



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS



Modelagem e Simulação do Agendamento e Despachamento de Manobras Espaciais por Tempo e por Eventos

RELATÓRIO FINAL DE PROJETO DE INICIAÇÃO CIENTÍFICA
(PIBIC/CNPq/INPE)

Anna Carolina Moreira (ETEP, Bolsista PIBIC/CNPq)
E-mail: annac_moreira@hotmail.com

Marcelo Lopes de Oliveira e Souza (DMC/ETE/INPE, Orientador)
E-mail: marcelo@dem.inpe.br

Julho de 2013

RESUMO

Este trabalho, iniciado em Janeiro de 2012, tem como objetivo estudar o agendamento e despachamento de manobras orbitais por tempo e eventos. Um satélite artificial, assim que é posto em órbita, precisa ser manobrado: inicialmente para atingir sua órbita final; e/ou periodicamente, para manter-se em sua órbita nominal, e assim concluir sua missão. As estações terrenas são as responsáveis por agendar (=planejar) essas manobras e enviá-las ao satélite para serem despachadas (=executadas). Para isto, as estações de rastreo no solo estabelecem o contato com o satélite quando ele passa sobre a região de visibilidade de suas antenas. Os sinais transmitidos pelo satélite são captados pela antena estabelecendo um enlace descendente de comunicação e telemetria, que revela o estado atual de funcionamento do veículo. Uma vez estabelecido o enlace descendente, a estação cria também um enlace ascendente, que é utilizado para o envio de telecomandos e a execução de medidas de rastreo (distância e velocidade). Baseadas nestas medidas, as manobras espaciais são calculadas e os telecomandos podem ser enviados para o planejamento e execução de tais manobras espaciais visando atingir a órbita final. Estes telecomandos serão executados com base nas medidas de tempo disponíveis a bordo; ou em sinais lógicos denotando a ocorrência de eventos internos ou externos ao satélite. Neste trabalho, a ferramenta utilizada é o Orbiter, software livre de simulação tridimensional de vôo, que permite a programação da modelagem e simulação de vôos espaciais. Para isto, o trabalho apresenta um estudo sobre a aplicação da linguagem de programação Lua utilizada no software, na transferência de Hohmann. E apresenta um algoritmo prévio que inclui chamadas para a propulsão, controle de atitude do veículo e o cálculo matemático da manobra. Posteriormente, o trabalho pretende apresentar a modelagem e simulação do agendamento e despachamento de uma sucessão de manobras espaciais mais realistas de satélites como os do INPE; e analisar os resultados obtidos. Este trabalho está em andamento; mas os resultados preliminares sugerem que os modelos e simulações usados nos permitem calibrar as manobras visando realizá-las com êxito e não perder o satélite. E até prever algumas das falhas que podem vir a ocorrer nas manobras e assim corrigi-las, sempre que possível.

Lista de Figuras

Figura 1: Relações geométricas do movimento elíptico.....	7
Figura 2: Elementos orbitais clássicos. Fonte: Aula 05 – Mecânica Celeste; Disponível em: http://www.fis.unb.br/plasmas/aula05.pdf	8
Figura 3: Transferência de Hohmann. Fonte: Santos (2005) [10].....	10
Figura 4: Transferência Bi-elíptica Tri-impulsiva. Fonte: Santos (2005) [10].....	12
Figura 5: Interface inicial do software Orbiter 2010.	14
Figura 6: Modo de cabine da espaçonave Atlantis em 3D.	15
Figura 7: Modo de cabine da espaçonave Atlantis em 2D.	16
Figura 8: Interface de Programação do software Orbiter 2010.	17
Figura 9: Console de Programação do software Orbiter executando a simulação.	18
Figura 10: Função "launch()" a ser executada.....	18
Figura 11: Estação de rastreamento de Cuiabá.	20
Figura 12: Biblioteca de métodos (comandos) do Orbiter.	22
Figura 13: Script em Lua para o controle de atitude da nave Atlantis.	23
Figura 14: Script de lançamento da nave Atlantis.....	23

Sumário

1. Introdução	5
1.1. Objetivo	5
1.2. Motivação	5
2. Histórico	6
3. Fundamentação Teórica	7
3.1. Introdução a Mecânica Orbital	7
3.2. Manobras Orbitais	9
3.2.1. Transferências Orbitais	10
4. O Software Orbiter	14
4.1. Interface de Programação	16
4.2. Vôo de demonstração	17
4.3. Limitações do Software Orbiter	19
5. Metodologia Aplicada	19
5.1. Estudo do Agendamento e Despachamento de Manobras Orbitais	19
5.2. Sequência para o Agendamento e Despachamento de Manobras Orbitais	21
5.3. Modelagem e simulação do agendamento e despachamento da transferência de Hohmann utilizando o console de programação Lua do ambiente Orbiter	21
Conclusão	24
Referências Bibliográficas	25

1. Introdução

1.1. Objetivo

O objetivo deste Trabalho de Iniciação Científica é estudar o problema da modelagem e simulação do agendamento e despacho de manobras espaciais por tempo e por eventos e mostrar a importância da modelagem e simulação antes da realização de manobras em satélites. Isso será feito através da modelagem e simulação do agendamento e despacho da correção dos movimentos de um satélite como os do INPE [1-4; 7-14]; e da discussão das sequências de simulação do agendamento e despacho nos ambientes de software Orbiter [5, 6] e, talvez, SimProcess/SimScript [15] e ProModel [16].

