





**INSTITUTO NACIONAL DE PESQUISAS**

**ESPACIAIS**

**Bolsa de Iniciação científica CNPq**

**Relatório Final**

**“IDENTIFICAÇÃO DE ESTRELAS**

**OBSERVADAS POR SENSORES**

**ESTELARES”**

**Gustavo Baldo Carvalho**

**Aluno da Escola de Engenharia de São Carlos da Universidade  
de São Paulo (EESC-USP) - Bolsa PIBIC / CNPq**

**Orientador: Dr. Roberto Vieira da Fonseca Lopes, Pesquisador,  
DMC-INPE**

**São José dos Campos - SP, 30 / 06 / 96**

# Agradecimentos

Gostaria de agradecer no presente momento às várias partes e pessoas que, direta ou indiretamente, envolveram-se com o desenvolvimento deste trabalho e contribuíram para o seu mérito.

Em primeiro lugar, a **DEUS** por todas as oportunidades, crescimentos desafios, vitórias, derrotas e inspirações que me levaram aos melhores caminhos.

Ao **Instituto Nacional de Pesquisas Espaciais (INPE)** pelas facilidades e apoio oferecidos, assim como ao **Ministério da Ciência e Tecnologia**.

Ao **Dr. Roberto Vieira da Fonseca Lopes** pela orientação segura, confiante, entusiástica e amiga, dispondo de seu tempo e acreditando no desenvolvimento do trabalho que aqui se apresenta.

Ao **CNPq** e aos **Departamentos responsáveis no INPE** pela oportunidade de participar no **Programa Institucional de Bolsas de Iniciação Científica (PIBIC)** e por acreditarem em minha capacidade pessoal.

À **EESC-USP-São Carlos** pela qual muitas portas foram abertas e que ainda estão por abrir.

Ao **Observatório de São Carlos (CDA)-CDCC** e responsáveis por todas as oportunidades de experiência e conhecimentos adquiridos.

A todos os colegas no **INPE** pelo apoio e força que me passam, e em especial ao **Dr. Luiz Carlos Gadelha de Souza** que abriu-me as portas do **INPE**, ao **Dr. Antônio Fernando Bertachini de A. Prado**, **Dr. Kondapalli Rama Rao** e **Msc. Mário César Ricci** pela amizade, ajuda e apoio.

Aos Professores **Dr. Fernando Martin Catalano** e **Dr. Michael George Maunsell** do **LAE (laboratório de Aeronaves)-EESC-USP**, pelo apoio e ajuda nos contatos iniciais com o **INPE**.

Aos meus pais, **Edson Carvalho** e **Maria Lúcia Baldo Carvalho** por todos os esforços e sacrifícios, ajuda, crença, apoio e força que sempre me deram para que eu chegasse até onde cheguei e ainda estou por chegar.

À minha namorada **Liliane Machado Bromberg** pela ajuda na confecção deste trabalho e por seu apoio, paciência e carinho; e ainda a seus pais, **Luiz Ricardo Terra Bromberg** e **Ana Maria Machado Bromberg** pelo grande apoio e ajuda.

A todos os pioneiros que lutaram e deram suas vidas pela conquista do ar e do espaço.

E em especial, a todos os meus amigos que se propuseram de **espírito e alma** a estar sempre me ajudando.

Obrigado,

**Gustavo Baldo Carvalho - 30 / 06 / 96**

## Abstract

*The aim of this work is to present an introductory study about **tree-axis attitude determination** of Earth artificial satellites from **star sensor** observations. Special emphasis was put in the **star identification** problem. It was solved by the **Direct Match Technique**, whose error window radius was optimized by the **Multi-criterion method**. Numerical simulation on a personal computer was carried out with **PC-Matlab**<sup>®</sup> based on a real star catalog. The results allow us to analyse the performance of the procedure for different star configuration, a priori attitude precision and star sensor precision.*

# Resumo

## IDENTIFICAÇÃO DE ESTRELAS OBSERVADAS POR SENSORES ESTELARES

O trabalho analisa determinações de atitude de um satélite artificial estabilizado triaxialmente, com identificação de estrelas por um sensor estelar rastreador CCD, e complementa o projeto do sensor estelar do INPE.

O método de comparação angular direta foi selecionado para a identificação das estrelas. Há necessidade de uma atitude pré-estimada, podendo ser obtida por sensores menos precisos. A técnica da comparação angular direta compara distâncias angulares de estrelas vistas pelo sensor e estrelas num catálogo, realizando um círculo de raio determinado em torno da posição vista. Possibilidades: identificação correta, identificação errônea, identificação ambígua e não identificação. O raio do círculo é escolhido para maximizar a probabilidade de identificação correta e minimizar a probabilidade de identificação errônea com base na distribuição de **Poisson** para as estrelas e de **Gauss** para o erro de observação, pelo método de otimização de múltiplos critérios.

A atitude é estimada por um método mínimos quadrados que utiliza dois ou mais versores medidos em dois referenciais diferentes. Os sistemas de coordenadas utilizados são o azimutal, vetorial-cartesiano e equatorial, sendo utilizados nos referenciais inercial e do satélite. A representação de atitude é feita através de ângulos de **Euler**. O transporte de atitude é feito por matrizes de rotação.

O catálogo de estrelas é dado no referencial inercial pelo Observatório Nacional. Selecionaram-se cinco regiões do céu onde a densidade de estrelas com magnitude visual menor que cinco é relativamente alta.

Os resultados a partir de testes do *software* descrevem o comportamento do mesmo, com os quais tiramos as conclusões que servem de base para desenvolvimentos futuros.

# Índice

	página
Título.....	I
Agradecimentos.....	II
Abstract.....	III
Resumo.....	IV
Índice.....	V
Lista de figuras.....	VII
Lista de tabelas.....	VIII
Lista de abreviaturas.....	VIII
Lista de símbolos.....	IX
Capítulo I.....	1
1. Introdução.....	2
Capítulo II.....	7
2. Conceitos básicos.....	8
2.1 Sistemas de Referência.....	8
- Esfera celeste.....	8
- Descrição dos sistemas.....	10
- Descrição dos referenciais.....	12
- Transformações.....	12
- Sistemas de representação de atitude e referencial.....	15
2.2 Problema da determinação de atitude.....	18
2.3 Sensores típicos de atitude.....	18
2.4 Sensor estelar.....	20
2.4.1 Sensores estelares.....	20
2.4.2 Sensor estelar alvo deste trabalho.....	22
- Precisão do sensor.....	25
2.5 Mapa estelar.....	27
Capítulo III.....	31
3. Fundamentos teóricos.....	32
3.1 O algoritmo.....	32
- Determinação de atitude triaxial exata.....	32
- Determinação de atitude triaxial otimizada.....	33
3.2 Métodos clássicos de identificação de estrelas.....	34
3.3 Técnica dos múltiplos critérios.....	36
Capítulo IV.....	38
4. Descrição do procedimento e resultados finais.....	39
4.1 Descrição do procedimento.....	39
- Diagrama de acesso de rotinas principais.....	39
- Fluxograma do programa principal.....	40
- Rotinas auxiliares do sistema.....	41
- Seção de testes.....	45
4.2 Resultados finais.....	53
Capítulo Final.....	56
5. Comentários finais.....	57
- Atividades realizadas durante o decorrer do trabalho.....	57
- Opiniões gerais sobre a experiência.....	58
- Continuação do trabalho.....	58
6. Referências bibliográficas e <i>software</i> .....	59
Apêndices.....	61
A - Tipos de estabilizações de satélites.....	A1

B - Missões, aquisição e controle de atitude e torques atuantes.....	B1
C - Vetores e Matrizes.....	C1
D - Cálculo de área da região observada e sua densidade de estrelas.....	D1
E - Magnitudes.....	E1
F - Relações empíricas para o cálculo de $\phi$ e $\lambda$ em coordenadas da matriz eletrônica.....	F1
G - Tipos de representação de atitude e peculiaridades matemáticas.....	G1
- Ângulos de Euler.....	G1
- Matriz de rotação.....	G1
- Eixo e ângulo de rotação.....	G2
H - Geometria esférica.....	H1
I - Listagem de testes realizados para o <i>software</i> .....	I1
- Constelação: Crucis - requisitos 1 (testes iniciais).....	I1
- Constelação: Orionis.....	I2
- Constelação: Pegasus.....	I2
- Constelação: Scorpii.....	I2
- Constelação: Tauri.....	I3
- Constelação: Crucis - requisitos 2 (requisitos ótimos).....	I3
J - Listagem de subrotinas.....	J1



# Lista de figuras

Figura	Seção
1 - Satélite ISEE-1 estabilizado triaxialmente.....	(Cap.I.1.)
2 - Sonda ATS-6 estabilizada por rotação.....	(Cap.I.1.)
3 - Seqüência para o caso real de funcionamento do sistema de determinação de atitude.....	(Cap.I.1.)
4 - Seqüência para a simulação de funcionamento do sistema de determinação de atitude.....	(Cap.I.1.)
5 - Esfera celeste inercial.....	(Cap.II.2.1.)
6 - Esfera celeste do satélite.....	(Cap.II.2.1.)
7 - Codificação de coordenadas na esfera celeste.....	(Cap.II.2.1.)
8 - Sistema equatorial de referência.....	(Cap.II.2.1.)
9 - Sistema azimutal anti-horário.....	(Cap.II.2.1.)
10 - Sistema vetorial-cartesiano na esfera celeste.....	(Cap.II.2.1.)
11 - Sistema cartesiano plano.....	(Cap.II.2.1.)
12 - Sistema vetorial-cartesiano e equatorial de referência.....	(Cap.II.2.1.)
13 - Sistema vetorial-cartesiano e cartesiano plano.....	(Cap.II.2.1.)
14 - Rotações sucessivas em ângulos de Euler.....	(Cap.II.2.1.)
15 - Determinação de atitude de satélites rotativos.....	(Cap.II.2.3.)
16 - Determinação de atitude de satélites triaxialmente estabilizados.....	(Cap.II.2.3.)
17 - Determinação de versores de apontamento de estrela s a partir do satélite..	(Cap.II.2.3.)
18 - Giroscópios: sensores inerciais.....	(Cap.II.2.3.)
19 - Diagrama simplificado de um sensor estelar com disposição de fendas em "V".....	(Cap.II.2.4.1)
20 - Rastreador estelar de movimentos mecânicos.....	(Cap.II.2.4.1.)
21 - Diagrama simplificado de um sensor fixo com varredura eletrônica.....	(Cap.II.2.4.1.)
22 - Seqüência de detecção do sensor rastreador estelar CT - 401.....	(Cap.II.2.4.2.)
23 - Esfera celeste e sistema de coordenadas locais na matriz eletrônica.....	(Cap.II.2.4.2.)
24 - Relação de coordenadas no sistema óptico e matriz eletrônica.....	(Cap.II.2.4.2.)
25 - Matriz CCD do sensor de varredura eletrônica.....	(Cap.II.2.4.2.)
26 - Movimentos em torno dos eixos do sensor e do satélite - precisão do sensor.....	(Cap.II.2.4.2.)
27 - Regiões do céu escolhidas para os testes em Matlab.....	(Cap.II.2.5.)
28 - Posicionamento e determinação de regiões na esfera celeste.....	(Cap.II.2.5.)
29 - Constituição das regiões escolhidas.....	(Cap.II.2.5.)
30 - Matriz de representação de uma região do céu.....	(Cap.II.2.5.)
31 - Bases ortonormais construídas através dos versores medidos.....	(Cap.III.3.1.)
32 - Representação gráfica da técnica da comparação direta.....	(Cap.III.3.2.)
33 - Probabilidades de ocorrência para os vários tipos de identificação com relação ao raio escolhido na técnica da comparação direta.....	(Cap.III.3.2.)
34 - Faixas de aplicação ótimas às probabilidades de identificação.....	(Cap.III.3.2.)
35 - Função ótima, otimização e equilíbrio dos critérios para as probabilidades de identificação..	(Cap.III.3.3.)
36 - Seqüência para a simulação de funcionamento do sistema de determinação de atitude.....	(Cap.IV.4.1.)
37 - Representações gráficas dos resultados do primeiro teste.....	(Cap.IV.4.1.)
38 - Representações gráficas dos resultados do segundo teste.....	(Cap.IV.4.1.)
39 - Comparação entre % de estrelas vistas e % de estrelas identificadas.....	(Cap.IV.4.2.)

40 - Comparação entre % de estrelas identificadas corretamente para 1 <sup>o</sup> e 2 <sup>o</sup> de precisão do sensor de pré estimativa.....	(Cap.IV.4.2.)
41 - % de estrelas identificadas erroneamente e % de identificações ambíguas (requisito de 30%).....	(Cap.IV.4.2.)
42 - % de estrelas não identificadas e % de ocorrências de precisões obtidas em cálculo.....	(Cap.IV.4.2.)
43 - % de estrelas ambíguas em cálculo (requisito de 15%) e faixas ótimas de aplicação das probabilidades.....	(Cap.IV.4.2.)
A1 - Sonda OSO-8: estabilização composta rotativa- triaxial.....	(Ap. A)
A2 - Sonda RAE-2: estabilização por gradiente gravitacional.....	(Ap. A)
B1 - Forças Atuantes num elemento de massa do satélite.....	(Ap. B)
C1 - Sistema ortonormal de coordenadas.....	(Ap. C)
D1 - Elemento de área para a integração do cálculo de área da região.....	(Ap. D)
G1 - Rotação bidimensional de mudança de base.....	(Ap. G)
G2 - Representação de um vetor em ângulo e eixo de rotação.....	(Ap. G)
H1 - Triângulo esférico.....	(Ap. H)

## Lista de tabelas

Tabelas	Seção
Fig. 34 - Faixas de aplicação ótimas às probabilidades de identificação.....	(Cap.III.3.3.)
Fig. 43 - % de estrelas ambíguas em cálculo (requisito de 15%) e faixas ótimas de aplicação das probabilidades.....	(Cap.IV.4.2.)
Listagem de testes realizados para o <i>software</i> .....	(Ap. I)

## Lista de abreviaturas

Arccos - Função arco-cosseno
Arcsen - Função arco-seno
c - cosseno
CCD - Charge Couple Device
CDA - Centro de Divulgação da Astronomia
Cos - Função cosseno
det - determinante
EESC - Escola de Engenharia de São Carlos
Fig. - Figura
gps - global positioning system
INPE - Instituto Nacional de Pesquisas Espaciais
Mag. - Magnitude
parsec - Paralaxe-segundo (unidade de distância por paralaxe)
s - seno
Sen - Função seno
tr - traço
USP - Universidade de São Paulo

# Lista de símbolos

- ângulos:

- $az$  - azimute
- $\alpha$  - ascensão reta
- $\alpha_h$  - ascensão reta em horas no referencial inercial
- $\alpha_{hs}$  - ascensão reta em horas no referencial do satélite
- $\delta$  - declinação no referencial inercial
- $\delta_s$  - declinação no referencial do satélite
- $h$  - altura do sistema azimutal
- $\varphi, \theta, \psi$  - rotações sucessivas de Euler nos eixos cartesianos triaxiais
- $\theta$  - ângulo qualquer; ângulo de coordenada esférica
- $\lambda$  - ângulo de estrela referente ao eixo  $y$  do sensor
- $\phi$  - ângulo de estrela referente ao eixo  $x$  do sensor ; ângulo de coordenada esférica
- $\Phi, \Theta, \Lambda$  - ângulos esféricos

- eixos:

- $\hat{n}_i, \hat{n}_i', \hat{n}_i''$  - eixos de rotação de Euler
- $x, y$  - eixos do sistema cartesiano plano;
- $z$  - eixo do sistema cartesiano tridimensional
- $x', y', z'$  - eixos do sistema cartesiano tridimensional depois da rotação  $\varphi$
- $x'', y'', z''$  - eixos do sistema cartesiano tridimensional depois da rotação  $\theta$
- $x''', y''', z'''$  - eixos do sistema cartesiano tridimensional depois da rotação  $\psi$
- $X, Y, Z$  - Eixos do sistema triaxial cartesiano inercial
- eixos do sistema tridimensional cartesiano geral

- sistemas:

- $\epsilon, \epsilon'$  - bases
- $\{\hat{r}_1, \hat{r}_2, \hat{r}_3\}$  - base ortonormal inercial
- $\{\hat{s}_1, \hat{s}_2, \hat{s}_3\}$  - base ortonormal no referencial do satélite
- $xy$  - Sistema cartesiano plano
- $ZY$  - plano no referencial inercial que contém os eixos  $Z$  e  $Y$
- $XYZ$  - Sistema triaxial cartesiano

- definições::

- $C.G._{Terra}$  - Centro de gravidade da Terra
- $C.G.$  - centro de gravidade
- $\partial$  - diferencial parcial
- $\Delta$  - variação
- $f$  - distância focal da lente
- $G$  - constante gravitacional universal
- $N, S$  - Norte, Sul
- $\gamma$  - ponto de equinócio vernal
- $\Sigma$  - somatória
- $\int$  - símbolo de integração
- $h$  - hora
- $q$  - densidade estelar
- $M_T$  - massa da Terra
- $\rho$  - densidade de estrelas; raio de coordenada esférica
- $v$  - magnitude visual aparente
- $B$  - magnitude azul
- $m_i$  - magnitude instrumental

- matrizes::

- $A$  - matriz de atitude
- $A^+$  - matriz adjunta de  $A$
- $A^T$  - matriz transposta de  $A$
- $A^{-1}$  - matriz inversa de  $A$  única
- $A^d$  - matriz inversa de  $A$  à direita
- $A^e$  - matriz inversa de  $A$  à esquerda
- $A_s$  - matriz similar
- $A_u$  - matriz unitária
- $A_o$  - matriz ortogonal
- $A_{i-1}$  - matriz atitude de iteração anterior
- $A_{m \times n}$  - matriz de tamanho  $m \times n$
- $[a_{ij}]_{m \times n}$  - elemento de matriz  $m \times n$
- $[\bar{a}_{ij}]_{m \times n}$  - complexo do elemnto de matriz  $m \times n$
- $det$  - determinante de matriz
- $f(A)$  - função matriz
- $I$  - matriz identidade  $3 \times 3$ ;
- $M$  - matriz de rotação do referencial do sensor para o referencial do satélite
- $M_r$  - matriz representativa da base ortonormal inercial
- $M_s$  - matriz representativa da base ortonormal no referencial do satélite
- $M_r^T$  - matriz transposta de  $M_r$
- $M_i, U_i$  - matrizes definidas para o cálculo de  $\xi_i^*$
- $R_1$  - matriz de transformação de  $x''y''z''$  para  $x''''y''''z''''$

$R_2$  - matriz de transformação de  $x'y'z'$  para  $x''y''z''$

$R_3$  - matriz de transformação de  $xyz$  para  $x'y'z'$

$R_{313}$  - matriz de rotação de Euler em modo 313

$R_{321}$  - matriz de transformação de  $xyz$  para  $x''y''z''$  (em modo 321)

$tr$  - traço de matriz

$\Lambda$  - matriz diagonal

• vetores:

$\hat{e}_i$  - versor do sistema ortonormal de base  $\varepsilon'$

$\hat{e}_i, \hat{e}_j, \hat{e}_k$  - versores do sistema ortonormal

$\varepsilon_{ijk}$  - condição do sistema ortonormal

$\vec{dF}_i$  - força no elemento  $dm_i$

$\vec{I}$  - tensor momento de inércia

$\hat{i}, \hat{j}, \hat{k}$  - versores do sistema ortonormal

•  $dm_i$  - elemento de massa do satélite

$dN_i$  - torque aplicado ao centro geométrico pelo elemento  $dm_i$

$\lambda$  - autovalor

$\hat{\delta}$  - versor representativo da posição observada da estrela

$\rho$  - vetor posição do C.G. do satélite com relação ao seu centro geométrico

$\vec{r}$  - vetor qualquer

$\vec{r}_i$  - vetor posição de um elemento do satélite com relação ao seu centro geométrico

$\vec{r}_i'$  - vetor posição de um elemento do satélite com relação ao seu C.G.

$\vec{R}_i$  - vetor posição de um elemento do satélite com relação à Terra

$\vec{R}_s$  - vetor posição do centro geométrico do satélite com relação ao seu C.G.

$\hat{s}$  - versor representativo da posição catalogada da estrela

$\vec{V}$  - vetor de módulo  $V$

$\vec{V}_{n \times 1}$  - vetor ou matriz coluna

$V_x, V_y, V_z$  - Componentes de vetor no sistema triaxial cartesiano inercial

$\hat{V}_s$  - versor de posicionamento da estrela no sistema do sensor

$\hat{V}_i$  - versor de observação no referencial inercial

$W_x, W_y, W_z$  - componentes de vetor no sistema triaxial cartesiano no referencial do satélite.

