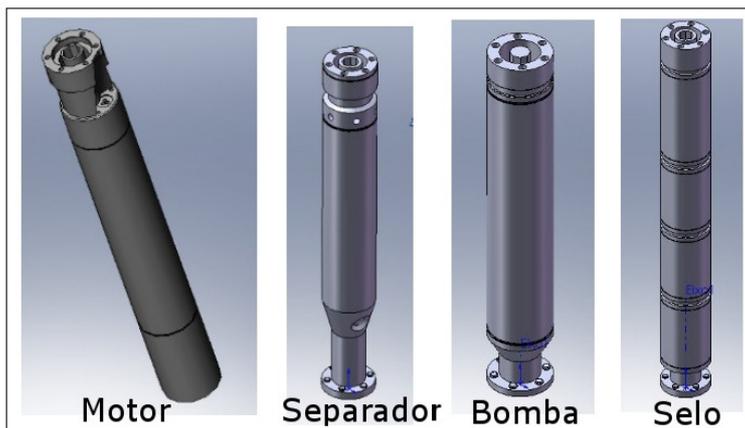


Figura 5. Visão externa de alguns componentes do simulador.

5. Conclusões

O referido trabalho procurou analisar as características básicas relacionadas à implementação de um simulador computacional para o método de elevação artificial por bombeio centrífugo submerso. Foram descritos o modelo matemático do motor de indução trifásico (incluindo seus cálculos térmicos), simulações do escoamento no poço baseado em correlações empíricas para cálculo de perda de carga em sistemas multifásicos e análise tridimensional dos principais componentes do conjunto BCS. Uma vantagem do simulador em desenvolvimento quando comparado a outros simuladores BCS é a utilização de uma interface gráfica de fácil interação com o usuário.

Para soluções futuras será adicionado ao simulador em desenvolvimento novas funcionalidades relacionadas com o comportamento dinâmico deste método de elevação, parametrização e comunicação com controladores.

6. Referências

- RANGEL, J. José de Assis. *Simulação Computacional - Uma Abordagem Introductória*. Centro Federal de Educação Tecnológica de Campos, julho, 1999.
- LAUGHY, R., LIU, J., XIAODONG, L. "Induction Motor Starting for ESP System Applications", Society of Petroleum Engineers - Gulf Coast Section Electric Submersible Pump Workshop held in The Woodlands, Texas, 2007.
- KREITH, Frank. *Princípios da Transmissão de Calor*; 3ª edição (São Paulo: Edgard Blucher, 1997).
- LEE, H.K., Amoco production Co. *Computer Modeling and optimization for submersible pump lifted wells*. Society of Petroleum Engineers (SPE 17586), China, 1988.
- FRENCH, CAWLEY, GILLESPIE & Assocs. Inc. *Optimum Cable Selection for Electrical*. Society of Petroleum Engineers (SPE 21693), 1991.
- SHORAM, Ovadia. "Mechanistic Modeling of Gas-Liquid Two-Phase Flow in Pipes", University of Tulsa - Society of Petroleum Engineers, 2006.