1.2. Motivação

Durante a Guerra Fria, a URSS e os EUA, as duas superpotências da época, iniciaram uma competição pela supremacia da tecnologia espacial, conhecida como Corrida Espacial. Tal competição nos proporcionou grandes avanços na exploração espacial. A União Soviética foi o primeiro país a colocar um satélite em órbita no espaço, o Sputnik, em 04/10/1957. A partir de então, vários outros satélites foram postos em órbita com diferentes funções: sensoriamento remoto, coleta de dados, previsão do tempo, comunicação e outros.

Os satélites em órbita necessitam ser controlados de forma que permaneçam em suas órbitas, ou sejam transferidos para outras órbitas, a fim de realizar suas missões. As estações terrenas são as responsáveis pelas atividades de controle em órbita, planejando e executando manobras.

A modelagem e a simulação de manobras orbitais são frequentemente utilizadas para a realização dessas atividades, pois permitem: 1) inferir o comportamento mais próximo do real do sistema analisado; 2) prever possíveis falhas que possam ocorrer nas missões; e 3) corrigi-las sempre que possível.

2. Histórico

Este Projeto de Iniciação Científica foi iniciado em 1º de Dezembro de 2011 sob a orientação do Dr. Marcelo Lopes de Oliveira e Souza. Durante o mês de Janeiro a Julho de 2012 foi realizado o estudo da apostila de Introdução a Dinâmica Orbital e outras semelhantes [1-4] e a familiarização das ferramentas do software Orbiter e sua linguagem de programação [5-6] e das demais referências bibliográficas sobre 1) manobras orbitais e agendamento e 2) despachamento de manobras orbitais [7-14].

Durante o período de Agosto de 2012 e Janeiro de 2013, foi continuado o estudo do software Orbiter, a realização das simulações de seu manual [5-6] e a familiarização do com a linguagem de programação Lua [15].

E entre Fevereiro e Junho de 2013, foi realizado o estudo da implementação da transferência de Hohmann no Orbiter, utilizando a linguagem Lua [15-16].

3. Fundamentação Teórica

3.1. Introdução a Mecânica Orbital

A Mecânica Orbital é o estudo do movimento de corpos celestes artificiais, como satélites artificiais e veículos espaciais sob a ação de forças gravitacionais, de arrasto atmosférico, pressão de radiação solar, forças eletromagnéticas, entre outras.

Com base nas observações de Tycho Brahe, Johannes Kepler reformou radicalmente o modelo copernicano ao identificar empiricamente suas três leis que descrevem o movimento dos planetas. Tais leis, então, forneceram umas das bases para Isaac Newton formular a Lei da Gravitação Universal, definindo os processos físicos que governam esse movimento.

Kepler, a partir de estudos do movimento elíptico e da equação da elipse, determinou relações geométricas do movimento elíptico, apresentadas na Figura 3.1. Elas são: r_p = raio do perigeu, r_a = raio do apogeu, f = anomalia verdadeira, a = semieixo maior, u = anomalia excêntrica e b = semieixo menor. E deduziu, a partir da equação da elipse, que quando a anomalia verdadeira é $f = 0^\circ$ o satélite está no ponto da trajetória mais próximo da Terra (perigeu) e quando $f = 180^\circ$ no ponto mais distante (apogeu).

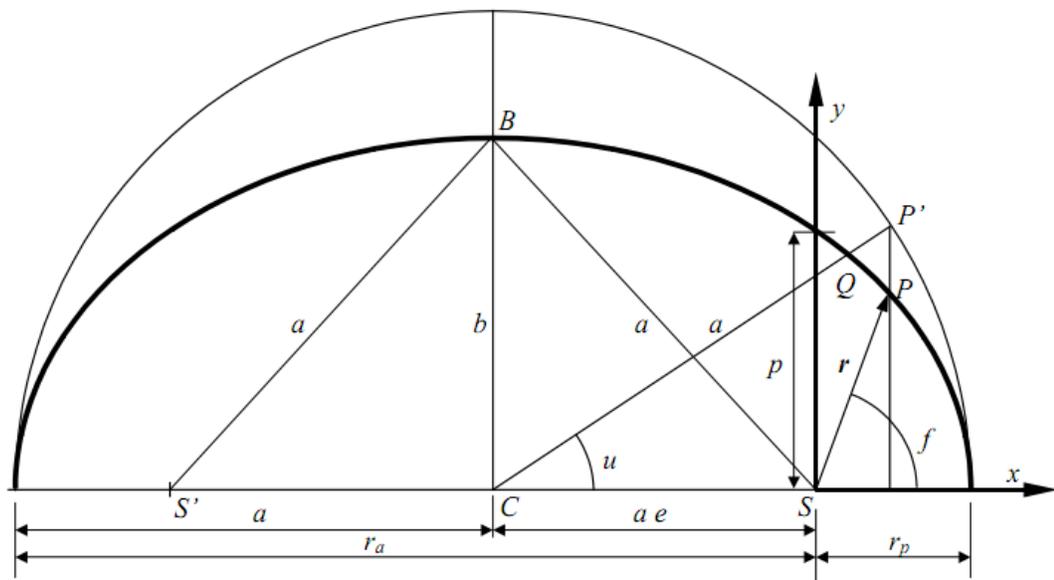


Figura 1: Relações geométricas do movimento elíptico.