$W_i$  - versor de observação no referencial do satélite

$u, v$  - coordenadas cartesianas planas na matriz do sensor

$\vec{x}^*$  - conjugado complexo do vetor  $x$

$x, y, z$  - componentes de vetor no sistema ortonormal

• funções:

$f_i(r)$  - função de probabilidades modificada por faixa de critérios

$f_{ótima}(r)$  - função ótima dos múltiplos critérios

$J$  - função erro

$P_0$  - probabilidade de identificação correta

$P_1$  - probabilidade de não identificação

$P_2$  - probabilidade de identificação ambígua

$P_3$  - probabilidade de identificação errônea

$P_i(r)$  - função de probabilidade de identificação

• erros, coeficientes, tolerâncias e fatores:

$Bom$  - % de critério para a qual a probabilidade é boa

$c_i, d_i$  - coeficientes dependentes do sensor no cálculo do sinal de saída

$\varepsilon$  - tolerância permitida

$h$  - fator de erro de probabilidades;

fator empírico no cálculo de magnitude instrumental.

$I$  - sinal de saída relacionado à magnitude instrumental

$\mu$  - fator de presença de estrelas no campo para as probabilidades

$\xi_i$  - erro da atitude calculado com relação à real

$\xi_i^*$  - erro que otimiza  $J$

$Ruim$  - % de critério para a qual a probabilidade é ruim

$\sigma_k$  - desvio padrão de medida (erro do sensor)

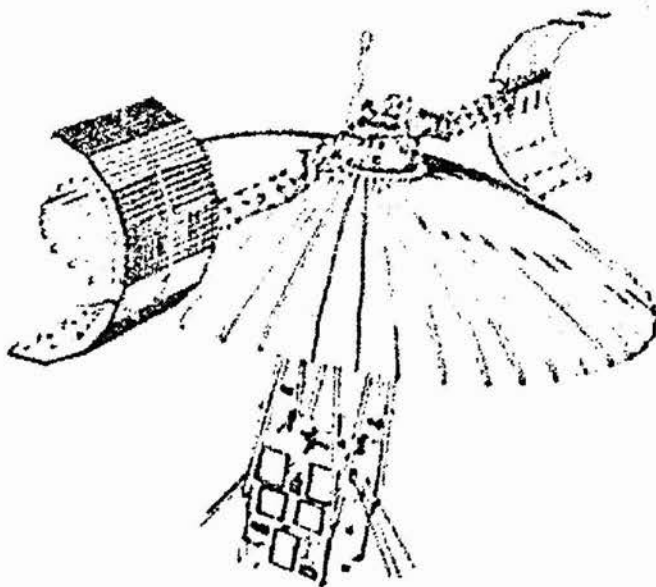
# Capítulo I

# 1. Introdução

O trabalho aqui desenvolvido, tem claros objetivos no estudo da determinação de atitude de um satélite em órbita, em torno do planeta Terra, e com estabilização triaxial através da identificação de estrelas por um sensor estelar.

A necessidade de um estudo a este nível é devida à importância de uma correta atitude. O correto posicionamento de um satélite é de extrema importância, uma vez que vários fatores estão intrinsecamente ligados à sua atitude, como a iluminação de painéis solares para geração de energia, interpretação de dados e reconhecimento através do posicionamento, minimização de forças e torques atuantes (que tendem a tirar o satélite de sua posição correta), satisfação de todos os requisitos exigidos pela missão a ser desenvolvida e o controle do mesmo.

Para a obtenção da atitude de um satélite, podemos utilizar diversas formas, como estabilização triaxial ou por rotação, cada qual adaptando-se ao tipo do mesmo e de sua missão (ver fig. 1 e 2).



**Fig. 1. - Satélite ISEE - 1 estabilizado triaxialmente.**

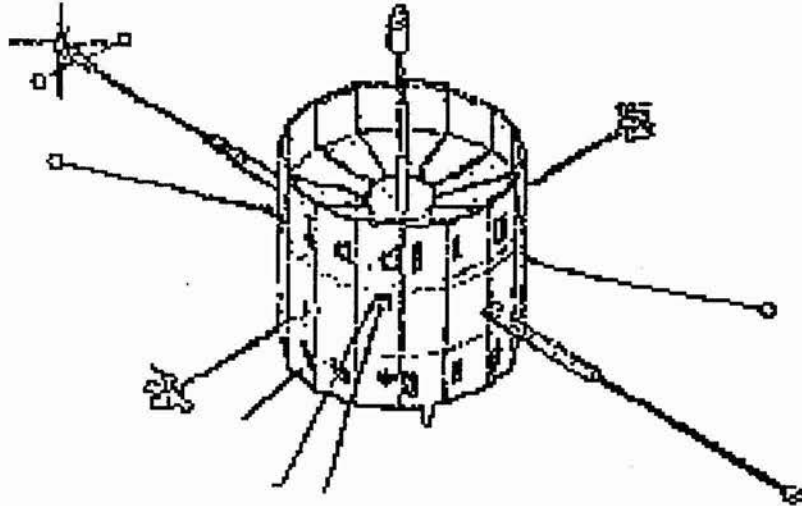


Fig. 2. - Sonda ATS - 6 estabilizada por rotação.

No presente caso, para o satélite estabilizado triaxialmente, a utilização do sensor estelar será trilhada e analisada com o desenvolvimento de um *software* de identificação de estrelas por alguns métodos pré-escolhidos segundo alguns critérios, que tentam balancear simplicidade e funcionalidade, tendo em vista o desenvolvimento de métodos mais complexos num segundo instante.

O critério de utilização de um sensor estelar baseia-se no fato de que estes possuem alta precisão e são muito versáteis, o que permite sua instalação a qualquer tipo de missão.

No entanto, para esse tipo há necessidade de uma atitude pré-estimada, que pode ser obtida por outros sensores ou sistemas menos precisos, tais como magnetômetros. Uma vez obtida a pré-estimativa, entrará no processo de cálculo para a avaliação da correta atitude. Neste trabalho, a atitude será gerada por um simulador e com esta, a atitude pré estimada pelo sensor de pré-estimativa, levando em conta seus erros, não sendo especificado o sistema que a gerou.

Num caso real (ver fig. 3), teríamos a informação chegando ao sensor e deste passando a todo o sistema onde seria calculada a atitude com o acréscimo da atitude de pré-estimativa.

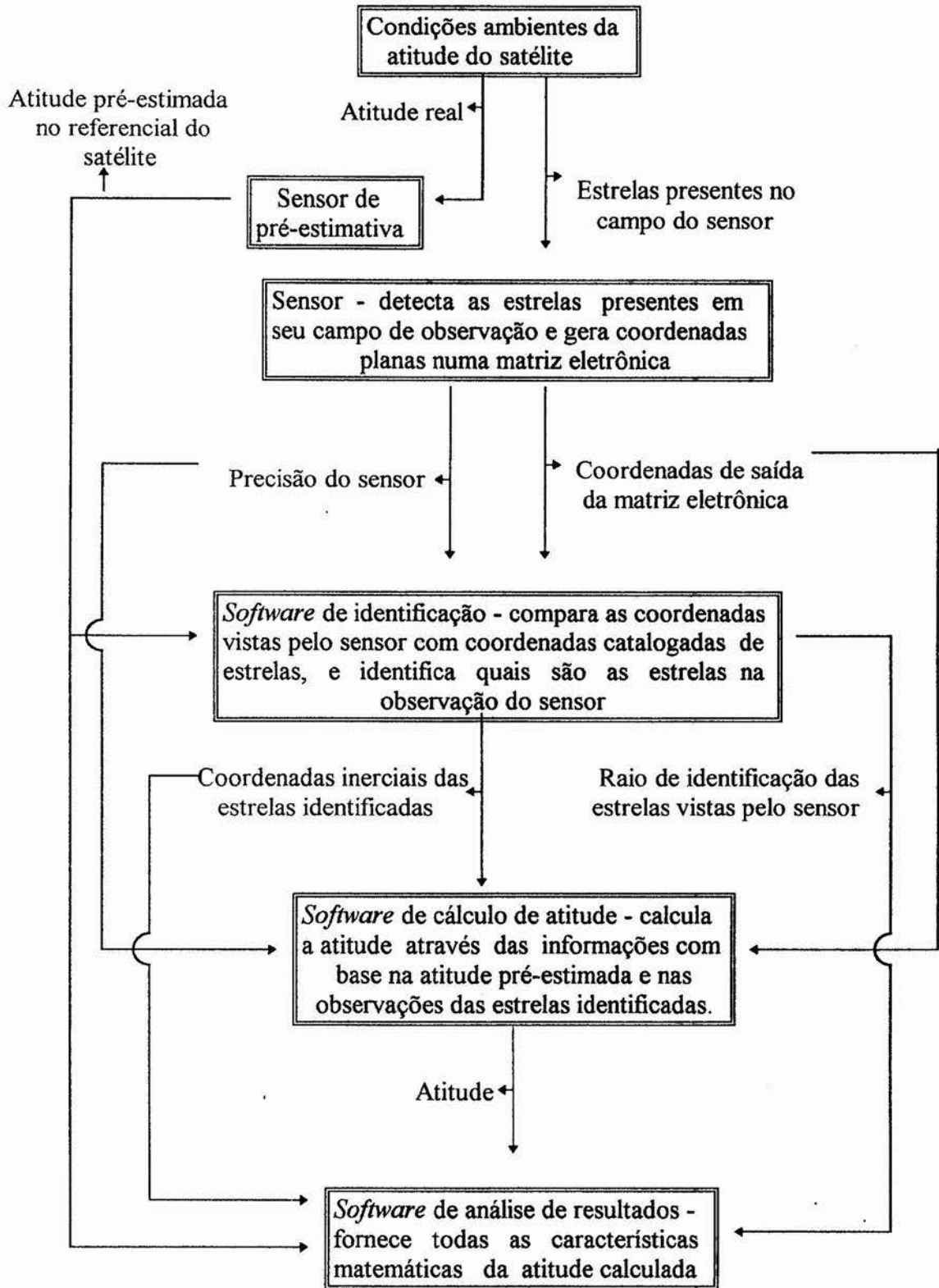


Fig. 3. - Sequência para o caso real de funcionamento do sistema de determinação de atitude.



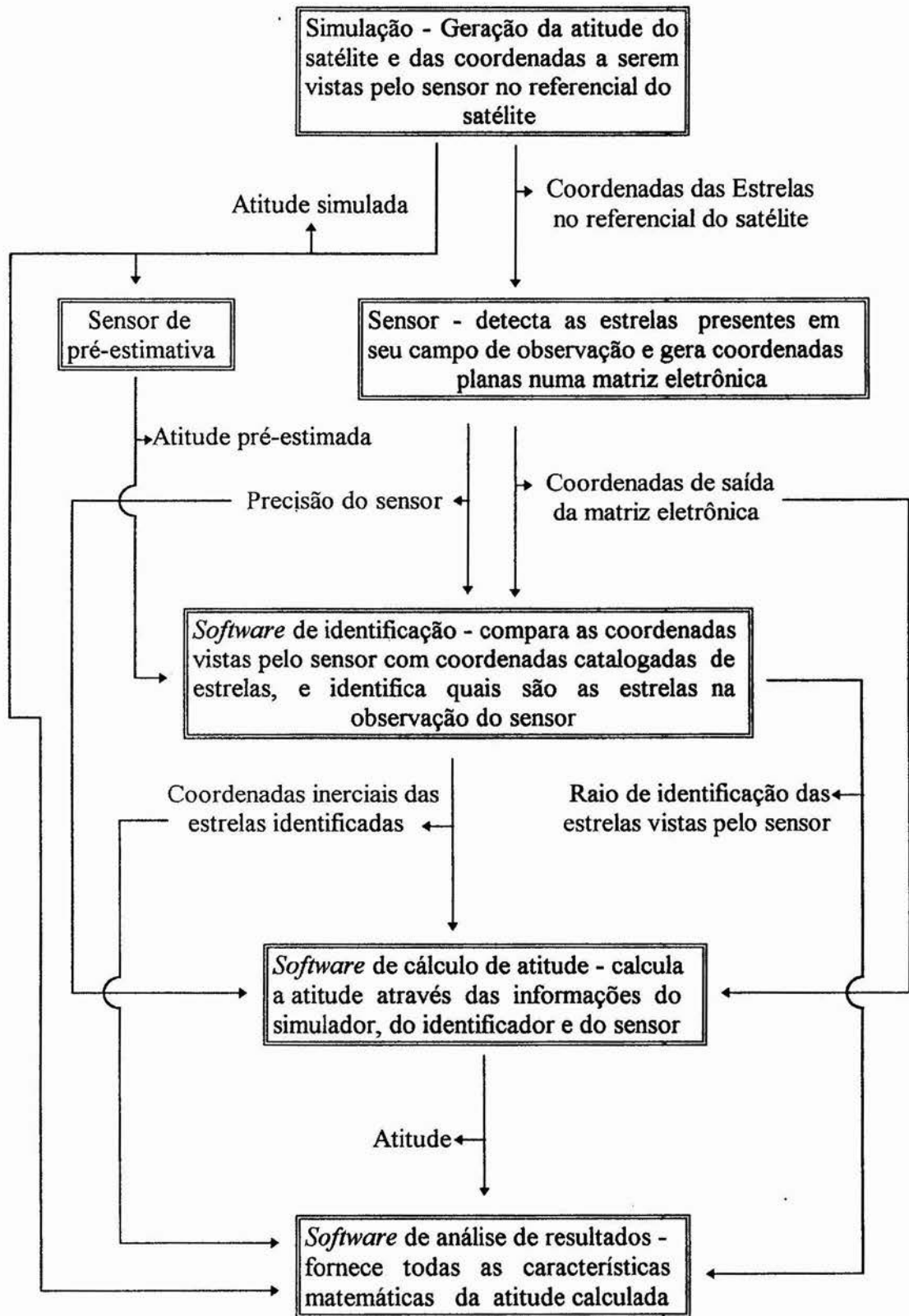


Fig. 4. - Seqüência para a simulação de funcionamento do sistema de determinação de atitude.

Para o desenvolvimento (ver fig. 4), as informações das estrelas a serem observadas e a atitude serão geradas por um simulador com participação interativa de um usuário, sendo calculada a atitude pré-estimada a partir da atitude real e dos erros dos sensores auxiliares. A informação das estrelas será passada a um *software* que representa o sensor e que gera toda a informação sobre as estrelas vistas a partir da matriz eletrônica do sensor.

Então, junto com a pré estimativa da atitude, as informações das estrelas vistas passam ao sistema de identificação. Na identificação são selecionadas as estrelas a serem realmente utilizadas no cálculo da atitude através da **técnica de comparação direta** [b9.] assistida pela **técnica dos múltiplos critérios** [b15.]. Atente-se para o fato de que neste desenvolvimento, utilizaram-se cinco regiões pré-determinadas do céu onde a densidade de estrelas com magnitude visual menor que cinco é relativamente boa. Num posterior desenvolvimento, a idéia é utilizar-se todo o céu, modificando desta forma algumas das partes da área de simulação e identificação do *software* desenvolvido.

Posteriormente as coordenadas das estrelas selecionadas são repassadas ao sistema de cálculo junto com as coordenadas vistas pela matriz do sensor. Nesta etapa, a atitude final é calculada pelo **método de determinação de atitude triaxial otimizada** [b1.], sendo posteriormente analisada e comparada com a pré estimada por um sistema de análise que então fornece todas as características desejadas sobre a atitude calculada.

Os sistemas de coordenadas utilizados são representados pelo sistema azimutal, vetorial-cartesiano e equatorial de referência, sendo utilizados em dois referenciais básicos, o inercial (relacionado com a esfera celeste centrada na Terra) e o do satélite (relacionado com uma esfera celeste centrada no próprio satélite). Como em termos astronômicos a distância que separa o satélite do planeta é ínfima, foi desconsiderada qualquer variação nas coordenadas das estrelas catalogadas com referência ao planeta, devido à mudança do referencial de observação, a menos das transformações angulares.

A representação de atitude neste trabalho é feita através de ângulos de **Euler** e matrizes de rotação. Os ângulos de **Euler** são utilizados em função da simplicidade de composição e reduzida dimensão de representação, o que reduz erros computacionais, apesar de possuir singularidades. No entanto, para o transporte de atitude é utilizada com maior eficiência a matriz de rotação.

A seguir serão expostos os conceitos básicos que permitem o desenvolvimento, os fundamentos teóricos que suportam o trabalho, estando em seguida o desenvolvimento propriamente dito do trabalho.

# Capítulo II

## 2. Conceitos básicos

Esta parte do desenvolvimento propõe-se a uma breve discussão a respeito de conceitos básicos, sem os quais seria impossível o perfeito andamento do trabalho.

Todos os conceitos a serem explicados, formam em conjunto a base que dita toda a forma e as características com que o sistema desenvolvido funciona.

### 2.1. Sistemas de referência

- Esfera Celeste:

Para o desenvolvimento de toda a base do posicionamento e referência do sistema, apresentamos agora o conceito de esfera celeste.

O conceito de esfera celeste teve suas origens na forma com que o homem um dia interpretou o céu.

Movido pelo sentimento de que a Terra encontrava-se no centro do universo, o homem começou a imaginar o céu que o rodeava como uma gigantesca esfera onde fixavam-se todos os astros, estando a Terra ocupando exatamente o seu centro. Essa esfera realizava um movimento rotativo em torno da Terra, carregando consigo todos os astros do céu (hoje sabemos que é devido à rotação do planeta, e não dos céus como se pensava).

Apesar de ser conceitualmente errônea, a idéia de esfera celeste, além de didática, é muito prática no que se refere ao posicionamento e referência, onde a distância dos astros em profundidade nada influencia. Imaginemos então, numa aplicação qualquer, que o nosso planeta esteja estático e que todo o céu acima de nós resume-se a uma esfera rotativa centrada em nosso planeta. Através deste conceito, projetamos todas as características presentes na Terra para esta esfera, como o eixo de rotação e pólos, sendo estes denominados eixo e pólos celestes (Ver fig. 5).

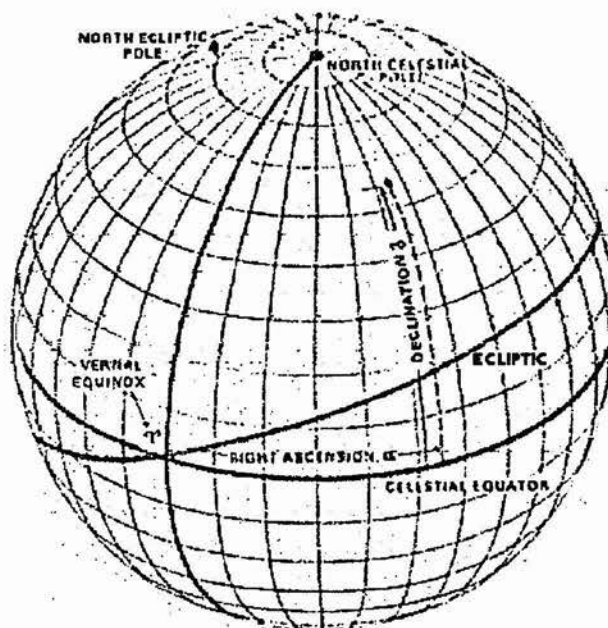


Fig. 5. - Esfera celeste inercial.

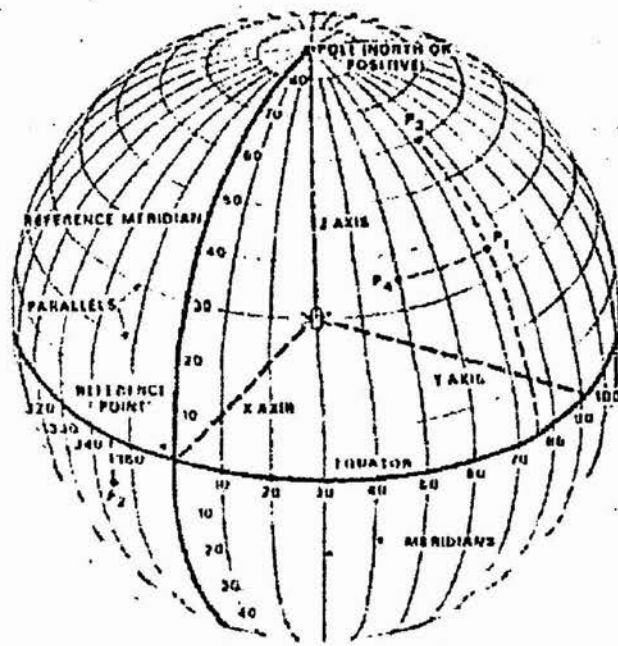


Fig. 6. - Esfera celeste do satélite.

Desprezamos neste trabalho, os movimentos próprios de todos os astros, imaginando que estejam estáticos eternamente. Além disso retiramos da esfera todos os planetas, cometas, asteróides, o Sol e a Lua, agindo como se somente houvessem estrelas.