A equação de Kepler fornece uma relação entre a anomalia excêntrica e o tempo. Através dela, é possível localizar a posição do satélite em um determinado instante. Outra contribuição importante foi a definição de dados matemáticos que permitem determinar as órbitas dos satélites. Estes dados constituem um sistema de coordenadas que posicionam completamente o satélite e sua órbita. Veja a Figura 3.2 abaixo:

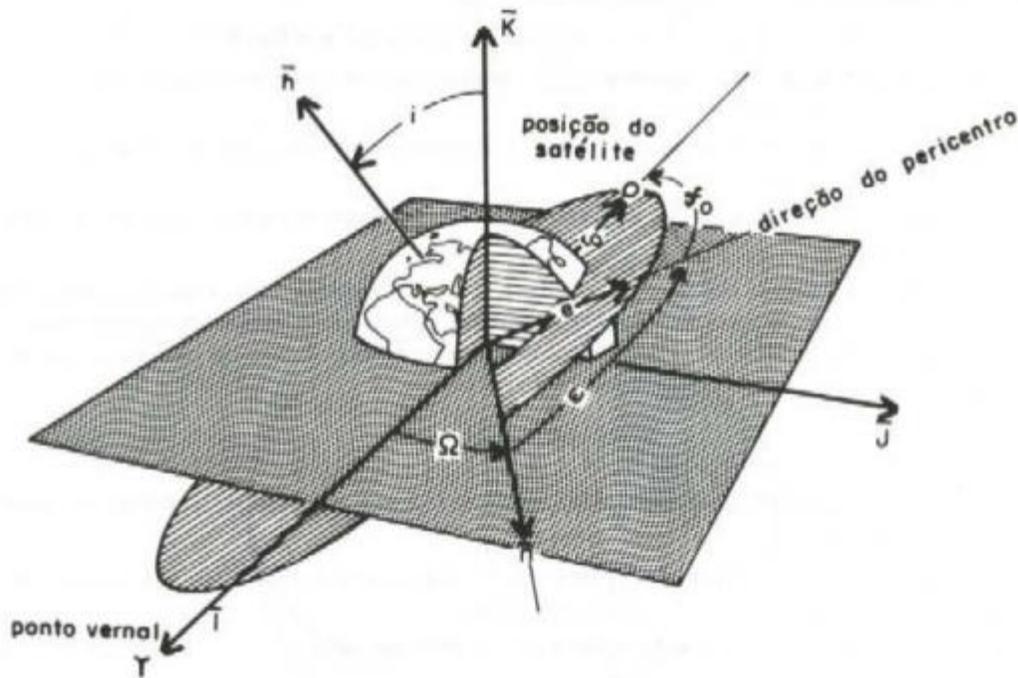


Figura 2: Elementos orbitais clássicos. Fonte: Aula 05 – Mecânica Celeste; Disponível em: <http://www.fis.unb.br/plasmas/aula05.pdf>

Os elementos orbitais clássicos são:

- 1) Semieixo maior (a): define a dimensão da órbita.

Excentricidade (e): define o tipo da órbita.

- 2) Inclinação (I): ângulo entre o momento angular orbital e o versor k .
- 3) Longitude do nodo ascendente (Ω): ângulo, no plano fundamental, entre o versor i e a direção do nodo ascendente.
- 4) Argumento do pericentro (ω): ângulo, no plano da órbita, entre o nodo ascendente e a direção do pericentro.
- 5) Tempo de passagem pelo pericentro (T): instante em que o corpo retorna ao pericentro.

3.2. Manobras Orbitais

De acordo com Prado [7], há três tipos básicos de manobras, a saber:

1) Transferência Inicial

Um satélite artificial é geralmente colocado pelo veículo lançador em uma órbita inicial de baixa altitude (Órbita de Estacionamento); e, a partir daí, o mesmo deve ser elevado à sua órbita final com recursos próprios. A primeira manobra será colocar o satélite em uma órbita próxima da nominal, através da alteração de alguns ou de todos os seus elementos orbitais. Essa primeira fase será chamada de “transferência inicial”.

2) Refinamento Final

Uma vez que o satélite se encontre em uma órbita próxima da nominal, passa-se à fase seguinte da missão, que consiste em aplicar pequenos impulsos até que a órbita nominal seja satisfatoriamente atingida. Aqui são feitas alterações no semieixo, excentricidade e, em alguns casos, no plano orbital. Essa segunda fase será denominada de “refinamento final”; e ela tem, como diferença principal em relação à anterior, o fato de as amplitudes das manobras serem muito menores. Vale a pena enfatizar a necessidade dessa etapa para o atendimento das especificações de um satélite de sensoriamento remoto, pois é preciso que a posição relativa entre o Sol, a Terra e o satélite garantam heliossincronismo (para garantir condições de iluminação constantes na região observada). É preciso também levar em conta que, para garantir a repetibilidade (isto é, garantir que o satélite passe pelo mesmo ponto da Terra após um certo número de órbitas), é necessário impor certas restrições aos valores do semieixo maior.

3) Correções Periódicas

Porém, uma vez atingida a órbita nominal, forças e torques perturbadores indesejáveis, como o arrasto atmosférico, a não esfericidade da Terra, as influências gravitacionais de outros corpos (principalmente a Lua e o Sol), a pressão da radiação da luz solar, etc, fazem com que a órbita se desvie lentamente da nominal e, de tempos em tempos, se faz necessária a execução de manobras de correção. Essa terceira fase consiste em leves alterações dos cinco

elementos orbitais e durará enquanto houver combustível, após o que a missão estará terminada.