Unindo a este conceito, acrescentamos sistemas que codificam coordenadas, facilitando a localização de qualquer astro no céu. Imaginamos circunferências envolvendo a esfera e concorrendo nos pólos, constituindo os meridianos (que comumente chamamos de círculos horários) e circunferências perpendiculares ao eixo de rotação que constituem os paralelos (Ver fig. 6 e 7).

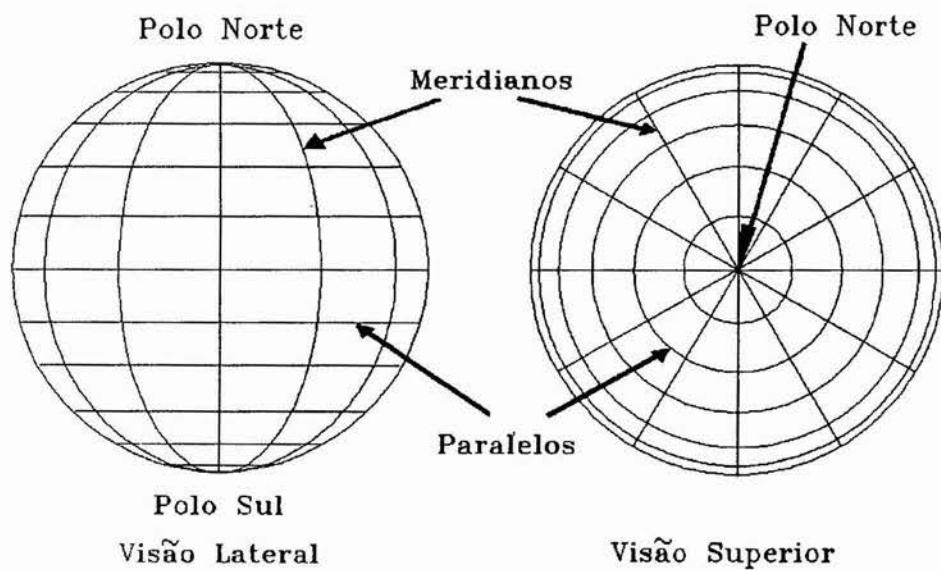


Fig. 7. - Codificação de coordenadas na esfera celeste

Tomando uma referência para os meridianos e paralelos, podemos agora ter um sistema que possibilite-nos qualquer posicionamento. A referência para os paralelos é adotada no paralelo que divide a esfera em dois hemisférios idênticos e é perpendicular ao eixo de rotação, sendo denominada equador. Os hemisférios levam os nomes dos pólos que contém, portanto sendo denominados hemisfério norte o que contém o pólo norte celeste que é uma projeção do pólo norte terrestre na esfera celeste.

A partir dos paralelos surge a coordenada de declinação que equivale ao ângulo medido entre dois segmentos com vértice no centro da esfera e apontando, um para um paralelo e o outro para o equador em cima de um mesmo meridiano, sendo medido a partir do equador positivamente de 0 a 90° para o norte e negativamente de 0 a -90° para o sul.

Já para os círculos horários, a referência adotada é tomada como sendo o meridiano que contém o ponto de interseção entre o plano do equador celeste e o plano orbital terrestre (chamado plano da eclíptica) que está na constelação de peixes, ao qual chamamos de ponto  $\gamma$ . Esta interseção deve-se ao fato de que a Terra não possui seu eixo perpendicular ao plano de órbita, fazendo com que seu eixo forme um ângulo com a normal ao plano de órbita. É importante ressaltar que este ponto de interseção não é estático na esfera celeste, pois caminha devido ao movimento de precessão da Terra, modificando lentamente então todas as coordenadas a ele associadas a cada vinte e seis mil anos, quando completa um ciclo em torno da esfera. Neste trabalho, estas mudanças foram desprezadas utilizando-se um catálogo de coordenadas de estrelas estático calculado pelo **Observatório Astronômico Nacional** com data de referência a 1950.

Todos esses fatores associados à Terra, são repassados logicamente à esfera celeste, tendo esta então a mesma inclinação do seu eixo de rotação com relação ao plano da eclíptica. No entanto, esse movimento de precessão só se aplica à esfera celeste inercial que é relacionada à Terra, não sendo aplicada à esfera celeste com referência ao satélite.

Dessa forma, a partir dos círculos horários surge a coordenada de ascensão reta que equivale ao ângulo medido entre dois seguimentos com vértice no centro da esfera e apontando, um para o um meridiano e o outro para o de referência em cima do equador, sendo medido em horas de 0 a 24 hs a partir do meridiano de referência no sentido de rotação da Terra, ou seja, no sentido de rotação do vetor de momento angular que aponta no sentido sul-norte.

- Descrição dos sistemas:

O sistema de coordenadas que utiliza a esfera celeste com coordenadas em ascensão reta e declinação é o chamado sistema equatorial de referência (Ver fig. 8).

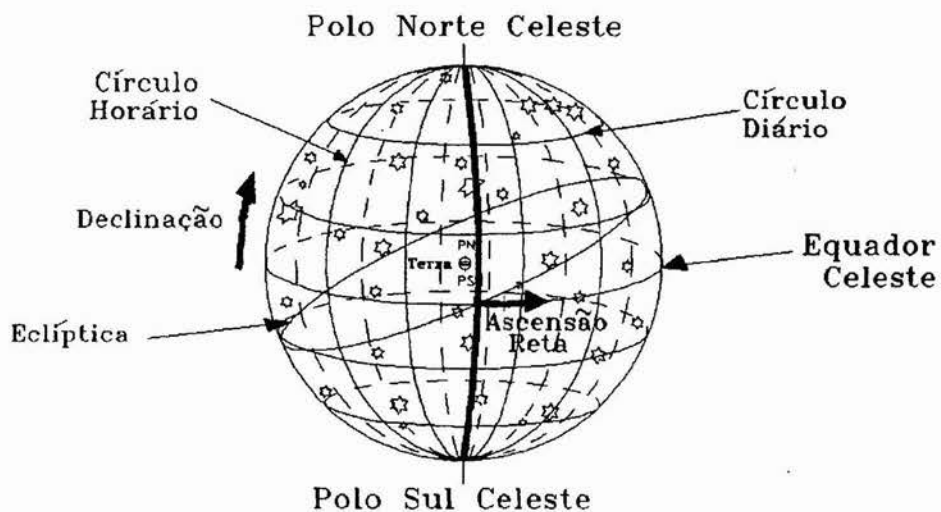
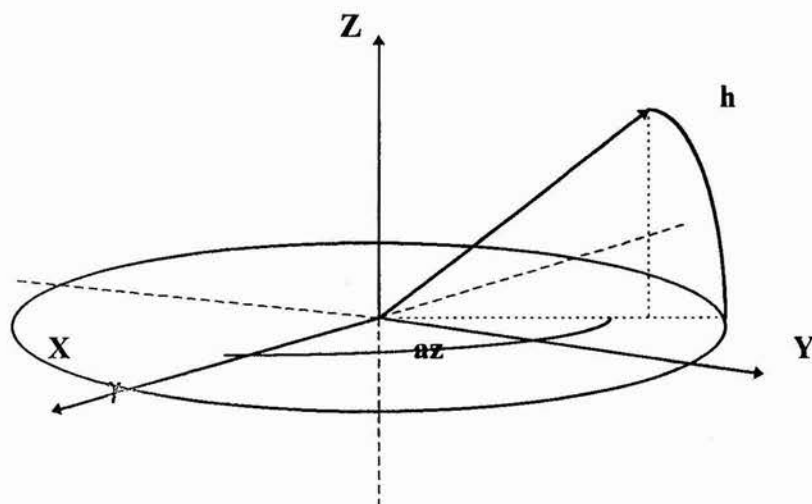


Fig. 8. - Sistema Equatorial de referência.

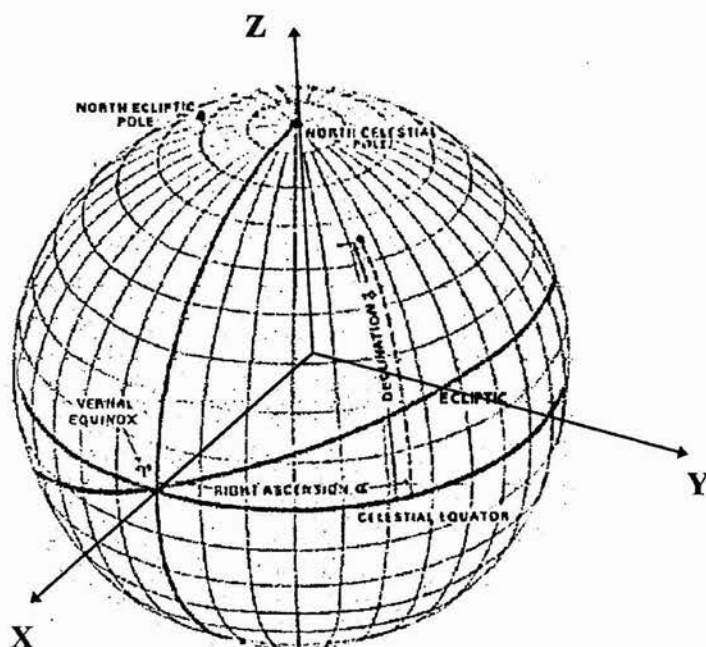


**Fig. 9. - Sistema Azimutal anti-horário.**

Além dele, ainda temos o sistema azimutal que constitui-se de duas coordenadas angulares, o azimute e a altura (Ver fig. 9).

A altura é exatamente idêntica à declinação, e o azimute é idêntico à ascensão reta medido de 0 a 360°, porém que no sentido contrário. No entanto, por comodidade de cálculo, definimos para o trabalho um sistema onde o azimute acompanhe o mesmo sentido da ascensão reta ao que chamaremos sistema azimutal anti-horário.

Apesar da praticidade em visualização, os dois sistemas angulares já descritos possuem alguns inconvenientes no que se refere aos cálculos com a utilização dos valores angulares e funções trigonométricas. Dessa forma, utilizamos então um terceiro sistema, o vetorial-cartesiano (Ver fig. 10).



**Fig. 10. - Sistema Vetorial-Cartesiano na esfera celeste.**

Este é definido basicamente por um sistema triaxial  $XYZ$  cartesiano com a origem no centro da esfera onde o eixo  $Z$  aponta para o pólo norte, e o eixo  $X$  aponta para o ponto  $\gamma$ . Esse sistema tem basicamente a função de ditar o posicionamento na esfera celeste, não tendo o módulo vetorial absolutamente função nenhuma. Assim ao adotarmos para o mesmo o módulo unitário, ou seja, uma esfera de raio unitário, trabalhamos com versores sem absolutamente nenhum prejuízo para o desenvolvimento.

Além de todos os sistemas de coordenadas utilizados para posicionamento, ainda há um quarto sistema somente utilizado no sensor e que representa a matriz eletrônica do sensor que é plana e será chamado sistema cartesiano plano. Esse sistema é definido por um par de eixos  $xy$  cartesianos e constituindo um plano perpendicular ao eixo  $X$  do sistema vetorial-cartesiano no referencial do satélite, sendo que  $y$  possui a mesma direção do eixo  $Z$  do sistema vetorial-cartesiano e  $x$  tem sentido oposto a  $Y$  também do sistema vetorial-cartesiano (Ver fig. 11).

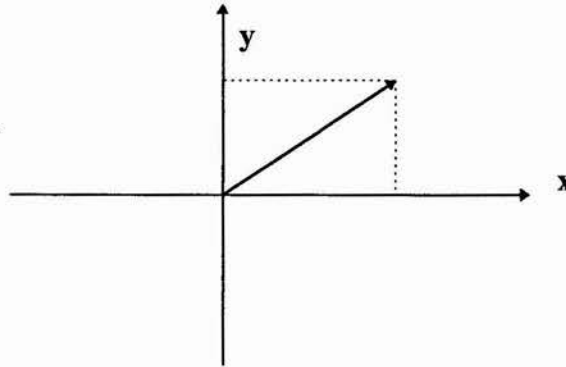


Fig. 11. - Sistema Cartesiano plano.

- Descrição dos referenciais:

Depois de definidos os sistemas de coordenadas utilizados, definiremos agora os referenciais aos quais eles são aplicados.

O primeiro referencial é o chamado referencial inercial e está estritamente ligado com as características físicas da órbita terrestre como já explicado. Para o caso adotaremos que estas são características estáticas eternamente. É nesse referencial que encontram-se catalogadas todas as coordenadas de estrelas utilizadas, que no caso foram retiradas das efemérides astronômicas de 1994 do Observatório Astronômico Nacional com cálculo base para 1950.

O segundo referencial é o do satélite, definido com base em eixos cartesianos  $XYZ$  definidos no satélite, com os quais é construída a esfera celeste para o referencial do satélite. São definidos pontos e características análogas na esfera celeste do satélite com relação à inercial.

Para definirmos os referenciais e suas relações trabalharemos com dois tipos de codificação de posicionamento, os chamados ângulos de **Euler** que definem rotações em torno de cada um dos eixos do sistema e a matriz de rotação que transforma as coordenadas de um vetor de um referencial para outro.

- Transformações:

Depois de definidos todos os sistemas e referenciais utilizados, passamos agora ao desenvolvimento matemático das transformações de coordenadas utilizadas entre os sistemas e referenciais.



a) Transformação do sistema equatorial de referência para o vetorial-cartesiano no mesmo referencial (Ver fig. 12):

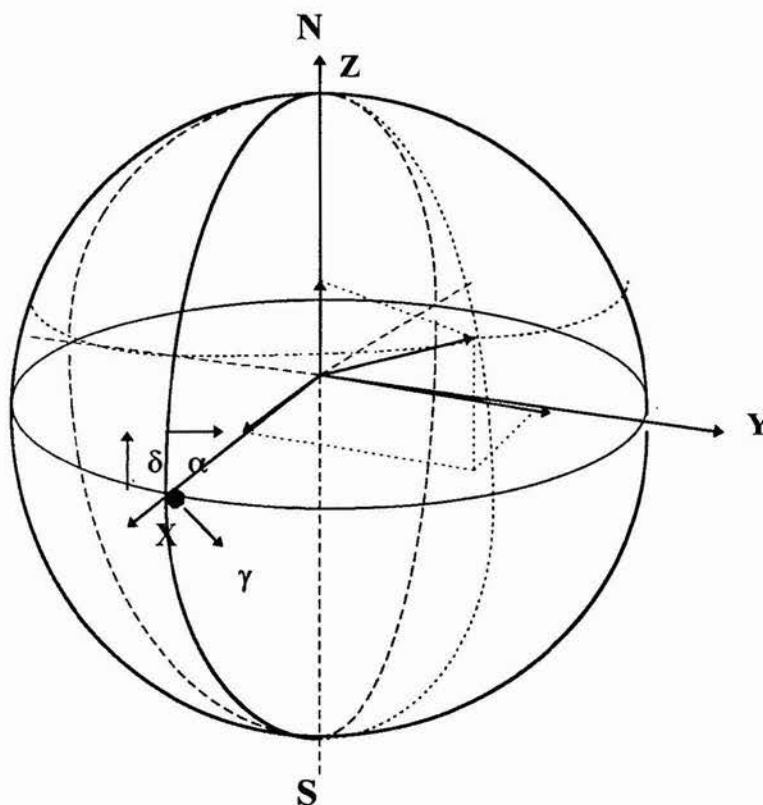


Fig. 12. - Sistema Vetorial-Cartesiano e Equatorial de referência.

Suponha que temos um astro com coordenadas no sistema equatorial de referência, sendo sua declinação  $\delta$  e sua ascensão reta  $\alpha_h$ . Queremos suas coordenadas no sistema vetorial-cartesiano e que terá três componentes  $V_x, V_y$  e  $V_z$ . Estamos passando as coordenadas de posicionamento de um sistema angular para um vetorial. Temos então através da figura as seguintes relações:

Como  $\delta$  está em graus e  $|\vec{V}| = 1$  :  $V_z = \text{sen } \delta$  ;

$\alpha_h$  está em horas, portanto sendo necessária a transformação para graus. Numa circunferência temos  $360^\circ$  ou 24 h, ou seja,  $15^\circ/\text{h}$ . Dessa forma, para termos  $\alpha_h$  em graus:

$$\alpha_g = \alpha_h \cdot 15.$$

Então:

$$V_x = \text{sen } \delta \cdot \cos \alpha_g, \quad V_y = \text{sen } \delta \cdot \text{sen } \alpha_g.$$

A transformação inversa daria-se da seguinte forma:

$$\delta = \text{arcsen}(V_z), \quad \alpha_g = \text{arccos}(V_x/V_z), \quad \alpha_h = \alpha_g/15.$$

b) Transformação do sistema equatorial de referência para o azimutal anti-horário no mesmo referencial:

Estamos agora passando de um sistema angular para outro muito semelhante.

A altura do sistema azimutal é idêntica à declinação do sistema equatorial de referência:

$$h = \delta.$$

Já o azimute é medido de 0 a 360° a partir do ponto  $\gamma$  em cima do equador. Portanto:

$$az = \alpha_h \cdot 15.$$

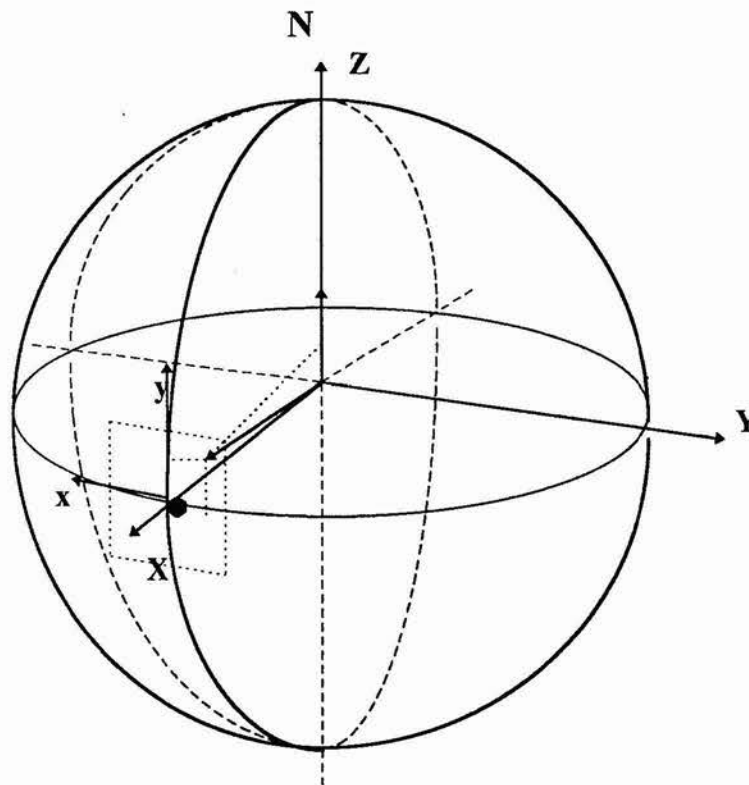
A transformação contrária daria-se da seguinte forma:

$$\delta = h, \alpha_h = az/15.$$

As demais transformações entre o sistema azimutal anti-horário, equatorial de referência e vetorial-cartesiano utilizam-se das apresentadas acima.

c) Transformação do sistema vetorial-cartesiano para o sistema cartesiano plano (Ver fig. 13).

Através da figura abaixo podemos obter as relações:



**Fig. 13. - Sistema Vetorial-Cartesiano e Cartesiano plano.**

Como  $|\vec{v}| = 1$ ,  $y = \text{sen } \delta$ . Mas  $V_z = \text{sen } \delta$ ,  $\therefore y = V_z$ .

$$x = -\text{cos } \delta \cdot \text{sen } \alpha_g = -V_y.$$

A transformação inversa seria da seguinte forma:

$$V_y = -x, V_z = y, V_x = 1.$$

Logo após a esta transformação, o vetor deve ser normalizado para que tenhamos módulo unitário. Usamos sempre o sistema Cartesiano plano referenciado no ponto  $\gamma$ , ou seja, o campo de visada do sensor sempre estará apontando para o ponto  $\gamma$  da esfera celeste com referencial do satélite.

Estas transformações tratam-se de aproximações de uma pequena parte da esfera por um plano ( $\alpha_g$  e  $\delta$  pequenos), logicamente contendo algumas imprecisões.

- Sistemas de representação de atitude e referencial:

Damos aqui uma breve descrição, sendo os detalhes tratados nos apêndices.

a) Ângulos de **Euler** (Ver fig. 14):

Neste trabalho usaremos os ângulos de **Euler** em rotação 3-2-1, isto é, rotações sucessivas em torno do eixo Z(3), em seguida Y(2) e por último X(1), todas no sentido positivo do vetor rotação.