Neste trabalho, a princípio, vamos dar um enfoque às manobras de transferência inicial. Existem várias formas de se transferir um satélite de uma órbita para a outra, mas os modelos clássicos são todos baseados nos modelos de propulsão com empuxo infinito, como a Transferência de Hohmann e a Transferência Bi-Elíptica Tri-Impulsiva.

3.2.1. Transferências Orbitais

- **Transferência de Hohmann**

É uma transferência bi-impulsiva entre órbitas keplerianas circulares e coplanares, e ótima apenas quando $r_f/r_0 < 11,93876$ (Figura 3.3). E segue os seguintes passos:

- 1) Na órbita inicial, um impulso ΔV_0 é aplicado na direção do movimento. Com esse impulso o veículo entra em uma órbita elíptica com periapsis r_0 e apoapsis r_f ;
- 2) O segundo impulso ΔV_f é aplicado quando o veículo está no apoapsis e esse impulso circulariza a órbita no raio final desejado.

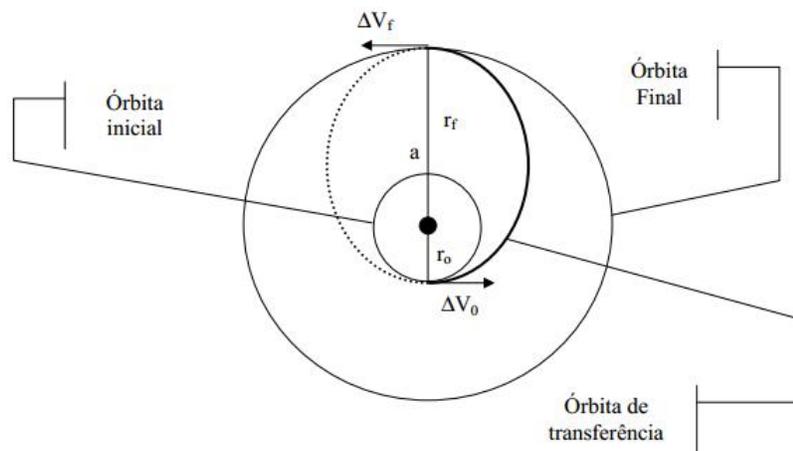


Figura 3: Transferência de Hohmann. Fonte: Santos (2005) [10].

As magnitudes desses impulsos e o tempo de transferência são dados por:

$$\Delta V_0 = V_0 \left| \sqrt{\frac{2 \left(\frac{r_f}{r_0} \right)}{1 + \left(\frac{r_f}{r_0} \right)} - 1} \right|$$

$$\Delta V_f = V_0 \left| 1 - \frac{\sqrt{\frac{2}{\left(\frac{r_f}{r_0} \right) + 1}} \sqrt{\frac{r_0}{r_f}}}{\sqrt{\left(\frac{r_f}{r_0} \right) + 1}} \right|$$

$$t = \frac{1}{2} \left(\frac{1 + \left(\frac{r_f}{r_0} \right)}{2} \right)^{\frac{3}{2}} :$$

- **Transferência Bi-elíptica Tri-impulsiva**

Caso a Transferência de Hohmann não seja ótima devido à razão $(r_f/r_0) > 1.93876$ entre órbitas circulares e coplanares, a transferência bi-elíptica com três impulsos pode apresentar um menor ΔV . Os passos para essa transferência, mostrados na Figura 3.4, podem ser encontrados em Prado (1989) [7].

- 1) O primeiro impulso ΔV_0 é aplicado na órbita inicial para colocar o veículo espacial em uma órbita com periapsis r_0 e apoapsis r_1 ($r_1 > r_0$);
- 2) Quando o veículo estiver no apoapsis, aplica-se o segundo impulso ΔV_1 para aumentar a altura do periapsis para r_f ;
- 3) Enfim, aplica-se o terceiro impulso, na direção oposta ao movimento e com magnitude ΔV_f , quando o veículo espacial está no periapsis e esse impulso circulariza o veículo em sua órbita final desejada.

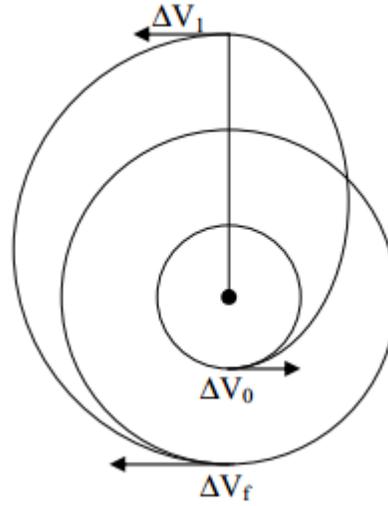


Figura 4: Transferência Bi-elíptica Tri-impulsiva. Fonte: Santos (2005) [10].

Os impulsos ΔV_0 , ΔV_1 e ΔV_f e o tempo Δt de transferência são dados pelas seguintes equações:

$$\Delta V_0 = \varepsilon_0 \left(\sqrt{\frac{2r_1}{1+r_1}} - 1 \right)$$

$$\Delta V_1 = \left(\sqrt{\frac{2r_f}{r_1(r_f+r_1)}} - \sqrt{\frac{2}{r_1(1+r_1)}} \right)$$

$$\Delta V_f = \varepsilon_f \left(\sqrt{\frac{2r_f}{r_1(r_f+r_1)}} - r_f^{\frac{1}{2}} \right)$$

$$\Delta t = \frac{\pi}{2\sqrt{2\mu}} \left(\sqrt{(r_0+r_1)^3} + \sqrt{(r_1+r_f)^3} \right)$$

Onde:

$$\epsilon_0 = \text{sinal}(r_1 - 1)$$

$$\epsilon_f = \text{sinal}(r_1 - r_f)$$

4. O Software Orbiter

O Orbiter é um software livre de simulação tridimensional de vôos espaciais. Tal software permite-nos observar fielmente certos fenômenos físicos, tais como a gravidade, perturbações atmosféricas, atração gravitacional dos planetas, etc. É possível dentro do ambiente do software Orbiter incluir e modificar espaçonaves e satélites, de modo a fornecer um leque considerável de opções de simulação. A interface de programação nos garante precisão nas simulações e os vôos podem ser gravados para revisão posterior.

A Figura 5 mostra o menu inicial do software Orbiter 2010:



Figura 5: Interface inicial do software Orbiter 2010.

Para nos familiarizarmos com o software Orbiter executaremos alguns exemplos iniciais.

Exemplo 1: A princípio, iremos usar a nave Atlantis, do software Orbiter para realizar as primeiras manobras deste trabalho. Para abrir a interface de lançamento desta espaçonave, abre-se a pasta “Space Shuttle Atlantis” dentro do menu do software Orbiter , e seleciona-se “Launch Orbiter”. Então, o simulador será executado.

Em seguida, através do atalho de teclado F1, é possível acessar os modos de cabine da espaçonave Atlantis, mostradas na Figura 6 e Figura 7.



Figura 6: Modo de cabine da espaçonave Atlantis em 3D.



Figura 7: Modo de cabine da espaçonave Atlantis em 2D.

As Figuras 6 e 7 mostram as opções de visualização das cabines virtuais da espaçonave Atlantis, alternadas pelo F8 do teclado. A primeira (Figura 6) é a visão interior da espaçonave, composta por botões e por dois monitores multifuncionais (MFDs); a segunda (Figura 7) mostra apenas dois monitores multifuncionais. É possível controlar a espaçonave manualmente através de comandos do teclado, do joystick ou utilizando os botões do console mostrado no modo de cabine tridimensional. Porém, o enfoque do trabalho é na interface de programação que o software Orbiter possui, que nos permite fazer simulações precisas e mais próximas da realidade do agendamento e despacho de manobras orbitais. Esta será descrita a seguir.

4.1. Interface de Programação

Através do comando de atalho Ctrl+F4, podemos selecionar a opção “Lua Console Window” e apertar o botão OK, o que nos abrirá o console de programação do software Orbiter, como se vê na Figura 8.

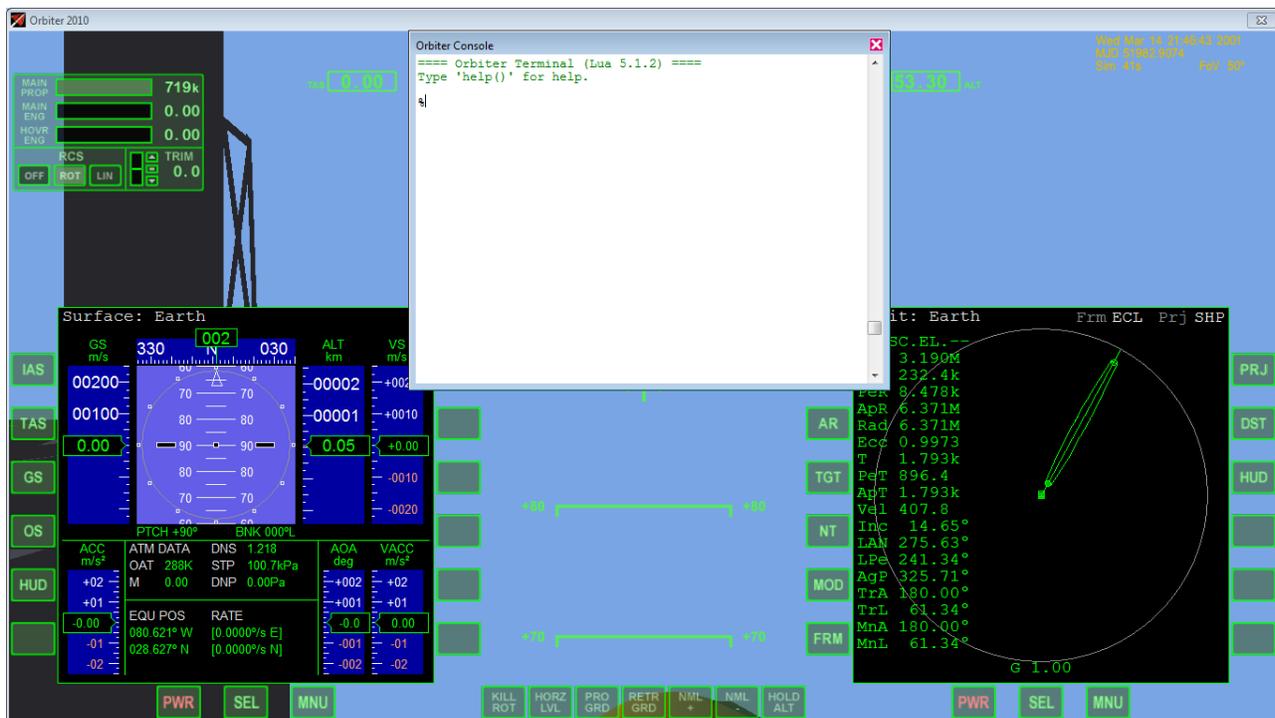


Figura 8: Interface de Programação do software Orbiter 2010.