Chamaremos a primeira rotação  $\varphi$ , a segunda de  $\theta$  e a terceira  $\psi$  como mostrado na seqüência de eixos abaixo:

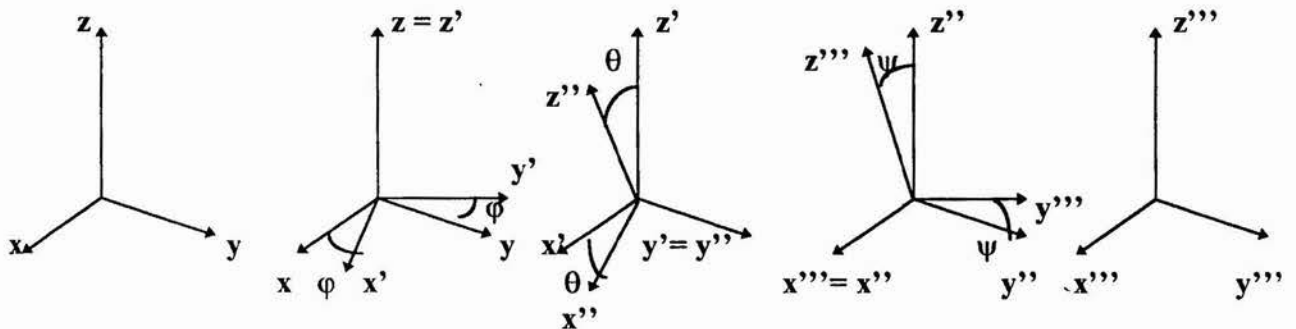


Fig. 14. - Rotações sucessivas em ângulos de *Euler*.

b) Matriz de rotação:

Seguindo a seqüência de eixos e fazendo as transformações, podemos chegar à matriz de rotação 3-2-1 que leva  $xyz$  para  $x'''$ ,  $y'''$  e  $z'''$ .

1. Primeira transformação - rotação  $\varphi$ :

$$\begin{aligned} x' &= x \cdot \cos \varphi + y \cdot \sin \varphi + z \cdot 0, \\ y' &= x \cdot (-\sin \varphi) + y \cdot \cos \varphi + z \cdot 0, \\ z' &= x \cdot 0 + y \cdot 0 + z. \end{aligned}$$

Em forma matricial:

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} \cos \varphi & \sin \varphi & 0 \\ -\sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

Chamaremos esta matriz de transformação de  $R_3$ .

2. Segunda transformação - rotação  $\theta$ :

$$\begin{aligned} x'' &= x' \cdot \cos \theta + y' \cdot 0 + z' \cdot (-\sin \theta), \\ y'' &= x' \cdot 0 + y' \cdot 1 + z' \cdot 0, \\ z'' &= x' \cdot \sin \theta + y' \cdot 0 + z' \cdot \cos \theta. \end{aligned}$$

Em forma matricial:

$$\begin{pmatrix} x'' \\ y'' \\ z'' \end{pmatrix} = \begin{pmatrix} \cos \theta & 0 & -\operatorname{sen} \theta \\ 0 & 1 & 0 \\ \operatorname{sen} \theta & 0 & \cos \theta \end{pmatrix} \cdot \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix}$$

Chamaremos esta matriz de transformação de  $R_2$ .

3. Terceira transformação - rotação  $\psi$ :

$$\begin{aligned} x''' &= x'' \cdot 1 + y'' \cdot 0 + z'' \cdot 0, \\ y''' &= x'' \cdot 0 + y'' \cdot \cos \psi + z'' \cdot \operatorname{sen} \psi, \\ z''' &= x'' \cdot 0 + y'' \cdot (-\operatorname{sen} \psi) + z'' \cdot \cos \psi. \end{aligned}$$

Em forma matricial:

$$\begin{pmatrix} x''' \\ y''' \\ z''' \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \psi & \operatorname{sen} \psi \\ 0 & -\operatorname{sen} \psi & \cos \psi \end{pmatrix} \cdot \begin{pmatrix} x'' \\ y'' \\ z'' \end{pmatrix}$$

Chamaremos esta matriz de transformação de  $R_1$ .

Enfim:

$$\begin{pmatrix} x''' \\ y''' \\ z''' \end{pmatrix} = R_1 \cdot \begin{pmatrix} x'' \\ y'' \\ z'' \end{pmatrix} = R_1 \cdot R_2 \cdot \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = R_1 \cdot R_2 \cdot R_3 \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} = R_{321} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

Portanto a matriz de rotação de  $xyz$  para  $x'''y'''z'''$  pode ser obtida da multiplicação matricial:  $R_{321} = R_1 \cdot R_2 \cdot R_3$ .

A título de simplicidade de notação, utilizaremos  $\cos \alpha = c\alpha$  e  $\operatorname{sen} \alpha = s\alpha$ .

$$\therefore R_{321} = \begin{pmatrix} c\varphi \cdot c\theta & c\theta \cdot s\varphi & -s\theta \\ c\varphi \cdot s\theta \cdot s\psi - s\varphi \cdot c\psi & s\varphi \cdot s\theta \cdot s\psi + c\varphi \cdot c\psi & c\theta \cdot s\psi \\ c\varphi \cdot s\theta \cdot c\psi + s\varphi \cdot s\psi & s\varphi \cdot s\theta \cdot c\psi - c\varphi \cdot s\psi & c\theta \cdot c\psi \end{pmatrix}$$

c) Transformação da matriz de rotação para ângulos de **Euler**:

Podemos chegar aos ângulos de **Euler** equivalentes utilizando alguns dos termos da matriz de rotação. Tomando  $R_{321}$ , chegamos à seguinte relação:

$$\begin{aligned} R_{321}(1,3) &= -\operatorname{sen} \theta \rightarrow \theta = \operatorname{arcsen} (-R_{321}(1,3)), \\ R_{321}(2,3) &= -\cos \theta \cdot \operatorname{sen} \psi \rightarrow \psi = \operatorname{arcsen} (R_{321}(2,3) / \cos \theta), \\ R_{321}(1,2) &= \cos \theta \cdot \operatorname{sen} \varphi \rightarrow \varphi = \operatorname{arcsen} (R_{321}(1,2) / \cos \theta). \end{aligned}$$

É conveniente computacionalmente, uma referência vetorial aos ângulos de **Euler** da seguinte forma:

$$\therefore \text{eulers} = (\varphi \ \theta \ \psi).$$

Ainda há uma última transformação importante no processo:

**d) Transformação de ângulos de Euler para o sistema equatorial de referência no referencial inercial:**

Nesta transformação, a rotação  $\varphi$  equivale ao movimento em ascensão reta:

$$\alpha_h = \varphi / 15.$$

O movimento em declinação é o oposto ao da rotação  $\theta$  por causa do sentido de giro:

$$\delta = -\theta.$$

O último giro em nada afeta a coordenada de apontamento, sendo assim denominado como fase  $\psi$ . Neste trabalho utilizaremos apenas o sentido de giro positivo para  $\varphi$ , transformando da seguinte forma quando dos cálculos resultar negativo:

$$\text{Se } \varphi < 0 \longrightarrow \varphi = \varphi + 360^\circ.$$

As transformações de coordenadas do referencial inercial para o satélite e vice-versa dão-se numa seqüência de transformações utilizando-se os métodos de transformações descritos.

Dessa forma, se temos as coordenadas  $\alpha_h$  e  $\delta$  quaisquer no referencial inercial, podemos transformá-las para coordenadas  $\alpha_{hs}$  e  $\delta_s$  no referencial do satélite da seguinte forma:

I - Transformação das coordenadas  $(\alpha_h, \delta)$  do sistema equatorial de referência para o sistema vetorial-cartesiano  $(V_x, V_y, V_z)$ ;

II - Transformação das coordenadas do sistema vetorial-cartesiano inercial  $(V_x, V_y, V_z)$  para coordenadas do sistema vetorial-cartesiano no satélite  $(W_x, W_y, W_z)$  através da matriz de rotação:  $W = R_{321} \cdot V$ ;

III - Transformação das coordenadas do sistema vetorial-cartesiano no satélite  $(W_x, W_y, W_z)$  para coordenadas  $(\alpha_{hs}, \delta)$  do sistema equatorial de referência no referencial do satélite.

## 2.2. Problema da determinação de atitude

Para a determinação de atitude, fixamos um sistema de eixos numa posição conhecida e orientamos dois ou mais versores do sistema para algumas referências, como por exemplo, o Sol, a Terra e as estrelas. Determinamos a atitude calculando o desvio do sistema com relação àquelas referências.

O processo da determinação de atitude de um satélite é muito dependente do tipo de estabilização do mesmo, por exemplo triaxial ou por rotação.

Para um satélite estabilizado por rotação, necessita-se da determinação da atitude de seu eixo de rotação. Esse processo resume-se, numa explicação simplificada, à determinação da intersecção de um cone cujo eixo liga o satélite ao Sol através de um sensor solar, com um outro cone determinado por um sensor de horizonte que liga o satélite ao centro da Terra (Ver fig. 15).

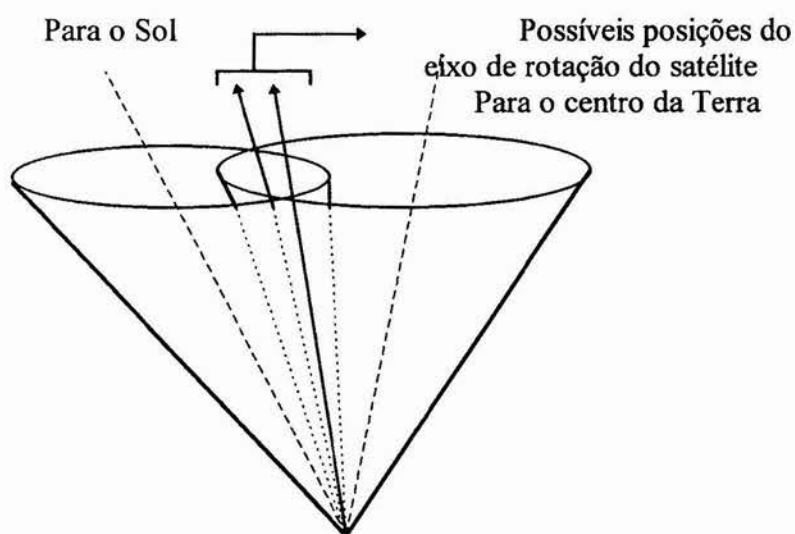
Para satélites estabilizados em três eixos utiliza-se um segundo método, o qual será estudado em detalhes neste trabalho, e aplicado para determinar a de atitude através de observações feitas por sensores de estrelas CCD.

## 2.3. Sensores típicos de atitude

Com base nas necessidades exigidas por cada sensor de cada tipo é que se desenvolve todo o método de funcionamento do mesmo.

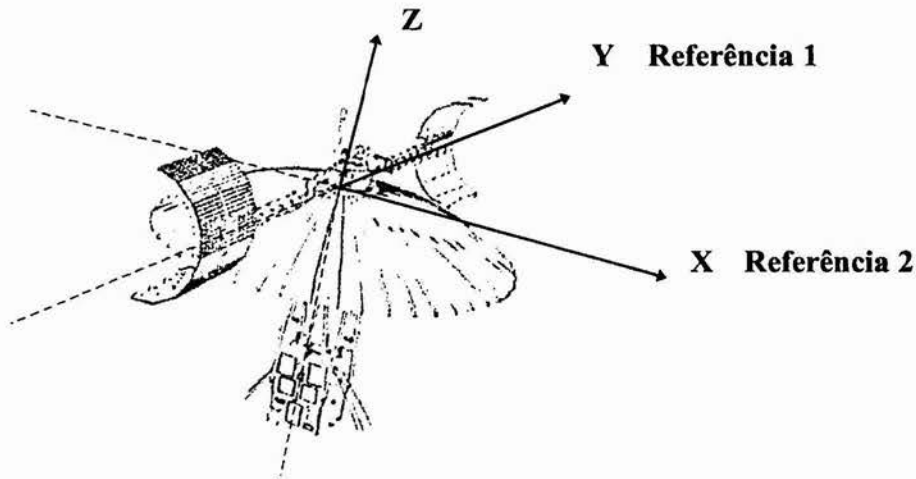
Os tipos mais comuns utilizados em missões atuais são os sensores solares, de horizonte (Terra) e de estrelas, além de giroscópios e magnetômetros.

Como já mencionado, os sensores solar e de horizonte funcionam com base na determinação de dois cones. Um, cujo eixo liga o satélite ao Sol e do qual as geratrizes são as prováveis posições do eixo de rotação ao que chamamos cone solar, e o outro, cujo eixo liga o satélite ao centro da Terra e do qual as geratrizes são as possíveis posições desse eixo de rotação. Fazendo a intersecção destes dois cones e tendo dados pressupostos, podemos determinar a posição do eixo. No entanto, teremos problemas caso os cones coincidam ou quase coincidam, ou ainda, não se interceptem, pois não teremos uma boa definição da intersecção, ou não teremos intersecção, sendo então necessária a aplicação de outros métodos que permitam o cálculo.



**Fig. 15. - Determinação de atitude de satélites rotativos.**

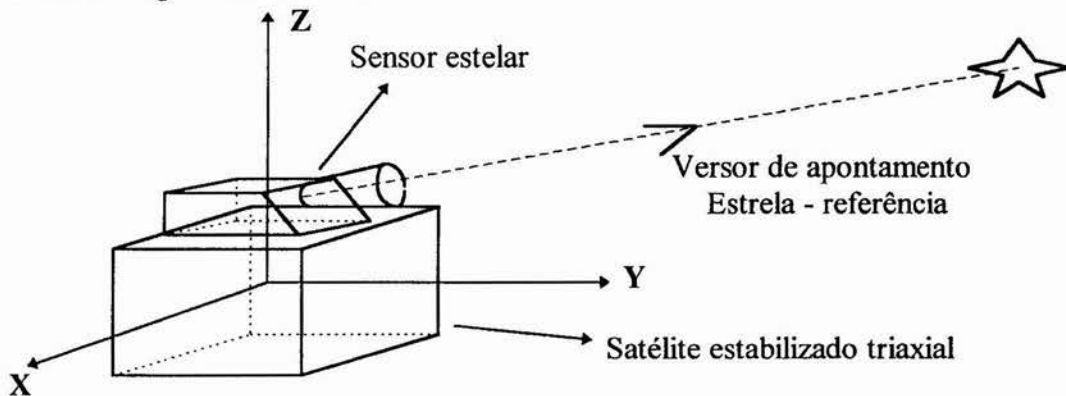
Como já mencionado na introdução, para um satélite estabilizado triaxialmente não podemos utilizar sensores com varredura já que não há a rotação necessária. Podemos utilizar sensores solares que determinem o vetor posição do Sol com relação ao satélite e magnetômetros que determinem o vetor campo magnético da Terra com relação a um campo gerado na satélite. Com estes dois vetores (Ver fig. 16) é possível a determinação da atitude do satélite, no entanto, se estiverem muito próximos, há uma indeterminação, não sendo possível o cálculo.



**Fig. 16. - Determinação de atitude de satélites triaxialmente estabilizados.**

Uma outra classe de sensores é a de sensores estelares que se divide entre três principais tipos: o de varredura (scanner) que utiliza a rotação do satélite (obviamente utilizados em satélites com estabilização rotativa); de movimentos mecânicos que possui movimento próprio para promover a varredura (utilizado em satélites triaxialmente estabilizados); e o de cabeça fixa, com matriz CCD que possui varredura eletrônica de um campo de visada fixo.

Os sensores estelares que não precisam de rotação para a leitura de suas referências, necessitam, no entanto, de uma pré-atitude determinada por qualquer outro tipo de sensor. Porém existem hoje métodos que tentam eliminar tal necessidade, fazendo com que o sistema seja capaz de se autodeterminar, ou seja, não há necessidade de pré-estimativa. No caso do sensor de estrelas, a cada estrela identificada associa-se um versor que liga o satélite à estrela (Ver fig. 17), sendo possível a determinação de diversos versores, o que intensifica a precisão do mesmo. No entanto, no caso da CCD, alguns erros também são introduzidos devido à constituição plana da matriz eletrônica que tenta representar uma porção esférica do céu, além de erros de origem foto-eletrônica.



**Fig. 17. - Determinação de versores de apontamento de estrelas a partir do satélite.**

Até aqui foram mencionados os tipos que constituem os sensores principais de determinação de atitude. Ainda existem os sensores inerciais que auxiliam a determinação de atitude. Estes encontram-se representados pelos giroscópios, além de outros que estimam versores sendo baseados na diferença angular a partir de uma referência (Ver fig. 18).

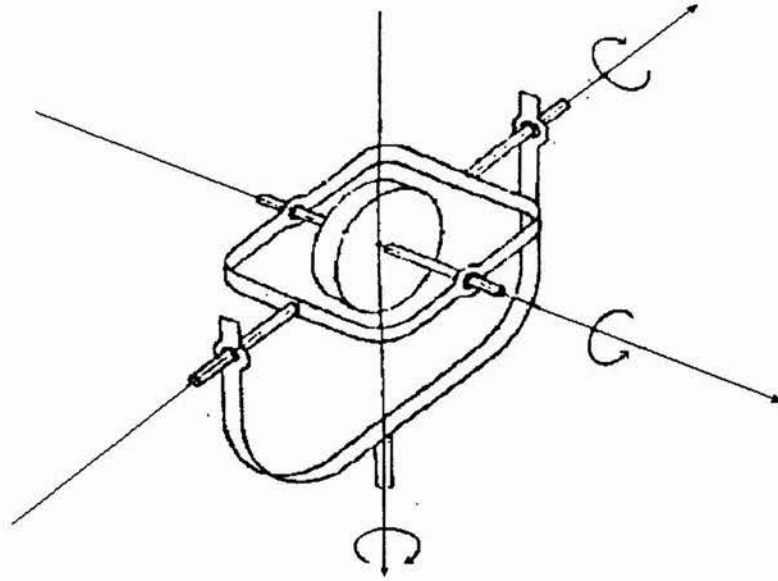


Fig. 18. - Giroscópios: sensores inerciais.

## 2.4. Sensor estelar

Nessa seção, faremos uma análise um pouco mais ampla dos sensores estelares e detalharemos o sensor alvo deste trabalho.

### 2.4.1. Sensores estelares

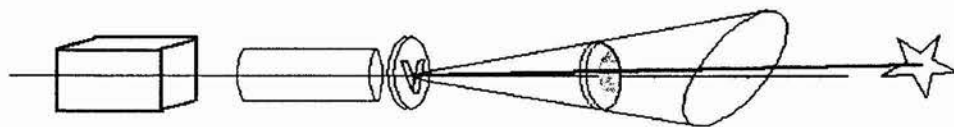
Os sensores estelares são capazes de medir a posição de estrelas com relação aos eixos do satélite e determinar a atitude com base em dados de posições das estrelas em catálogos estelares. São mais caros, pesados e requerem grande esforço computacional, pois os dados precisam ser analisados e calculados antes de calcular a manobra de atitude. Porém, apesar destas desvantagens, têm sido largamente utilizados em várias missões devido às suas resoluções da ordem de segundos de arco, propiciando grande precisão no posicionamento.

Como já mencionado, podemos dividi-los em três classes distintas: o sensor com varredura (scanner) utilizado em satélites rotativos, o de movimentos mecânicos para varredura e o de cabeça fixa com matriz CCD.

Todos eles são constituídos de partes em comum que são: o protetor solar gerador de sombra que minimiza efeitos indesejáveis da luminosidade solar; sistema óptico; equipamento de definição de imagem de escolha de campo de visada; detetor de imagem e eletrônica de processamento. Além destes, o sensor de movimentos mecânicos ainda possuem partes mecânicas e sistemas de auto posicionamento.

O maior problema com sensores estelares é a perda de luminosidade estelar, o que torna interessante o anteparo solar. Também são desenvolvidos sistemas de proteção à parte óptica que evitam a luminosidade procedente do Sol, da difusão em partículas de poeira, exaustão de gases e refletida pelo próprio satélite. Mesmo assim ainda são inoperantes com  $30^\circ$  a  $60^\circ$  de distância angular do Sol.





**Fig. 19. - Diagrama simplificado de um sensor estelar com disposição de fendas em "V".**

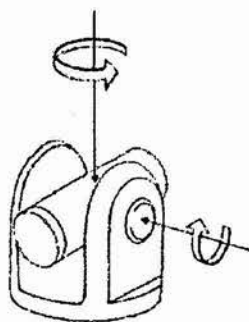
Os sensores de varredura são os mais simples pois não possuem partes móveis. São constituídos de equipamentos de definição de imagem com fendas em disposição "V" (Ver fig. 19).

Quando uma estrela passa pelo campo visual, sua imagem é projetada no plano focal, passando pela fenda e sensibilizando o detector. Se a amplificação for compatível com a sensibilidade da eletrônica, o detector gera um pulso elétrico, significando a presença da estrela para a eletrônica.

A varredura ocorre sobre a fenda que determina a posição da estrela com base numa referência e o intervalo de tempo entre a passagem da estrela pelas fendas e a fase de rotação do satélite.