A janela do console nos permite inserir os comandos para a simulação do voo ou então importar o algoritmo em questão de um arquivo de extensão “.lua” através do comando run <local do script>. A linguagem de programação utilizada pelo software Orbiter é a Lua, que é uma linguagem orientada a objetos, leve e reflexiva.

4.2. Voo de demonstração

Podemos realizar uma simulação de voo pré-existente para esclarecer e ilustrar os comandos do software Orbiter, utilizando-se do script “atlantis.lua”, como vemos na Figura 9.

```

Orbiter Console
==== Orbiter Terminal (Lua 5.1.2) ====
Type 'help()' for help.

%run('atlantis/launch')
Space Shuttle Atlantis ascent autopilot
Run function 'launch()' to start
Global variables (can be adjusted before launch):
vif: target spacecraft interface (default: current)
orbit_alt: target altitude [m] (default: 300e3)
azimuth: launch azimuth [rad] (default: 90*RAD)
%launch()
Ascent autopilot initiated.
Attached to: STS-101
Launch azimuth: 90deg.
Target altitude: 300km
job id=1 (1 jobs)
initial pitch
  setbank initiated (180deg, tmax=500, tol=0)
job id=1 (1 jobs)
  setyaw initiated (90deg, tmax=500, tol=0)
job id=2 (2 jobs)
pitch 70
  setpitch initiated (70deg, tmax=20, tol=0)
job id=3 (3 jobs)

```

Figura 9: Console de Programação do software Orbiter executando a simulação.

O comando `%run('atlantis/launch')` irá carregar o script “launch.lua”, presente no diretório “Atlantis”. Ao carregar tal script, ainda é possível alterar determinadas variáveis como a altitude da órbita, o ângulo de lançamento, entre outros.

O comando “`%launch()`” irá então executar a função “`launch ()`”, presente no script “launch.lua”.

```

function launch ()
  term.out('Ascent autopilot initiated.')
  term.out('Attached to: '..vif:get_name())
  term.out('Launch azimuth: '..azimuth*DEG..'deg.')
  term.out('Target altitude: '..orbit_alt*1e-3..'km')
  if vif:get_classname() ~= 'Atlantis' then
    term.out('**** WARNING: Unexpected vessel type')
    term.out('**** '..vif:get_classname())
  end
  ascent(vif)
  if do_oms1==true then
    oms_burn1 (vif,orbit_alt)
    if do_oms2==true then
      oms_burn2 (vif,orbit_alt)
    end
  end
  term.out('Exit launch autopilot.')
end

```

Figura 10: Função “launch()” a ser executada.

O script mostrado na Figura 10 segue uma sequência clara de execução de funções, de modo a permitir que a espaçonave entre em órbita com a Terra. É possível acompanhar todo o trajeto da espaçonave pelo software Orbiter o que facilita a compreensão e a correção de possíveis erros. Através dos comandos da linguagem Lua, é possível otimizar a simulação e introduzir funções que executam as ordens dadas de maneira precisa, o que faz do software Orbiter não apenas um simulador preciso como uma ferramenta interessante para o desenvolvimento de lançamentos e voos agendados.

4.3. Limitações do Software Orbiter

O software Orbiter possui certas limitações devido à reduzida quantidade de corpos celestes, já que o sistema solar apresentado consiste no Sol, os oito planetas e suas luas principais. Planetas-anões, asteroides e cometas não estão incluídos no sistema solar junto com o simulador, porém podem ser incluídos como “add-nos”. O software Orbiter foi designado, principalmente, para a simulação de vôo de órbita de baixa altitude e espaço interplanetário. A simulação de vôo atmosférico não é tão boa quanto à de simuladores de vôo dedicados a tal tarefa, e o vôo interestelar ainda não possui muito suporte, o que pode mudar no futuro.

5. Metodologia Aplicada

5.1. Estudo do Agendamento e Despachamento de Manobras Orbitais

Segundo Rodrigues [9], as atividades de operação de satélites artificiais podem ser comandadas por centros de controle terrenos (Figura 11) denominados estações de solo. Todas as ações de controle são planejadas, coordenadas e executadas por essas estações. Planejar e executar, neste caso, é o mesmo que agendar e despachar.

As estações possuem antenas e outros equipamentos necessários para estabelecer a comunicação com os satélites monitorados. Quando o satélite se encontra sobre a região de visibilidade dessas antenas, os sinais de telemetria enviados pelo satélite são recebidos, analisados e armazenados com o auxílio de softwares. As estações também enviam telecomandos e executam as medidas de localização do satélite em tempo real. Para que

estas atividades sejam realizadas de forma satisfatória é muito importante que seja feito um minucioso planejamento (agendamento) e uma cuidadosa execução (despachamento).



Figura 11: Estação de rastreamento de Cuiabá.

5.2. Sequência para o Agendamento e Despachamento de Manobras Orbitais

Com base no estudo do artigo do Dr. Hélio Koiti Kuga aplicado ao satélite CBERS 1 do INPE [12], e no trabalho realizado, anteriormente, por Mary Ellen Rodrigues sobre este tema [9], a sequência para o agendamento e despachamento de manobras orbitais segue os seguintes passos:

Passo 1: As manobras são planejadas e desenvolvidas nas estações terrenas, também conhecidas como estações de solo;

Passo 2: Depois, elas são ordenadas e organizadas de acordo com a necessidade de execução de cada uma delas;

Passo 3: A seguir os dados são enviados, um por vez, ao satélite, porém cada um contendo muitas informações. Ao chegar ao satélite, eles são armazenados para serem executados (despachados) segundo o planejamento (agendamento);

Passo 4: Depois da execução dos dados telecomandados, a estação confere os dados de telemetria que revelam o estado atual de funcionamento do satélite a fim de analisar os resultados da missão;

Passo 5: Todos os arquivos históricos de missão são armazenados em bancos de dados para uma posterior análise.

5.3. Modelagem e simulação do agendamento e despachamento da transferência de Hohmann utilizando o console de programação Lua do ambiente Orbiter

A modelagem e simulação do agendamento e despachamento de uma manobra espacial dentro do ambiente Orbiter, utilizando o console Lua deve seguir comandos pré-existentes na biblioteca, mostrada na Figura 12.

Vessel methods

This list contains methods that can be applied to objects of type vessel. In the following, variable *v* is assumed to be a valid vessel object (see [vessel.get_interface](#)).

v.get_handle	Returns a handle to the vessel object.
v.send_bufferedkey	Sends a keystroke message to the vessel.
v.get_gravityref	Returns a handle to the main contributor of the gravity field at the vessel's current position.
v.get_surfaceref	Returns a handle to the surface reference object (planet or moon).
v.get_altitude	Returns the current vessel altitude.
v.get_pitch	Returns the current pitch angle.
v.get_bank	Returns the current bank angle.
v.get_yaw	Returns the current yaw (heading) angle.
v.get_angvel	Returns the current angular velocity components.
v.set_angvel	Resets the current angular velocity components.
v.get_elements	Returns osculating orbital elements.
v.get_elementsex	Returns osculating orbital elements and additional orbital parameters.
v.set_elements	Set vessel state (position and velocity by means of a set of osculating orbital elements).
v.get_prograddir	Returns the direction of the orbital velocity vector in vessel coordinates.

Figura 12: Biblioteca de métodos (comandos) do Orbiter.

A programação e a modelagem de uma manobra para simulação dentro do ambiente Orbiter deve conter implementações da dinâmica da manobra (*item 3.2.1.*), comandos para controle de atitude e comandos para propulsão da nave.

- Dinâmica da Manobra

O primeiro impulso, Δv , aplicado na nave será implementado no algoritmo da seguinte forma:

$$ht = ((MI/(alt+R_earth))*sqrt)*(((2*r_2)/((alt+R_earth)+r_2)*sqrt)-1))$$

Onde, “*MI*”, “*R_earth*”, e “*r_2*” são constantes declaradas no início do script, que representam, respectivamente, constante gravitacional, raio da Terra, e o raio final da transferência. E “*alt*” é uma variável, que representa a distância da superfície da Terra até o raio inicial.

- Controle de Atitude

A princípio, o controle de atitude será o mesmo utilizado para o lançamento da nave Atlantis, no “Exemplo 1”(*item 4.*), mostrado na Figura 13.

```

12 verbose = 1      -- set to nil to suppress console output
13
14 bank_a = 0.3
15 bank_b = -2.2
16 pitch_a = 0.3
17 pitch_b = 2.3
18 yaw_a = -0.3
19 yaw_b = 3.2
20
21 RCSMODE = {PITCH=0,BANK=1,YAW=2}
22
23 -- activate RCS for rotation in one of the vessel's three principal axes
24 -- v: vessel interface
25 -- mode: one of the RCSMODE keys
26 -- level: thrust level (0..1)
27
28 -function SetRCS (v, mode, level)
29 -   if mode == RCSMODE.PITCH then
30 -     if level >= 0 then
31 -       v:set_thrustergrplevel (THGROUP.ATT_PITCHDOWN, 0);
32 -       v:set_thrustergrplevel (THGROUP.ATT_PITCHUP, level);
33 -     else
34 -       v:set_thrustergrplevel (THGROUP.ATT_PITCHUP, 0);
35 -       v:set_thrustergrplevel (THGROUP.ATT_PITCHDOWN, -level);
36 -     end
37 -   elseif mode == RCSMODE.BANK then
38 -     if level >= 0 then
39 -       v:set_thrustergrplevel (THGROUP.ATT_BANKRIGHT, 0);
40 -       v:set_thrustergrplevel (THGROUP.ATT_BANKLEFT, level);
41 -     else
42 -       v:set_thrustergrplevel (THGROUP.ATT_BANKLEFT, 0);
43 -       v:set_thrustergrplevel (THGROUP.ATT_BANKRIGHT, -level);
44 -     end
45 -   elseif mode == RCSMODE.YAW then
46 -     if level >= 0 then
47 -       v:set_thrustergrplevel (THGROUP.ATT_YAWRIGHT, 0);
48 -       v:set_thrustergrplevel (THGROUP.ATT_YAWLEFT, level);
49 -     else
50 -       v:set_thrustergrplevel (THGROUP.ATT_YAWLEFT, 0);

```

Figura 13: Script em Lua para o controle de atitude da nave Atlantis.