A acuracidade depende da largura da fenda que é da ordem de 0.5' a 30' (minutos de arco). A varredura é feita conforme o movimento de rotação do satélite desenvolve-se por toda a esfera celeste. A dificuldade de interpretação aumenta conforme o movimento desvia-se do movimento de rotação simples, ou seja, sem precessões.

Para o caso dos satélites que operam com várias atitudes, o sensor com movimento mecânico é mais interessante, pois possui um campo de visão menor, mas pode varrer grandes áreas do céu (Ver fig. 20).

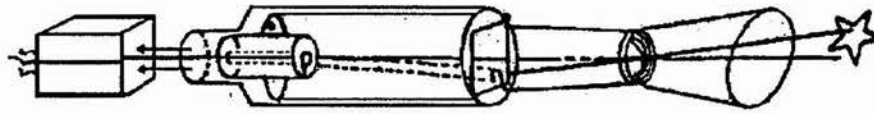


**Fig. 20. - Rastreador estelar de movimentos mecânicos.**

Diferentes tipos de definição de imagem são utilizados para determinar a posição da estrela com relação ao centro do campo. O sistema move o equipamento até que a estrela esteja no centro do campo e então a posição da mesma é dada de acordo com os ângulos de posicionamento do equipamento.

Os erros introduzidos na determinação da estrela afetam a acuracidade do sensor de movimento mecânico que vai de 1'' a 60'' (segundos de arco), excluindo o desalinhamento. A maior desvantagem está no movimento mecânico que diminui sua eficiência, além de torná-lo grande e pesado.

No caso dos sensores fixos que utilizam a eletrônica para promover a varredura, o uso de matrizes CCD's tem sido muito difundido. Elas são constituídas de uma matriz eletrônica foto-sensível que tem saída para um microprocessador que faz a identificação da posição da estrela através da posição de impressão na matriz (Ver fig. 21).



**Fig. 21. - Diagrama simplificado de um sensor fixo com varredura eletrônica.**

Neste caso o equipamento de definição de imagem consiste da matriz eletrônica que transforma o sinal óptico em elétrico através de foto-multiplicadores, passando à eletrônica onde é filtrado e processado.

O sensor fixo recebe a luz da estrela que passa através da óptica do equipamento e se projeta na matriz eletrônica disposta no plano focal. A escolha do campo de visada e da sensibilidade de magnitude depende dos requisitos de acuracidade. Um rastreador é sempre mais eficiente num campo de visada menor. No entanto, para pequenos campos é necessário que se aumente a sensibilidade afim de garantir que hajam estrelas no campo. E ainda, campos grandes precisam de grandes calibrações antes e depois do lançamento para correções devido às temperaturas e distorções. Porém o aumento de sensibilidade dá acesso a milhares de estrelas mais fracas, dificultando a identificação.

### **2.4.2. Sensor estelar alvo deste trabalho**

No caso deste trabalho, foi escolhido como modelo de teste o sensor CT-401 Rastreador Estelar de Cabeça Fixa.

Suas características constituem num campo de visada (janela do sensor por onde são observadas as estrelas) de  $8^{\circ} \times 8^{\circ}$  com precisão de mais ou menos  $3'$  de arco antes da calibração e  $10''$  de arco depois da calibração. É capaz de observar em curtos períodos de tempo, sendo por isso chamado de câmera. Possui quatro tipos de *threshold* selecionados conforme o brilho.

Pode ser utilizado tanto em aquisição, como em determinação de atitude. Funciona com o campo de visada instantâneo (suposta janela menor que se locomove pelo campo de visada) de  $9' \times 9'$  de arco varrendo o campo de visada continuamente até que alguma estrela sensibilize o equipamento.

Quando isso ocorre, inicia-se o rastreador que determina a posição da estrela no campo. O procedimento de rastreamento inicia-se com o movimento do campo de visada instantâneo dentro do campo de visada, quando então são detectadas algumas estrelas através dos pulsos elétricos por elas gerados numa matriz eletrônica foto-sensível, enviando-se informações ao *software* do sensor para que gere as informações de coordenadas em sua matriz. Quando a imagem cruza o pequeno campo de rastreamento, são gerados dois pulsos, um na entrada e outro na saída. Pela diferença de tempos entre eles é calculado o centro da imagem. Porém, o tempo não deve ser muito curto para não gerar erros. Conforme o campo move-se devido ao movimento de atitude, o rastreador segue e mantém a estrela centrada no campo de visada instantâneo. O rastreamento continua até que a estrela saia do campo de visada ou haja comando de parar rastreamento (Ver fig. 22). Para o caso deste trabalho, não foram considerados os movimentos de atitude, sendo então o rastreamento num campo de visada estático.

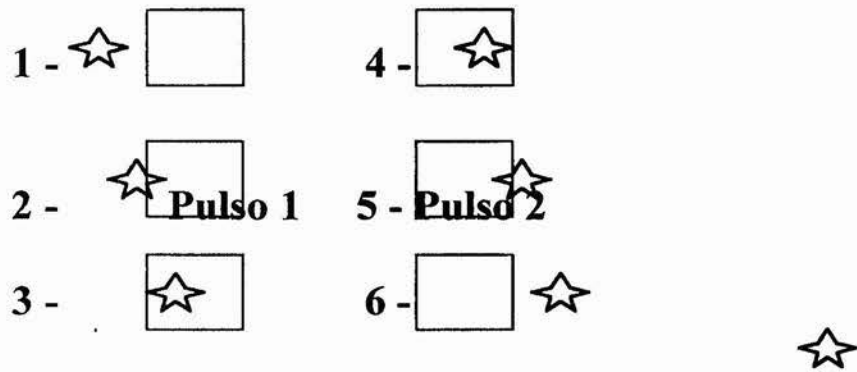


Fig. 22. - Seqüência de detecção do sensor rastreador estelar CT - 401.

Há um sistema de proteção que fecha a abertura quando algo muito luminoso atravessa o campo (isso ocorre quando o Sol localiza-se a  $42^\circ$  de distância angular do eixo do campo).

A procura pode ser realizada em vários estilos de operação. Se o rastreador tem que reinicializar a procura freqüentemente, é chamado de mapeador.

As principais fontes de erro no sensor listam-se abaixo:

- Flutuação de elétrons proporcional ao sinal;
- Ruído térmico e flutuações físicas nas células quanto ao tamanho e sensibilidade no espaço e no tempo.

O procedimento de determinação da posição da estrela com relação à matriz eletrônica segue os princípios abaixo:

- Cálculo da posição da estrela num campo  $3 \times 3$  células da matriz, onde a estrela é desfocada para que a informação seja distribuída e assim seja calculado o centróide da imagem (matriz = 100.000 células);

- Calibração com corrente de escuro (sinal presente mesmo em ausência de estrelas) a ser subtraído do sinal resultante;

- Resfriamento do sensor para a redução dos efeitos térmicos de distorção.

O modelamento matemático do sensor segue a medida de coordenadas proporcionalmente ao plano focal. A localização da direção é dada por ângulos referentes aos eixos do sensor, formados pela união das coordenadas do plano focal e o centro óptico da lente. O plano  $xy$  do campo de visada é paralelo ao plano  $ZY$  do referencial inercial, sendo  $\lambda$  medido no sentido positivo de  $y$  (ou  $Z$  do referencial inercial) e  $\phi$  no sentido positivo de  $x$  (ou negativo de  $Y$  do referencial inercial) (Ver fig. 23 e 24).

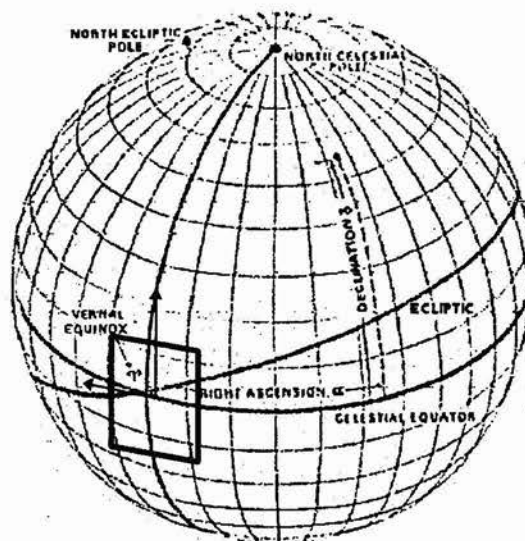


Fig. 23. - Esfera celeste e sistema de coordenadas locais na matriz eletrônica.

Porém, nas coordenadas da matriz eletrônica, os sinais invertem-se devido à influência da lente. Do vetor de apontamento da estrela no referencial do sensor, podemos obter as coordenadas planas na matriz CCD do sensor.

Dessa forma, o vetor de posicionamento da estrela no sistema do sensor é dado por:

$$\hat{V}_s = \begin{pmatrix} \cos \phi \cdot \cos \lambda \\ \sin \phi \cdot \cos \lambda \\ -\sin \lambda \end{pmatrix}$$

Onde  $\phi$  e  $\lambda$  são dados através das coordenadas da matriz e da distância focal da lente  $f$  através de:

$$\operatorname{tg} \phi = u/f, \quad \operatorname{tg} \lambda = (v/f) \cdot \cos \phi$$

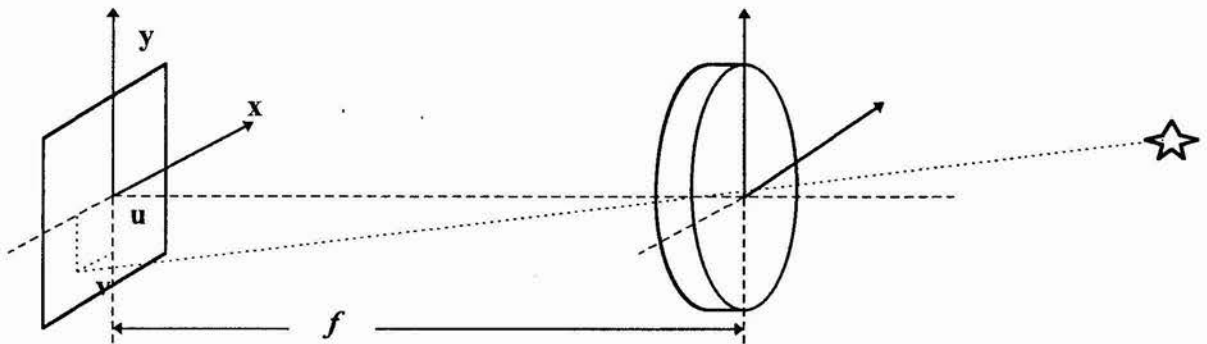


Fig. 24. - Relação de coordenadas no sistema óptico e matriz eletrônica.

No entanto, estas relações não são precisas devido às várias distorções de natureza óptica, eletro-eletrônica e térmica, surgindo então relações empíricas para o cálculo de  $\phi$  e  $\lambda$  a partir de  $u$  e  $v$ . Porém neste trabalho não foram levadas em consideração tais relações, sendo a imprecisão representada em apenas um número que reflete seus efeitos no posicionamento vetorial no método de determinação de atitude triaxial otimizada [b1.].

Dessa forma, para acharmos o vetor da estrela no referencial inercial temos:

$$\hat{V} = A \cdot M \cdot \hat{V}_s$$

Onde  $\hat{V}_s$  é dado no referencial do sensor,  $M$  é a matriz de rotação para o referencial do satélite e  $A$  é a matriz atitude.

Neste trabalho consideramos o sistema de eixos do sensor coincidente com o do satélite e dessa forma  $M$  é idêntica à identidade e pode ser omitida, obtendo-se apenas:

$$\hat{V} = A \cdot \hat{V}_s$$

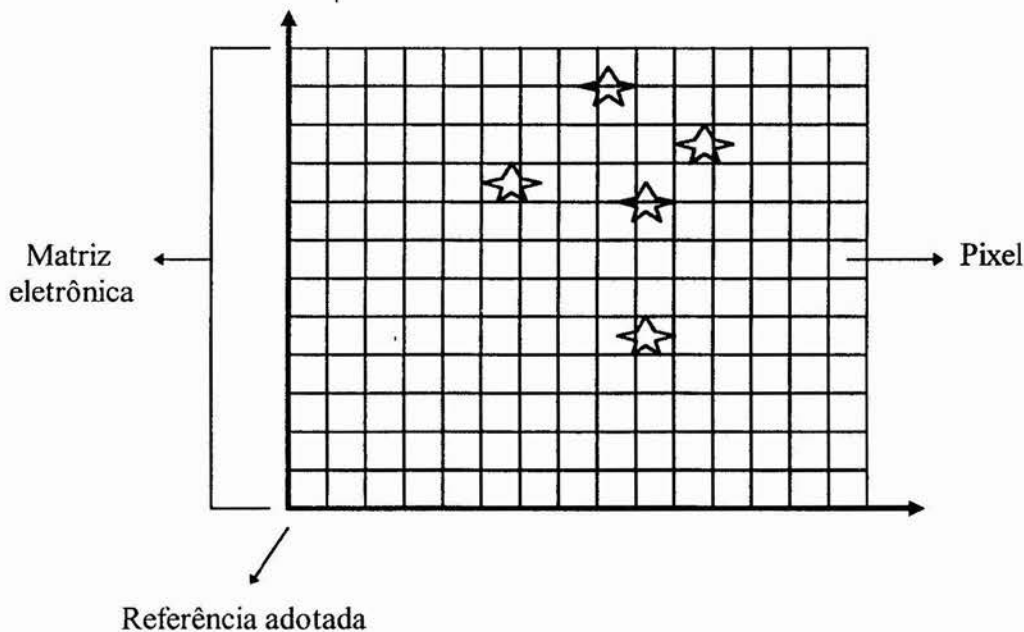
Um outro fator importante é a diferença de resposta em termos de magnitude. A magnitude percebida pelo instrumento é diferente e por isso é chamada de instrumental. Ela precisa ser calculada a partir das magnitudes ópticas das estrelas de forma a gerar um arquivo de estrelas não muito extenso, mas que inclua todas as que podem ser bem distinguidas. Até o momento, só foram citadas as magnitudes ópticas conforme apareceram no catálogo, não sendo até aqui considerados na identificação das estrelas.

Precisão do sensor:

Quando falamos em termos de precisão do sensor podemos distinguir dois tópicos diferentes, a precisão da matriz eletrônica em determinação do centro da estrela e a precisão quanto à sensibilidade de notar desvios quando há movimento por parte do satélite.

A precisão da matriz eletrônica está diretamente ligada a sua forma construtiva que define qual será a sensibilidade e tamanho de cada divisão sua que chamamos de pixel. Quanto menor os pixels da matriz, ou seja, quanto mais pixels por área, melhor para a definição de posicionamento de pontos de imagem, enquanto que sua sensibilidade dita a qualidade da imagem.

Os pixels são elementos foto-sensíveis que constituem toda a matriz e têm a eles associados uma posição em termos de coordenadas com relação a alguma referência na matriz (Ver fig. 25). É através destas coordenadas que são determinadas as coordenadas de uma estrela que impressione a matriz eletrônica. No entanto não é tão simples assim determinar as coordenadas da estrela impressionada com relação à matriz. A imagem recebida impressiona vários pixels ao mesmo tempo na região de impressão e em diferentes intensidades, causando um “borrão”. O cálculo do centro correto não é puro e simplesmente feito com o cálculo do centro geométrico da região, há que se levar em conta as intensidades de impressão, ficando desta forma determinado o centro por métodos estatísticos que determinam o “centróide” da região. É importante nesse momento levar em consideração o fato de que a sensibilidade de cada pixel não é idêntica à dos outros, sendo dessa forma necessária uma normalização da matriz com relação à sensibilidade. Tais métodos fogem ao escopo deste trabalho, sendo por isso não pesquisados.



**Fig. 25. - Matriz CCD do sensor de varredura eletrônica**

Já a precisão quanto à sensibilidade de notar desvios quando há movimento por parte do satélite depende diretamente da atitude do satélite, pois é influenciada pelos movimentos do mesmo com relação aos seus eixos de referência.

Os movimentos podem ser decompostos em dois movimentos componentes principais, rotação em torno do eixo do sensor e rotação perpendicular ao eixo do sensor (Ver fig. 26).

Percebe-se que a precisão de percepção de alteração de posição da imagem é melhor na direção perpendicular ao eixo do sensor, uma vez que o raio que vai do centro da referência de atitude do satélite ao sensor é maior que o raio que, no plano da matriz, vai do eixo do sensor a um ponto de imagem qualquer. Desta forma, para um mesmo ângulo de rotação, um ponto de imagem na matriz se locomove muito mais nos sentidos perpendiculares ao eixo do sensor que numa rotação em torno do mesmo, sendo então mais perceptível o seu movimento no sentido de rotação perpendicular ao eixo do sensor.

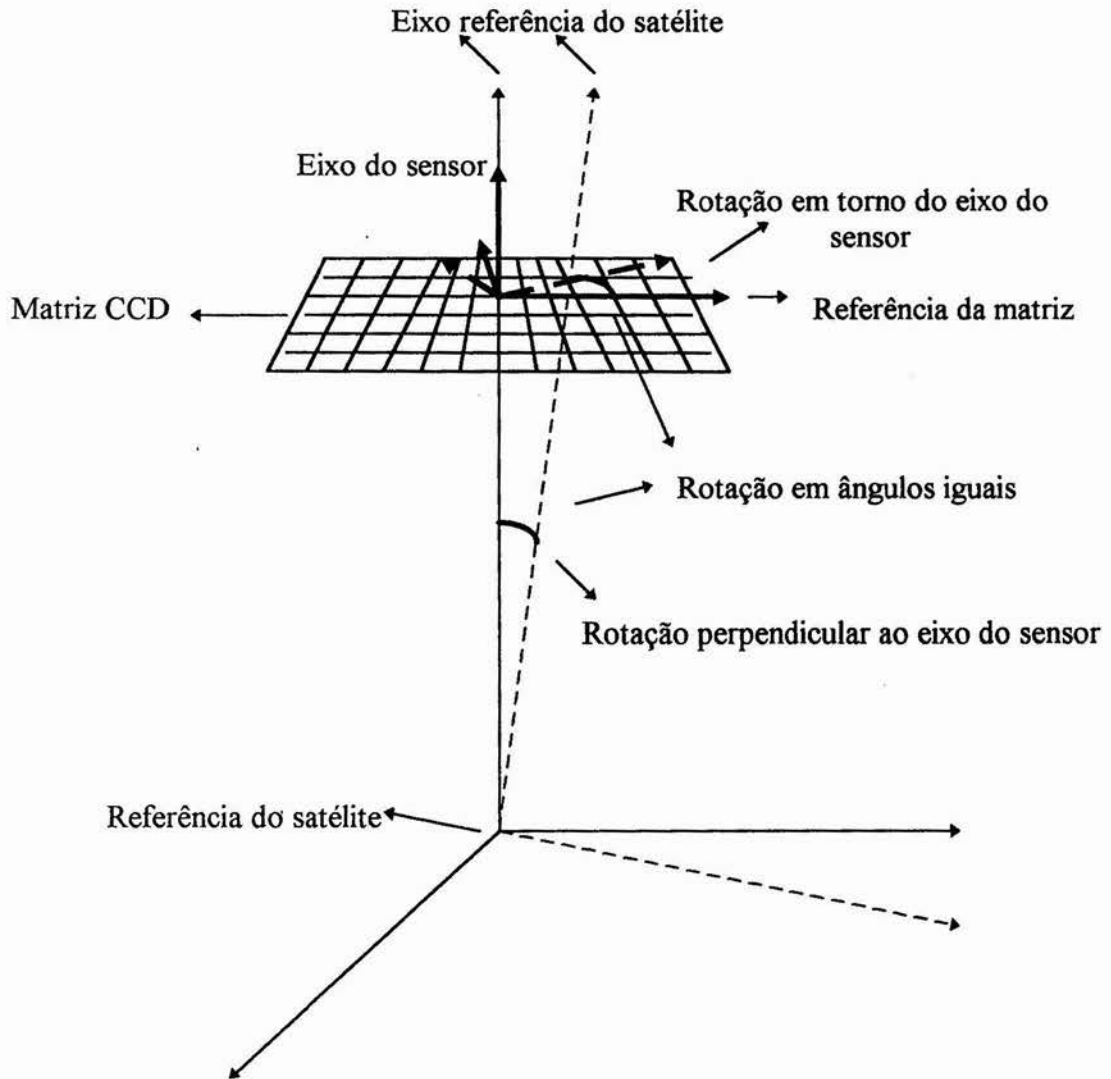


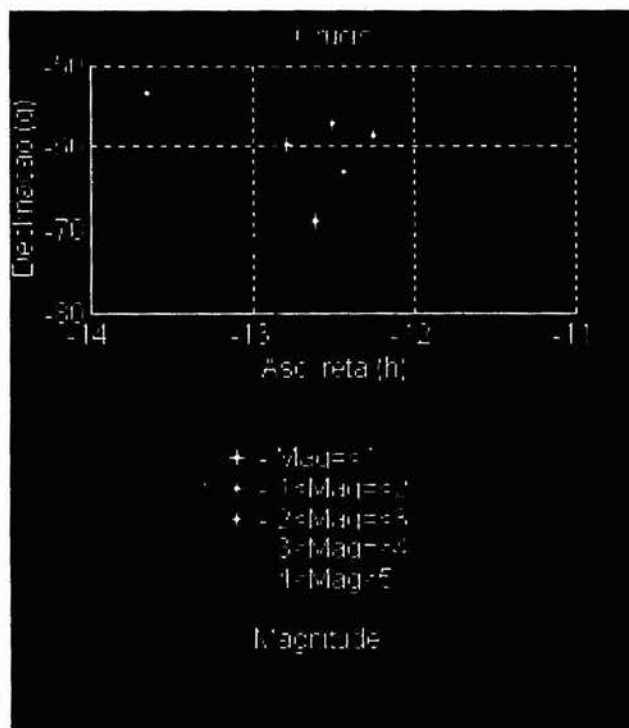
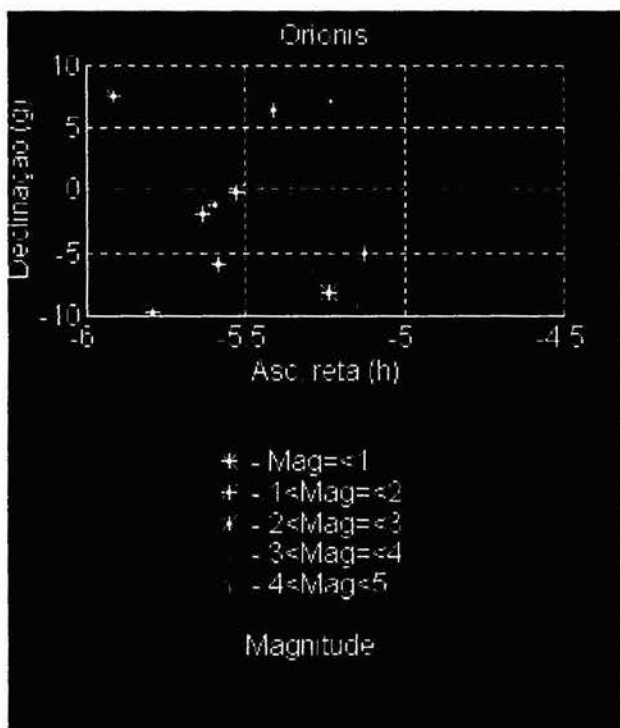
Fig. 26. - Movimentos em torno dos eixos do sensor e do satélite - Precisão do sensor

## 2.5. Mapa estelar

Ao tratarmos com as coordenadas de estrelas de um catálogo ou mapa de estrelas, devemos ter em mente que esse mapeamento é feito em uma esfera imaginária, e dessa forma, uma das melhores maneiras de representar tais coordenadas é a que envolve ângulos, sendo comumente utilizado o sistema equatorial de referências para a representação das coordenadas.