- Propulsão do foguete

A princípio, os comandos de propulsão implementados serão baseados nos comandos de lançamento da nave Atlantis (Figura 14)

```

99 -----
100 -- The complete autopilot
101 -----
102
103 -function launch ()
104   term.out('Ascent autopilot initiated.')
105   term.out('Attached to: '..vif:get_name())
106   term.out('Launch azimuth: '..azimuth*DEG..'deg;')
107   term.out('Target altitude: '..orbit_alt*1e-3..'km')
108   if vif:get_classname() ~= 'Atlantis' then
109     term.out('*** WARNING: unexpected vessel type')
110     term.out('*** '..vif:get_classname())
111   end
112   ascent(vif)
113   if do oms1==true then
114     oms_burn1 (vif,orbit_alt)
115   if do oms2==true then
116     oms_burn2 (vif,orbit_alt)
117   end
118   end
119   term.out('Exit launch autopilot.')
120 end
121
122 -----
123 -- First part of ascent autopilot.
124 -- Convers launch to ET separation
125 -----
126
127
128 -function ascent (v)
129   proc.bg(met_counter) -- launch MET counter
130
131   proc.wait_simdt(3);
132   v:set_thrustergrplevel (THGROUP.MAIN, 1)
133   comment = 'Main engine ignition'
134   proc.wait_simdt(4) -- SSME ignition
135

```

Figura 14: Script de lançamento da nave Atlantis.

Conclusão

O objetivo deste trabalho foi iniciar o estudo da modelagem e simulação do agendamento e despacho de manobras orbitais por tempo e por eventos, particularmente através do software Orbiter. Este trabalho ainda está em andamento, mas através do estudo das referências, da realização dos exemplos e da análise de seus resultados pode-se prever que apesar das limitações do software, na modelagem e simulação do agendamento e despacho de manobras orbitais, obteremos alta precisão nos resultados devido ao realismo que o software proporciona.

Posteriormente, com base neste programa, objetiva-se realizar a modelagem e simulação do agendamento e despacho da Transferência de Hohmann por tempo e eventos, e depois de manobras em satélites reais, como o CBERS3 do INPE.

Referências Bibliográficas

[1] KUGA, H.K., RAO, K.R., CARRARA, V. **Introdução à Dinâmica Orbital (2ª. ed.)**. INPE, São José dos Campos, SP, 2008.

[2] WINTER, O.C. (editor). **Apostila de Dinâmica Orbital e Planetologia**, UNESP, Guaratinguetá, SP, 2004.

[3] PRADO, A.F.B.A, KUGA, H.K. **Fundamentos de Tecnologia Espacial**. INPE, São José dos Campos, SP, 2001.

[4] Site <http://www.fis.unb.br/plasmas/aula05.pdf>. Acessado em Janeiro/2012.

[5] SCHWEIGER, M. **Interview-with-Orbiter-Sim-Creator**. Disponível em: <http://techhaze.com/2010/06/interview-with-orbiter-sim-creator-martin-schweiger/>. Acessado em 10/Julho/2012.

[6] SCHWEIGER, M. **Orbiter Space Flight Simulator 2010 Edition User Manual**. Disponível em: <http://orbit.medphys.ucl.ac.uk/download/Orbiter.pdf> . Acessado em 10/Julho/2012.

[7] PRADO, A.F.B.A. **Análise, Seleção e Implementação de Procedimentos que visem Manobras Ótimas de Satélites Artificiais**. Dissertação de Mestrado. INPE, São José dos Campos, SP, 1989.

[8] FONSECA, J.B.S. **Manobras de Órbita e Atitude de Satélites Artificiais**. In: SEMINÁRIO DE INICIAÇÃO CIENTÍFICA DO INPE (SICINPE), INPE, São José dos Campos, SP, 2010, Papel. Disponível em: <<http://urlib.net/8JMKD3MGP7W/38A2F7B>>. Acessado em Junho/2012.

[9] RODRIGUES, M.E. **Modelagem e simulação do agendamento e despacho de Manobras espaciais por tempo e eventos.** Relatório Final de Projeto de Iniciação Científica. (PIBIC/CNPQ). INPE, São José dos Campos, SP, Julho de 2011.

[10] SANTOS, D.P.S. **Aplicações em Manobras Espaciais do Problema de Múltiplos Encontros.** Dissertação de Mestrado. INPE, São José dos Campos, SP, 2005.

[11] SANTOS, M.P. **Introdução à Simulação Discreta.** Material de apoio. Universidade do Estado do Rio de Janeiro, Rio de Janeiro, RJ, 1999.

[12] KUGA, H.K., ORLANDO, V. **Orbit Control of CBERS-1 Satellite at INPE.** In: XVI INTERNATIONAL SYMPOSIUM ON SPACE FLIGHT DYNAMICS, Pasadena, USA, 2001, Pasadena, USA.

[13] ZHU, K.J., LI, J.F., BAOYIN, H.X., **Satellite Scheduling Considering Maximum Observation Coverage Time and Minimum Orbital Transfer Fuel Cost.** Acta Astronautica, Janeiro/2009.

[14] RODRIQUEZ, M.D., BORRAJO, D., MEZIAT, D., **An AI Planning- based Tool for Scheduling Satellite Nominal Operations.** AI Magazine, Volume 25, Number 4. 2004.

[15] Ierusalimschy, R. **Uma Introdução à Programação em Lua.** Disponível em: <http://www.lua.org/doc/jai2009.pdf>. Acessado em 03/2013.

[16] Site: em http://www.keplerproject.org/docs/apostila_lua_2008.pdf. Acessado em 05/2013.