Po isso, quando representado em um plano, o mapa não corresponde realmente ao que deveria representar, havendo distorções inseridas pela planificação. No entanto, esse problema não acarreta dano algum ao desenvolvimento matemático, visto que todo este é feito vetorialmente, e ainda, a representação planificada apenas serve de meio de interação com o usuário do *software*.

Neste trabalho, todo o mapeamento da esfera celeste foi seguido a partir do catálogo de efemérides de 1994 com data base para 1950, do **Observatório Astronômico Nacional do Rio de Janeiro**, que utiliza estrelas de até 5ª magnitude. Por motivos que referem-se à natureza de testes do *software* desenvolvido, foram escolhidas cinco regiões do céu de  $20^\circ \times 20^\circ$  que abrangem as estrelas mais brilhantes do céu (Ver fig. 27).



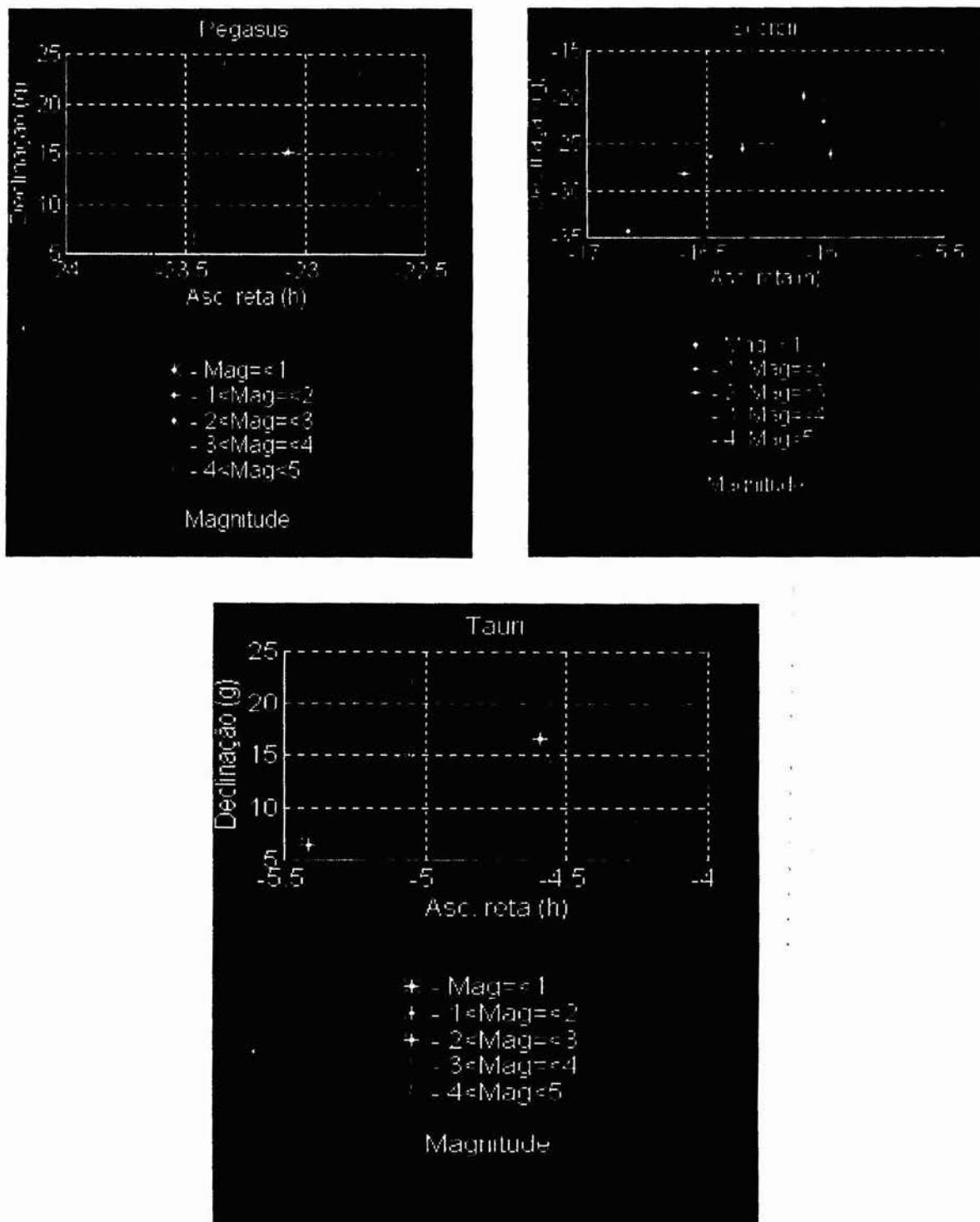


Fig. 27. - Regiões do céu escolhidas para os testes em Matlab.

Num trabalho integral, todo o catálogo seria utilizado abrangendo toda a esfera celeste, não havendo, no entanto, necessidade de tão grande abrangência numa fase inicial de testes.

No catálogo utilizado, as estrelas são catalogadas no sistema equatorial de referências em ascensões retas e declinações. São apresentadas em ordem crescente de ascensão reta e suas variações, fornecendo ainda a coordenada de declinação e variações e ainda a magnitude. No entanto, para o desenvolvimento foram tomados apenas os valores médios das coordenadas das estrelas que se encontram dentro das regiões estipuladas, assim como suas magnitudes. Nas representações acima nota-se a inversão de sinais para a ascensão reta, fato este que é devido ao sentido positivo de crescimento da ascensão ser contrário ao da representação cartesiana quando olhamos a esfera celeste.



O processo de escolha das regiões seguiu uma pré-observação das regiões que mais estrelas de baixa magnitude continham. Tomaram-se estrelas de referência a partir das quais delimitou-se regiões de  $20^\circ \times 20^\circ$  em direções paralelas à declinação e ascensão reta da estrela referência (Ver fig. 28 e 29). Tais estrelas são  $\alpha$ -crucis,  $\alpha$ -tauri,  $\alpha$ -scorpii,  $\delta$ -orionis e  $\alpha$ -pegasi. Com as coordenadas catalogadas destas estrelas referência foram constituídas cinco regiões com centro na estrela referência, estendendo-se  $10^\circ$  ao norte, ao sul, à esquerda e à direita.

Logicamente, a região acaba não sendo um quadrado de  $20^\circ \times 20^\circ$ , uma vez que fez-se seguir as linhas de meridiano delimitadas a partir da declinação da estrela. O que obtemos é algo aproximado a uma região de  $20^\circ \times 20^\circ$ .

O cálculo dos limites de declinação foi efetuado somando e subtraindo-se  $10^\circ$  da coordenada de declinação da estrela referência. No entanto, para o cálculo dos limites de ascensão reta, foi necessário uma transformação segundo a declinação, pois para declinações diferentes de zero, a distância angular no paralelo não corresponde à mesma distância angular no equador. Assim transformando-se  $10^\circ$  no equador para a correspondência no paralelo da estrela, foi somado e subtraído o número  $\Delta\alpha$  correspondente à coordenada de ascensão reta da estrela referência.



Fig. 28. - Posicionamento e determinação de regiões na esfera celeste.

As transformações foram as seguintes:

Como  $0 \leq \alpha \leq 24$  hs e uma circunferência tem  $360^\circ$ , a cada 1h em ascensão reta corresponde  $15^\circ$ . Assim:

$$\Delta\alpha = \frac{10^\circ}{15^\circ/\text{h} \cdot \cos \delta}$$

e para a declinação:  $\Delta\delta = 10^\circ$ .

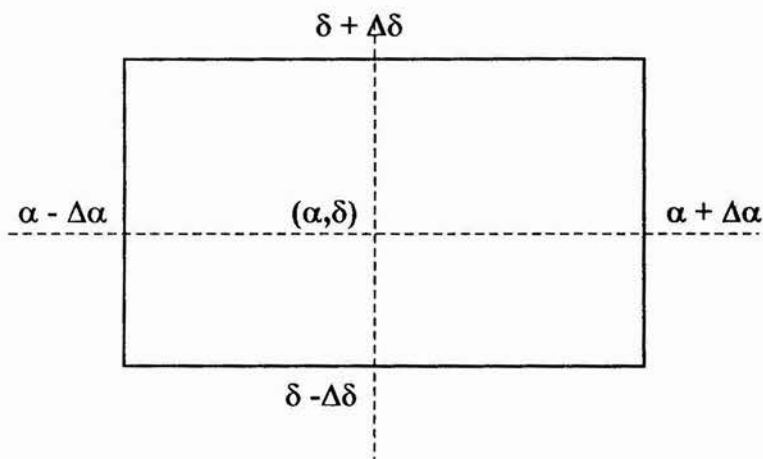


Fig. 29. - Constituição das regiões escolhidas.

Depois de obtida a região básica com seus limites, alguns dos limites foram aproximados para evitarem-se números quebrados, ficando dessa forma a estrela referência descentralizada na região.

Para a visualização, as estrelas foram separadas em classes de magnitudes ( $\text{Mag.} \leq 1$ ;  $1 < \text{Mag.} \leq 2$ ;  $2 < \text{Mag.} \leq 3$ ;  $3 < \text{Mag.} \leq 4$  e  $\text{Mag.} > 4$ ) e cada classe plotada numa cor diferente.

As regiões obtidas foram batizadas com os nomes das constelações a que pertencem suas estrelas referência, apesar de abrigarem estrelas de outras constelações.

Com as coordenadas e magnitudes das estrelas catalogadas em cada região e os respectivos limites, foram criadas matrizes para uso do *software*, representando as estipuladas regiões. Cada matriz leva o nome da região que representa e tem representada na primeira coluna a ascensão reta, na segunda a declinação e na terceira a magnitude. As estrelas são representadas a partir da terceira linha, estando a primeira e a segunda reservadas para os limites inferiores e superiores da ascensão reta e declinação da região respectivamente, sendo que a terceira coluna destas duas primeiras linhas nada representa, possuindo valores nulos. Todas as matrizes foram armazenadas num único arquivo representando o catálogo (Ver fig. 30).

crucis =			Nome da variável matriz
11.0000	-73.0000	0	→ Limites inferiores da região (o 0 não tem significado algum)
14.0000	-53.0000	0	→ Limites superiores da região(o 0 não tem significado algum)
11.1389	-58.9425	4.0000	
11.3456	-54.4581	4.2000	
11.5917	-62.9864	3.2000	
11.7553	-66.6953	3.7000	
11.7703	-61.1450	4.1000	
12.2469	-58.7153	3.0000	
12.4378	-63.0658	1.5000	
12.5139	-57.0797	1.5000	
12.5350	-72.0997	4.0000	
12.6136	-69.1025	2.8000	
12.7894	-59.6558	1.4000	
13.0308	-71.5167	3.5000	
13.3706	-60.9569	4.5000	
13.6583	-53.4358	2.5000	
13.9536	-63.6575	4.6000	

			→ Campo de magnitudes
			→ Campo de declinações
			→ Campo de ascensões

Fig. 30. - Matriz de representação de uma região do céu.

# Capítulo III

### 3. Fundamentos teóricos

Uma vez descritos e explicados os métodos e fundamentos básicos deste trabalho, passamos a descrever a forma como foram utilizados para que se obtivesse o resultado desejado.

#### 3.1. O algoritmo

- Determinação de atitude triaxial exata. [b1.]:

Este método aqui apresentado, apesar de desconsiderar erros e ser exato, é importante pois é utilizado num contexto mais amplo da determinação.

Se tivermos a medida de dois versores  $\hat{V}_1$  e  $\hat{V}_2$  no referencial inercial como duas referências para a determinação triaxial, devemos ter:

$$\hat{W}_1 = A \cdot \hat{V}_1, \quad \hat{W}_2 = A \cdot \hat{V}_2$$

Onde  $A$  é a matriz de rotação que leva as coordenadas inerciais para o referencial do satélite, sendo  $\hat{W}_1$  e  $\hat{W}_2$  os mesmos versores, mas representados no referencial do satélite (Ver fig. 31).

Se os dados forem perfeitos, isto é, sem erros, teremos:

$$\hat{W}_1 \cdot \hat{W}_2 = \hat{V}_1 \cdot \hat{V}_2$$

Dessa forma a determinação da atitude é exata e chamada determinística. Porém, se levarmos em conta os erros a solução não será única e deveremos aproximá-la.

Desconsiderando os erros, definimos duas bases ortonormais:

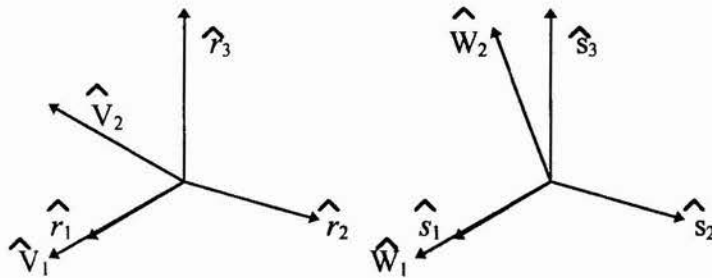


Fig. 31. - Bases ortonormais construídas através dos versores medidos.

$\{\hat{r}_1, \hat{r}_2, \hat{r}_3\}$  e  $\{\hat{s}_1, \hat{s}_2, \hat{s}_3\}$  a partir dos versores  $\hat{W}_1, \hat{W}_2, \hat{V}_1$  e  $\hat{V}_2$ .

Assim, podemos escrever:

$$\hat{r}_1 = \hat{V}_1, \quad \hat{r}_2 = \frac{\hat{V}_1 \times \hat{V}_2}{|\hat{V}_1 \times \hat{V}_2|}, \quad \hat{r}_3 = \hat{r}_1 \times \hat{r}_2$$

$$\hat{s}_1 = \hat{W}_1, \quad \hat{s}_2 = \frac{\hat{W}_1 \times \hat{W}_2}{|\hat{W}_1 \times \hat{W}_2|}, \quad \hat{s}_3 = \hat{s}_1 \times \hat{s}_2$$

Então teremos:

$$\hat{s}_i = A \cdot \hat{r}_i$$

Podemos definir matrizes:

$$\begin{array}{l} \text{matriz de } \hat{r} \longrightarrow M_r \equiv [\hat{r}_1, \hat{r}_2, \hat{r}_3] \\ \text{matriz de } \hat{s} \longrightarrow M_s \equiv [\hat{s}_1, \hat{s}_2, \hat{s}_3] \end{array}$$

Essas matrizes são ortogonais.

Dessa forma:

$$M_s = A \cdot M_r \longrightarrow A = M_s \cdot M_r^T$$

Sendo A a matriz de rotação ou atitude desejada.

No entanto, este algoritmo não é otimizado devido aos erros que não permitem a correta construção das bases ortogonais que permitem representar corretamente os versores em ambos os referenciais.

- Determinação de atitude triaxial otimizada. [b1.]:

Para o caso de considerarmos os erros nas medidas, podemos achar a matriz A que minimize o erro através da função erro:

$$J(A) = \frac{1}{2} \cdot \sum_{K=1}^N \cdot \frac{1}{\sigma_K^2} \cdot |\hat{W}_K - A \cdot \hat{V}_K|^2 \quad \text{para N versores.}$$

Onde os  $\sigma_K$  são os desvios padrões das medidas.

Podemos aproximar A por iteração através de:  $A \cong (I + [[\xi_i]]) \cdot A_{i-1}$

Onde I é a matriz identidade,  $[[\xi_i]]$  é a matriz da função ômega do erro  $\xi_i$  apresentada no apêndice e  $A_{i-1}$  é a matriz da iteração anterior.

Utilizando essa expressão para A na função J(A), obtemos a função J( $\xi_i$ ) e dessa forma o problema passa a ser achar  $\xi_i$  que minimize J( $\xi_i$ ).

Dessa forma, derivando J( $\xi_i$ ) com relação a  $\xi_i$  e igualando a zero, podemos achar  $\xi_i$  para o mínimo da função J( $\xi_i$ ), ou seja, achamos  $\xi_i^*$  para o qual  $\partial J(\xi_i)/\partial \xi_i = 0$ . Assim, com  $\xi_i^*$  voltamos à equação de A e achamos a matriz de rotação.

No entanto, para obter  $\xi_i^*$  devemos utilizar a equação resultante da condição de mínimo de J( $\xi_i$ ):

$$\xi_i^* = M_i^{-1} \cdot U_i$$

Onde  $M_i$  e  $U_i$  são dadas pelas equações abaixo, em função de iterações para os valores de A:

$$M_i = \sum_{K=1}^N \cdot \frac{1}{\sigma_K^2} \cdot [[A_{i-1} \cdot \hat{V}_K]]^T \cdot [[A_{i-1} \cdot \hat{V}_K]]$$

$$U_i = \sum_{K=1}^N \cdot \frac{1}{\sigma_K^2} \cdot [[A_{i-1} \cdot \hat{V}_K]] \cdot [[\hat{W}_K - A_{i-1} \cdot \hat{V}_K]]$$

Portanto o algoritmo inicia-se com um chute inicial que é dado pelo algoritmo determinístico.

A seqüência do algoritmo desenvolve-se da seguinte forma:

I - O algoritmo determinístico dá o chute inicial para A, a partir dos versores  $\hat{V}_i$  e  $\hat{W}_i$ .

II - Com  $A_1$  obtém-se  $M_1$  e  $U_1$  através das equações.

III - Com  $M_1$  e  $U_1$  obtém-se  $\xi_1^*$ .

IV - Com  $\xi_1^*$  e  $A_1$ , achamos  $A_2$  através da equação de A.

V - Com  $A_2$  prossegue-se até que  $\xi_i^*$  esteja muito próximo de  $\xi_{i-1}^*$ , sendo dessa forma a matriz  $A = A_i$ .

## 3.2. Métodos clássicos de identificação de estrelas

A identificação refere-se à associação de estrelas observadas pelos sensores, com estrelas catalogadas. Como as estrelas observadas tem coordenadas referenciadas ao sistema inercial, suas posições podem ser computadas, fornecendo a atitude do satélite.

O processo inicia-se com a transformação de coordenadas para o sistema inercial. Se a estimativa inicial de atitude é pobre, a posição transformada das coordenadas observadas pode estar distante angularmente da posição real, o que dificulta ou torna até impossível a identificação.

Serão brevemente citadas três técnicas, das quais uma será o tema da identificação de estrelas neste trabalho.

A técnica de comparação direta [b9.] é muito utilizada com processadores a bordo que comparam cada observação com o catálogo numa certa tolerância. Isso requer que a estimativa esteja bem perto da realidade. A técnica de separação angular compara as distâncias angulares com as do catálogo e é utilizada quando a observação é somente um pouco distorcida, mas a acuracidade da estimativa da atitude não permite o uso da comparação direta; e a técnica de fase é utilizada para satélites rotativos.

A distorção causa a necessidade de grandes tolerâncias e desta forma causa uma necessidade de uma boa quantidade exparsa de estrelas no campo. Para minimizar o problema pode-se reduzir o tempo de análise. Quanto mais divisões de coordenadas (número de pixels por unidade de área) a matriz pode representar melhor, pois pode representar mais facilmente a posição observada.

Normalmente, as coordenadas são transformadas para o sistema inercial, porém pode ocorrer o contrário.

Outro fator muito usado é a atribuição de notas a cada identificação, o que leva a um meio de comparação entre elas e uma possível escolha da melhor.

Comparações multidimensionais podem ser feitas se o sensor for capaz de medir outra variável além da posição, como por exemplo a magnitude, aumentando bastante a acuracidade.

Como este trabalho visa um procedimento simplificado, foi utilizada a técnica de comparação direta [b9.] sem a perda de nível de qualidade. Esta técnica compara cada observação com o catálogo, o que significa a necessidade de campo de visada pequeno para que não hajam tantas estrelas. Com isso a acuracidade da estimativa inicial de atitude deve ser razoavelmente alto.

A condição de comparação é tal que:  $d(\hat{o}, \hat{s}) < \epsilon$ . Ou seja, a distância da posição observada  $\hat{o}$  e da posição catalogada  $\hat{s}$  deve ser menor que uma tolerância  $\epsilon$ .

O processo desenvolve-se da seguinte forma: depois de transformar as coordenadas da observação para o sistema inercial, é construída uma circunferência com centro na observação e raio  $\epsilon$  (Ver fig. 32), sendo então verificadas quais estrelas posicionam-se dentro da circunferência. Depois da checagem, existem três possibilidades de identificação; nenhuma identificação com nenhuma estrela na circunferência; identificação ambígua com duas ou mais estrelas presentes ou identificação única com apenas uma estrela presente. O desejo é que haja uma identificação única, porém ainda pode ocorrer identificação errônea, isto é, a estrela

presente não é a estrela correta. Se a identificação está fora da circunferência ou não consta no catálogo, temos também identificação errônea, que pode causar divergência em algumas técnicas.

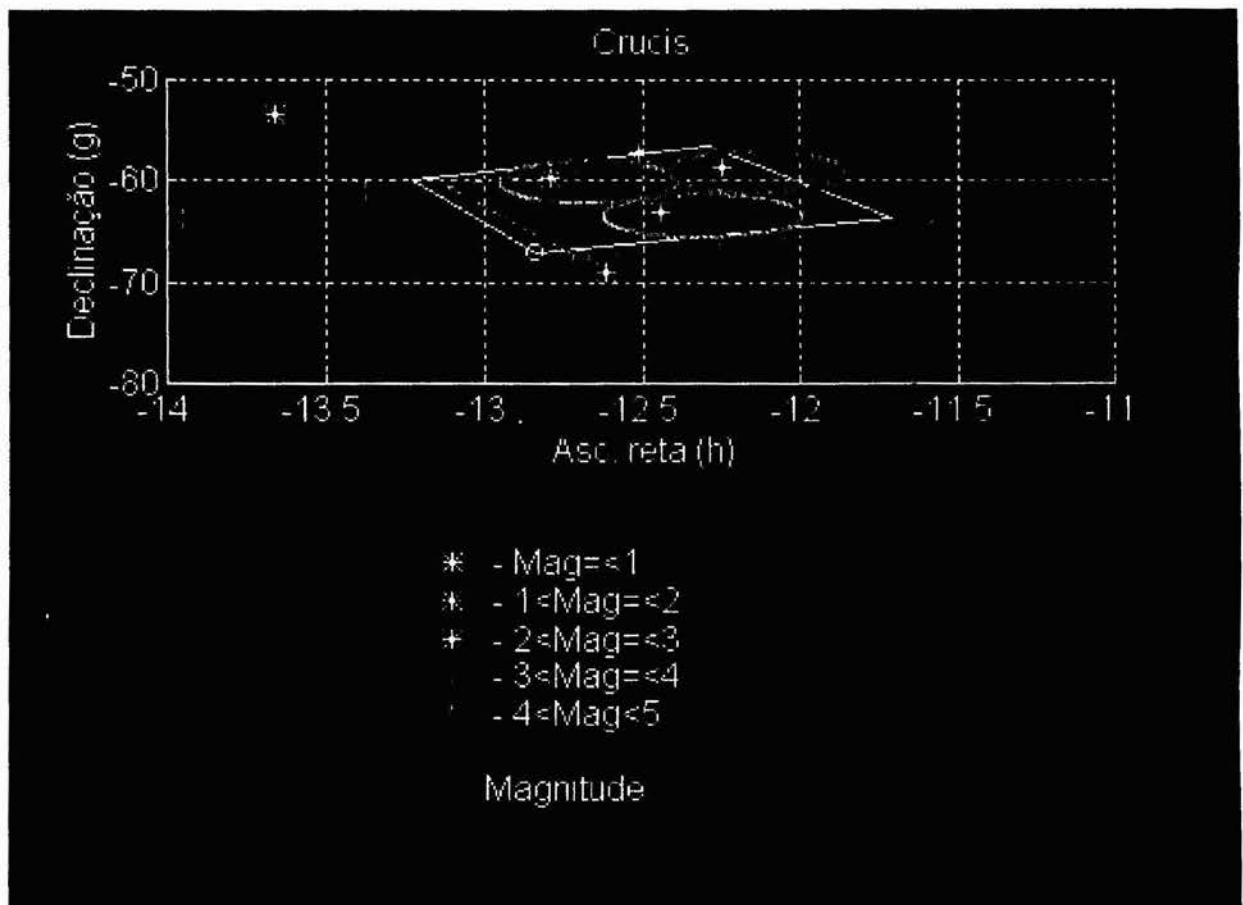


Fig. 32. - Representação gráfica da técnica de comparação direta.

Para isso, devemos ter uma estimativa do melhor raio ou tolerância  $\epsilon$  adotado para a circunferência de identificação. O procedimento de escolha do raio pode tomar dois caminhos: escolha arbitrária ou escolha otimizada por métodos estatísticos.

As técnicas estatísticas aqui apresentadas visam uma otimização na escolha da tolerância  $\epsilon$  para minimizar erros e maximizar identificações corretas. A escolha de um  $\epsilon$  muito pequeno leva a poucas identificações corretas, porém com maior certeza. Por outro lado, para que tenhamos maior porcentagem de identificações corretas devemos aumentar  $\epsilon$ , levando também a um aumento de identificações errôneas.

Estas técnicas estatísticas baseiam-se na técnica dos múltiplos critérios [b15.] que leva em conta critérios de desempenho utilizando-se de métodos estatísticos fundamentados em quatro probabilidades que resumem-se em:

I - Probabilidade de identificação correta:  $P0 = (1-h) \cdot q \cdot \mu \cdot e^{-\mu} / (1-e^{-\mu})$

II - Probabilidade de não identificação:  $P1 = (1 - q + q \cdot h) \cdot e^{-\mu}$ ;

III - Probabilidade de identificação errada:  $P3 = P1 \cdot \mu$ ;

IV - Probabilidade de identificação ambígua:  $P2 = 1-P0-P1-P3$ ;

Onde  $q$  é a probabilidade de que a estrela esteja no catálogo (adotado 0,99) e  $\mu$  é dado pela expressão:  $\mu = 2\pi\rho q \cdot (1 - \cos \epsilon)$ , onde  $\rho$  é a densidade estelar. Ainda:

$$h = e^{-\frac{\epsilon^2}{2\sigma^2}}$$

com  $\sigma$  sendo erro associado ao sensor estelar e à incerteza da estimativa a priori da atitude (sensor de pré-estimativa).

Esta técnica utiliza distribuição de **Poisson** para montar as probabilidades (Ver fig. 33).

Essas quatro probabilidades são utilizadas para escolher o melhor  $\epsilon$ , seguindo-se os conceitos de aceitação do valor para a otimização da identificação.

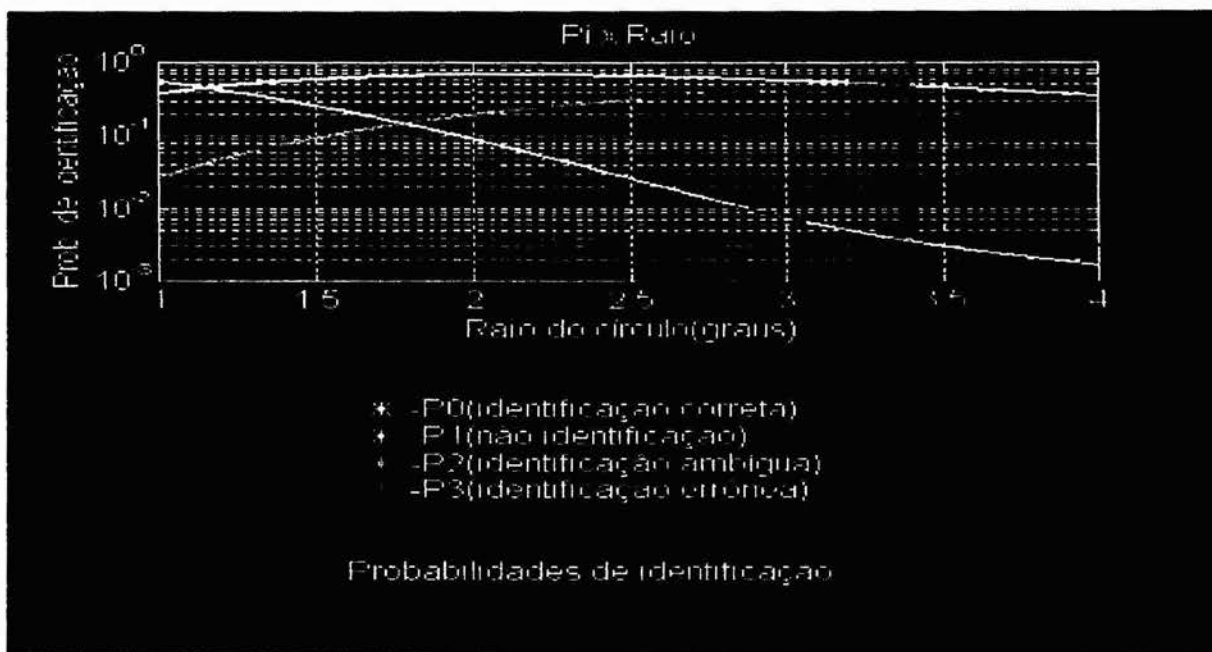


Fig. 33. - Probabilidades de ocorrência para os vários tipos de identificação com relação ao raio escolhido na técnica da comparação direta.

### 3.3. Técnica dos múltiplos critérios

A técnica descrita a seguir é um dos pontos alvo deste trabalho, pois através dela é possível otimizar a identificação das estrelas segundo alguns critérios pré-determinados que satisfaçam os requisitos estipulados por projetistas ou pela própria necessidade da missão, ou ainda pelas características construtivas do sistema e do *software*.

Esta técnica prevê e possibilita a aplicação de faixas de permissão aos vários tipos de identificação, estipulando como deve se comportar cada um de acordo com o critério para ele adotado.

Para a aplicação desta técnica, temos por ponto de partida as funções (acima definidas) de probabilidade de ocorrência para os vários tipos de identificação (correta, errônea, ambígua e não identificação) com relação ao raio para a técnica de comparação direta[b9.]. Estipula-se a faixa que se deseja para a ocorrência de cada tipo (Ver fig. 34) na qual são levadas em conta a porcentagem de ocorrências que considera-se boa e a porcentagem que considera-se ruim.

Para este desenvolvimento dos testes foram adotadas as faixas ruins de 50 e 30% para a identificação correta e para a ambígua respectivamente, sendo em seguida constatado, num teste com a região crucis que as melhores faixas são as seguintes:

(i)	Tipo de identificação	Bom %	Ruim %
-1	Errônea	0	5
0	Não identificação	0	15
1	Correta	100	65
2	Ambígua	0	15

Fig. 34. - Faixas de aplicação ótimas às probabilidades de identificação.



Dessas porcentagens, passamos a números reais (de 0 a 1) e aplicamos a cada uma das probabilidades em função do raio (ex.: 65%=0.65):

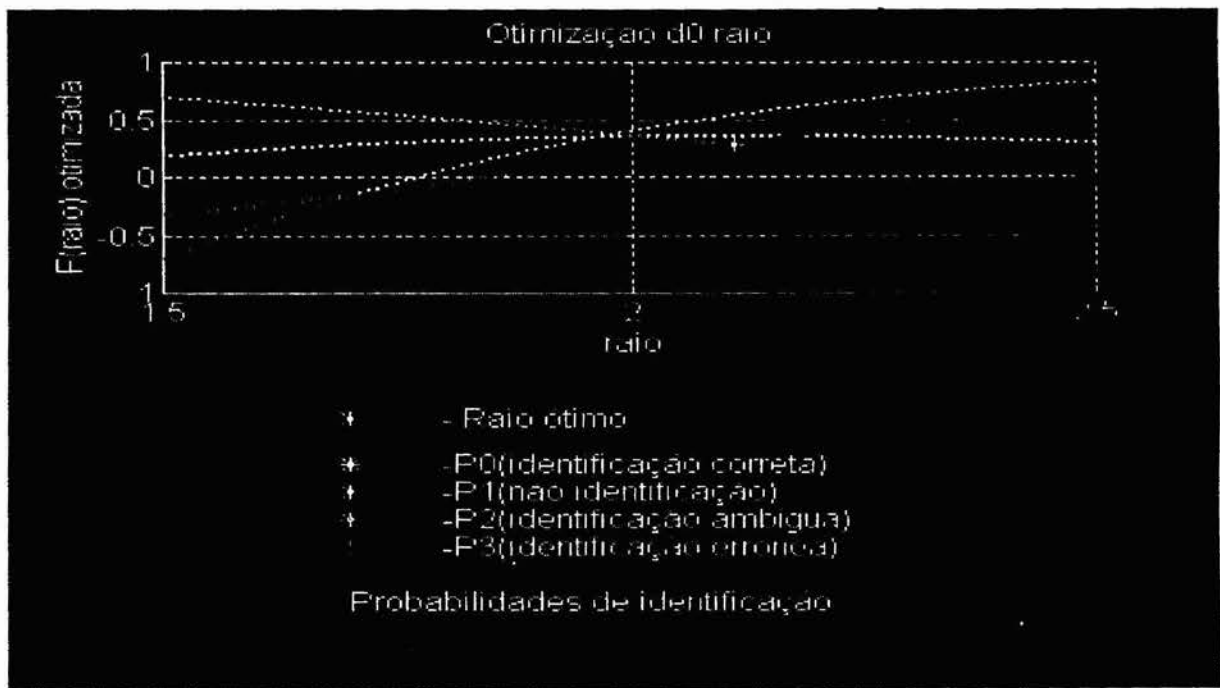
$$f_i (r) = \frac{P_i (r) - Ruim_i}{Bom_i - Ruim_i}$$

Com cada uma das funções modificadas, definimos a função ótima abaixo:

$$f_{ótima} (r) = \text{Min} (i) \{ f_i (r) \}, i \in \{-1, 0, 1, 2\}$$

Assim, temos em mãos uma função que leva em conta todos os “múltiplos critérios” a serem aplicados às funções de probabilidade de ocorrência dos tipos de identificação, ou seja, é uma função representativa de uma condição estipulada. Com ela, agora podemos obter a situação de máxima otimização da escolha do raio de identificação.

Ao tomarmos o valor do raio para o qual esta função ótima é maximizada, otimizamos ao máximo a situação desejada. Este ponto significa que se nos desviarmos um pouco à esquerda ou um pouco à direita, podemos otimizar um certo tipo, mas no entanto estaremos prejudicando o critério adotado para um outro tipo (Ver fig. 35). Assim teremos o máximo equilíbrio entre os critérios neste ponto de máxima função ótima.



**Fig. 35. - Função ótima, otimização e equilíbrio dos critérios para as probabilidades de identificação.**

Com estes resultados, podemos então dar um melhor direcionamento e criar uma sistematização na pesquisa de identificação por comparação direta.

# Capítulo IV

## 4. Descrição do procedimento e resultados finais

### 4.1. Descrição do procedimento

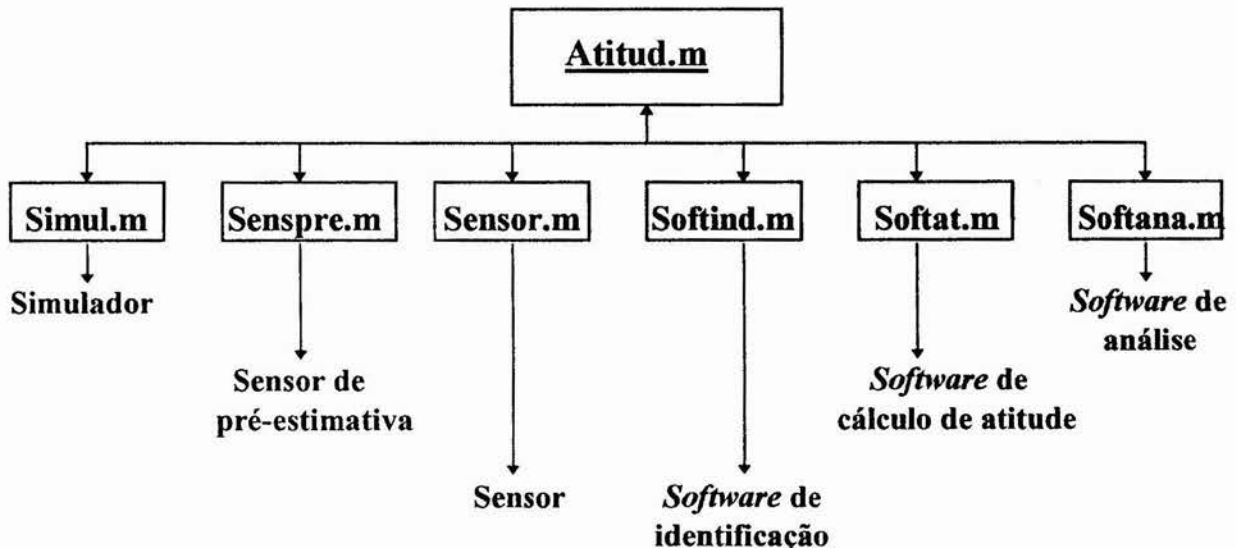
O *software* para o estudo neste trabalho, foi desenvolvido no ambiente *PC-Matlab* através do método de programação em módulos, ou seja, um programa principal, representando todo o sistema estudado, que acessa uma série de outros programas auxiliares que executam funções específicas de cálculo e análise.

A vantagem do método modular de programação encontra-se na facilidade de introdução de alterações nas funções a serem executadas, correções ou modificação na estrutura de funcionamento do sistema, sem que necessitemos modificar todo o programa principal, além de permitir uma direta análise dos resultados e da coerência dos cálculos executados por cada módulo.

Nesta seção apresentaremos as rotinas principais construídas com a citação de suas funções básicas e explicação mais detalhada, apresentando alguns fluxogramas de funcionamento e transferências de dados.

Em seguida serão apresentadas duas seções de funcionamento do *software* e analisadas as suas condições, assim como serão feitas comparações.

- Diagrama de acesso de rotinas principais:



• Fluxograma Programa principal:

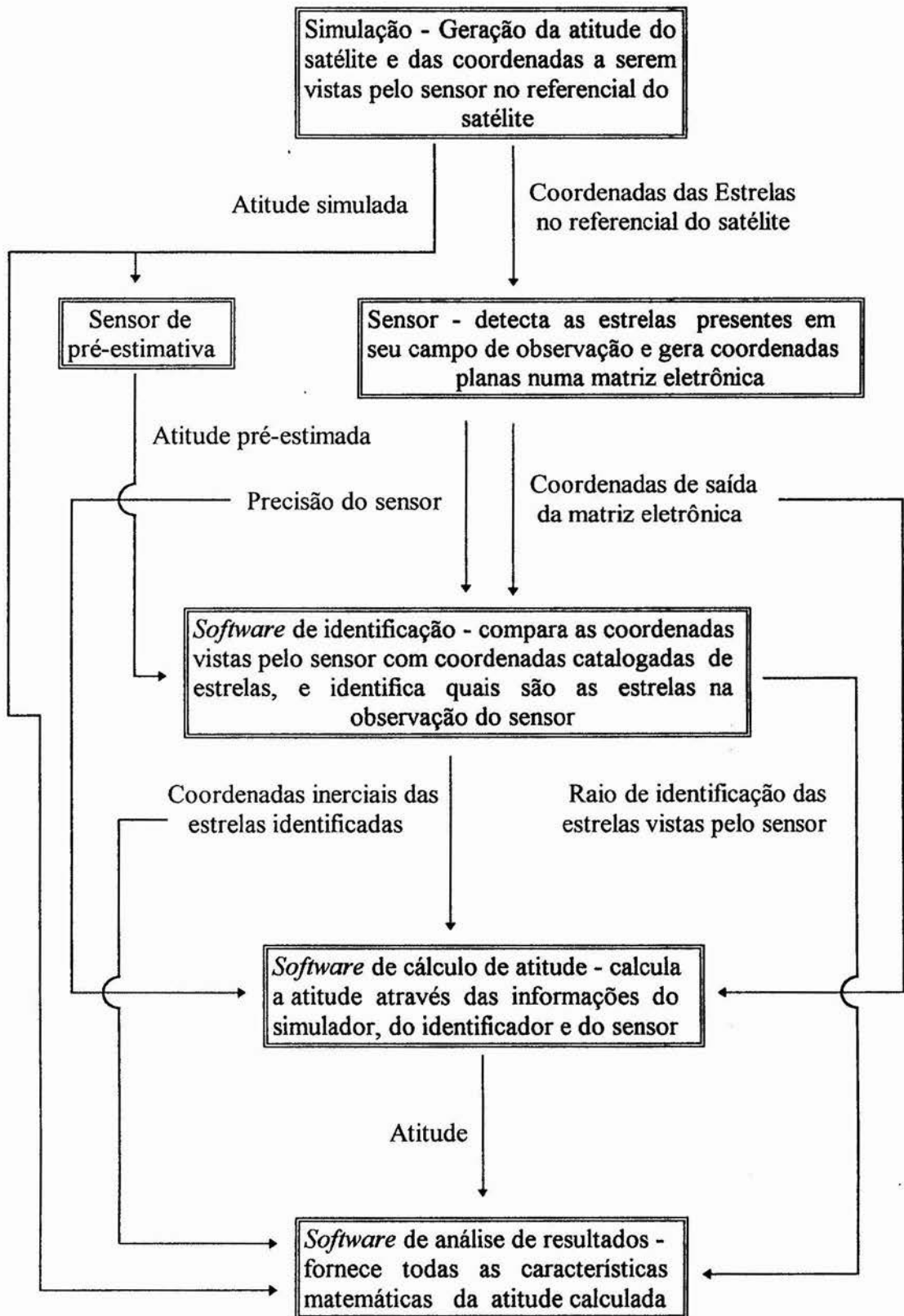


Fig. 36. - Sequência para a simulação de funcionamento do sistema de determinação de atitude.

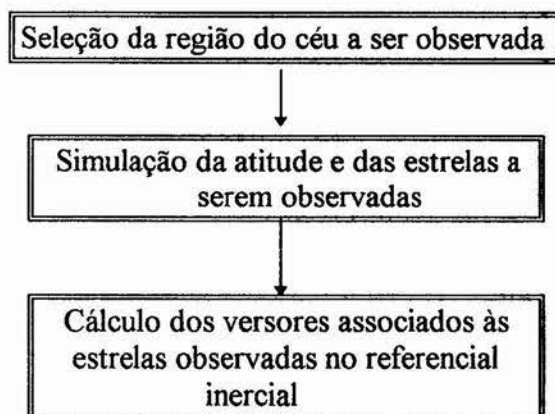
O fluxograma acima (Ver fig. 36) representa todo o desenvolvimento. As informações das estrelas a serem observadas são geradas por um simulador e a atitude pré-estimada é gerada por um sensor de pré-estimativa a partir da atitude gerada pelo simulador. A informação das estrelas é passada a um *software* que representa o sensor estelar e que gera toda a informação sobre as estrelas vistas a partir de sua matriz eletrônica.

Junto com a pré estimativa da atitude, as informações das estrelas vistas passam ao sistema de identificação. Na identificação são selecionadas as estrelas a serem realmente utilizadas no cálculo da atitude.

Posteriormente as coordenadas das estrelas selecionadas são repassadas ao sistema de cálculo junto com as coordenadas vistas pela matriz do sensor. Nesta etapa, a atitude final é calculada, sendo posteriormente analisada e comparada com a pré estimada por um sistema de análise que então fornece todas as características desejadas sobre a atitude calculada.

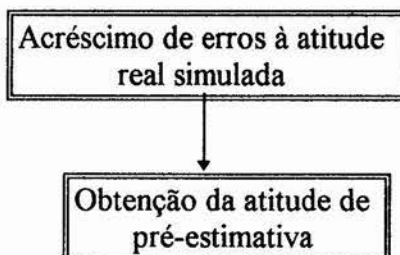
- Rotinas auxiliares do sistema:

### Simul.m



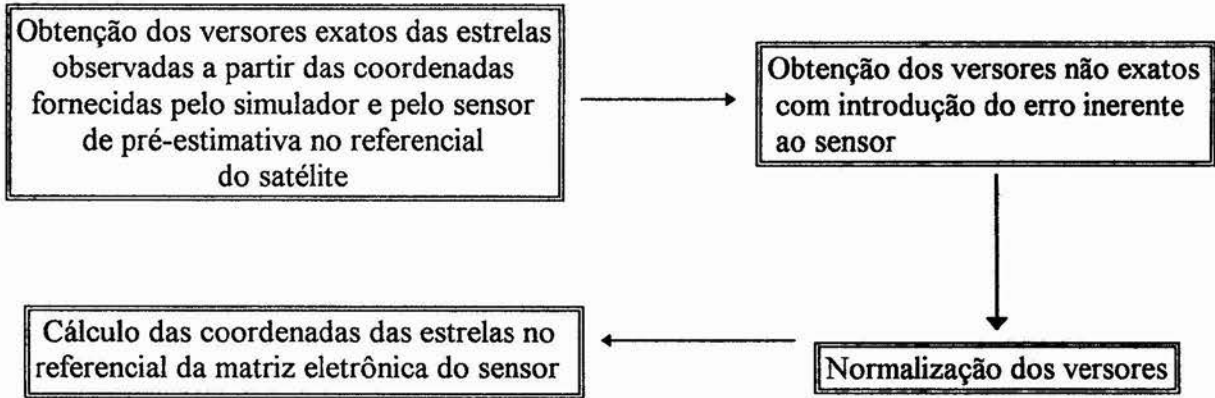
A rotina Simul.m propõe-se, com interação de um usuário, a um simulador que gera todas as informações necessárias ao cálculo da determinação de atitude como se as mesmas fossem provenientes do meio-ambiente onde se encontraria o satélite. Dessa forma a rotina simula o céu e o posicionamento do satélite com relação ao mesmo.

### Senspre.m



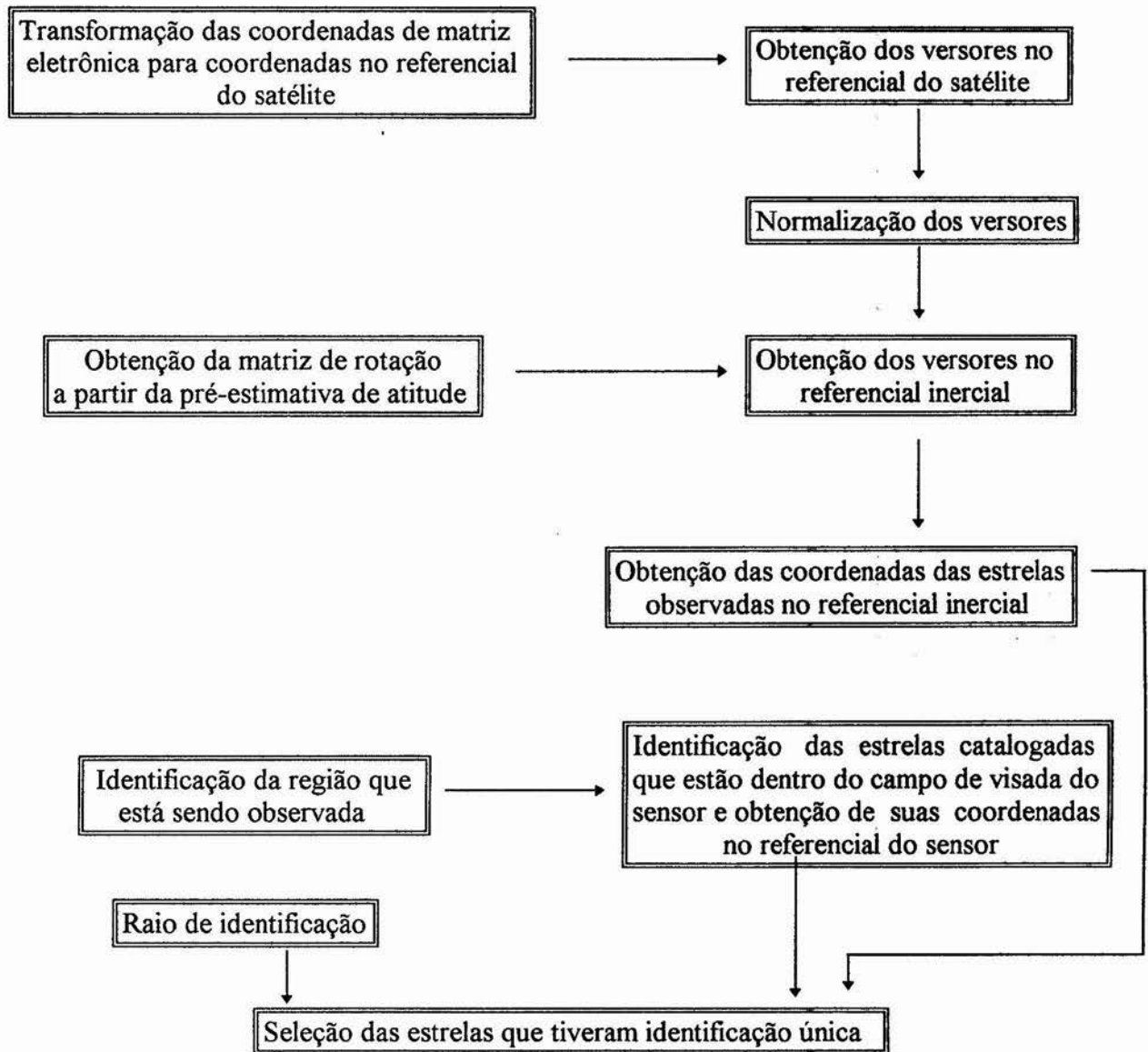
A rotina Senspre.m executa o papel de um sensor auxiliar de pré-estimativa que adquire uma pré-estimativa da atitude real e a fornece para o sistema.

### Sensor.m



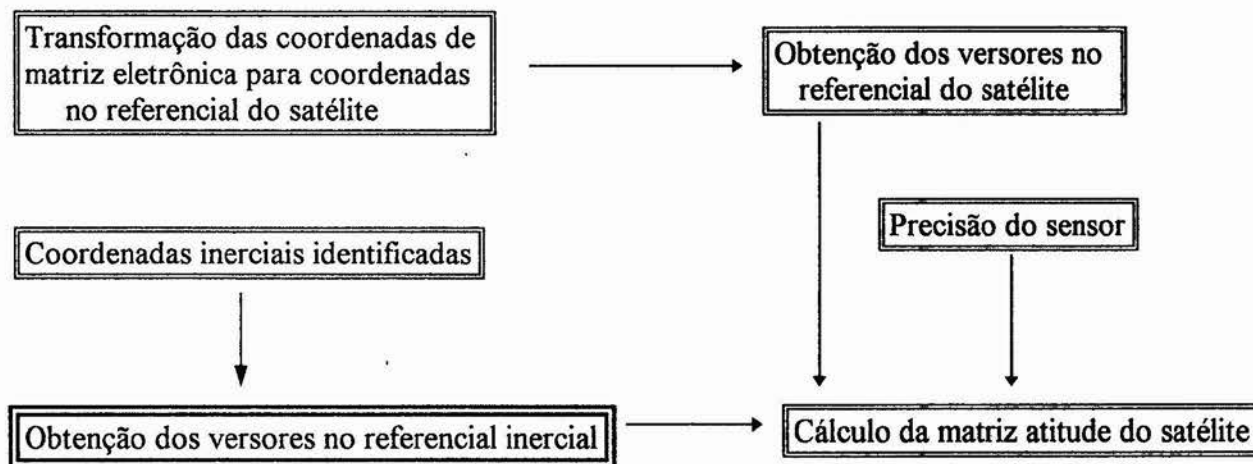
A rotina **Sensor.m** executa o papel de um sensor com uma matriz eletrônica CCD que é sensibilizada às estrelas observadas em seu campo de visada, e fornece coordenadas num referencial plano.

### Softind.m



A rotina Softind.m realiza as identificações das estrelas observadas pelo sensor comparando-as com as coordenadas catalogadas num certo raio de identificação em torno das posições observadas, tendo apenas uma única identificação para cada observação. Se a observação não possuir identificação ou tiver identificação ambígua, nenhuma coordenada associada será repassada. No entanto ainda existe a possibilidade de uma identificação errônea.

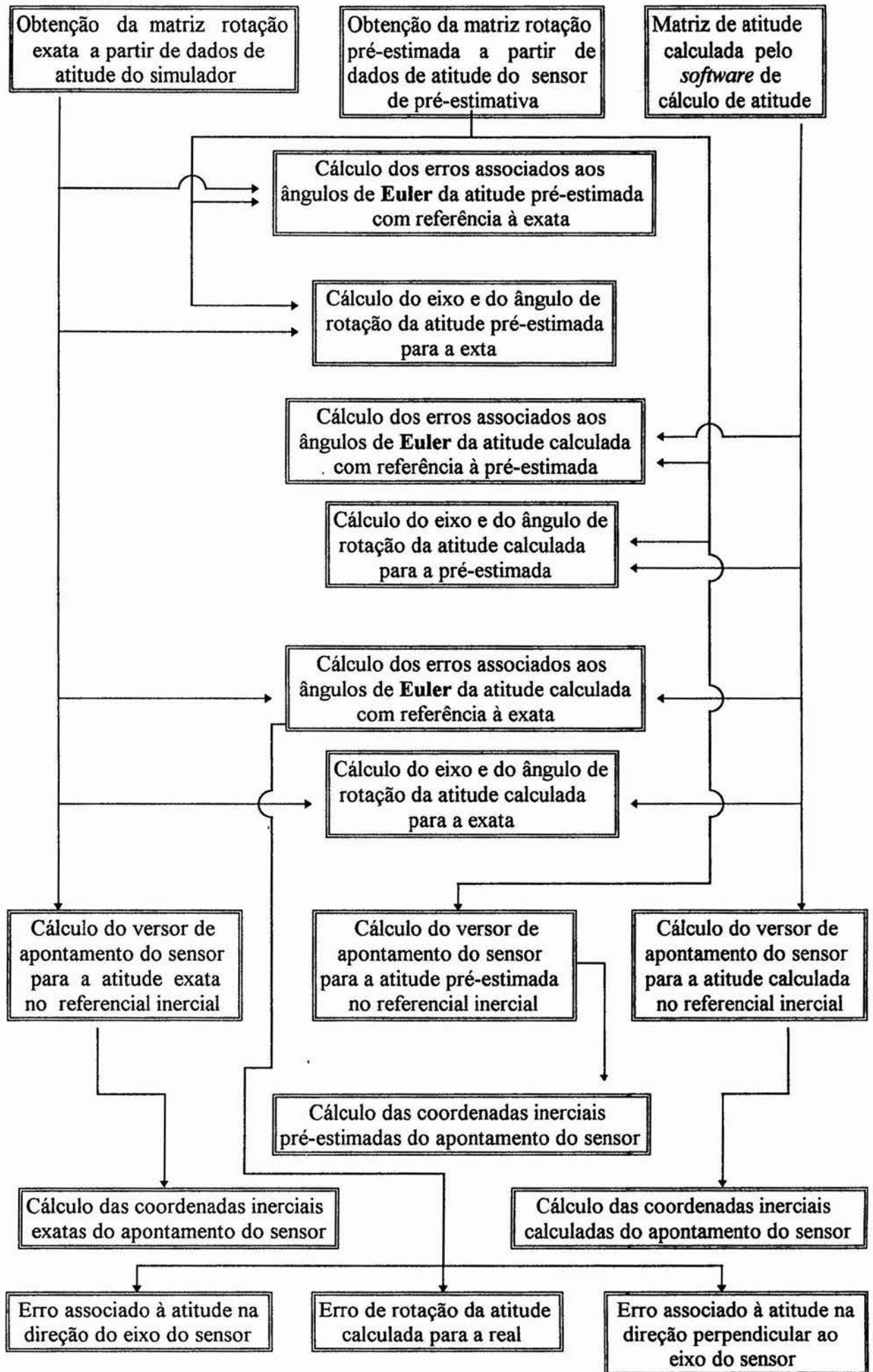
### Softat.m



A rotina Softat.m calcula a matriz de rotação ou atitude do satélite no referencial inercial com base nos dados fornecidos pelo sensor e pelo identificador de estrelas.

### Softana.m

O fluxograma apresenta-se em seguida:





A rotina Softana.m calcula itens de análise de toda a situação do cálculo de atitude executado com relação à atitude real simulada pelo simulador e à atitude pré-estimada pelo sensor de pré-estimativa, tais como erros nos ângulos de Euler, versores e coordenadas inerciais de apontamento do sensor, eixo e ângulo de rotação de uma atitude para a outra e erros associados à atitude nas direções paralela e perpendicular ao sensor

• **Seção de testes:**

A seguir serão demonstradas duas seções de testes como exemplo do procedimento, onde serão variados as precisões do sensor de pré-estimativa ( $1^{\circ}$  e  $2^{\circ}$ ), a forma de simulação e escolha do raio com aplicação dos requisitos ótimos obtidos. Os diversos testes realizados, nos quais baseiam-se os resultados a serem apresentados, estão tabelados no apêndice I.

Nas representações gráficas abaixo, em primeiro temos a representação da situação inicial, onde o retângulo branco representa a atitude simulada, o verde a atitude pré-estimada e os círculos verdes tem seu centro nas posições das estrelas vistas pelo sensor com o devido raio utilizado. Em segundo temos o gráfico de probabilidades para as ocorrências dos tipos de identificação. Em terceiro temos a aplicação da técnica dos múltiplos critérios [b15.] e por último a representação da situação final de cálculo onde os círculos amarelos estão centrados nas estrelas do campo que foram identificadas, o retângulo branco é a atitude pré-estimada e o verde a calculada.

**Condições 1:**

Região observada: Crucis - 15 estrelas

Cálculo dos desvios do centro da região: Aleatório

Precisão do sensor de pré-estimativa:  $1^{\circ}$

Precisão do sensor estelar:  $3'$

Raio do círculo:  $2.1100^{\circ}$

Raio ótimo:  $2.1100^{\circ}$

Número de estrelas no campo: 4

Densidade de estrelas no campo: 0.0369 estrelas / grau<sup>2</sup>

**Resultados:**

(Ver fig. 37)

Análise com base de dados fornecidos pelo sensor de estrelas, sensor de atitude pré-estimada e simulador.

Ângulos de Euler321 associados à atitude pré-estimada:

188.9032 58.0525 -115.5480

Ângulos de Euler321 associados à atitude calculados:

189.1195 59.9181 -115.7526

Ângulos de Euler321 associados à atitude real simulada:

189.1127 59.8668 244.5467

Erros calculado/pré-est. nos ângulos de euler321(g,m,s):

0 12 59

1 51 56

0 12 17

Erros calculado/real nos ângulos de euler321(g,m,s):

0 0 25

0 3 5

0 17 58

Erros pré-est./real nos ângulos de euler321(g,m,s):

0 12 34  
-1 48 52  
0 5 41

Matriz de atitude 321 calculada A321:

-0.4949 -0.0794 -0.8653  
0.7007 0.5525 -0.4515  
0.5140 -0.8297 -0.2178

Matriz de atitude 321 pré-estimada A:

-0.5228 -0.0819 -0.8485  
0.6896 0.5446 -0.4774  
0.5012 -0.8347 -0.2282

Matriz de atitude 321 real As:

-0.4957 -0.0795 -0.8649  
0.7030 0.5480 -0.4533  
0.5100 -0.8327 -0.2158

Versor de giro de erro de A321 para A(no ref. do sat.):

-0.2102  
-0.4765  
0.8537

Rotação em torno de n calc./pré-est.:

1 55 1

Versor de giro de erro de A321 para As(no ref. do sat.):

-0.9860  
-0.0820  
0.1449

Rotação em torno de n calc./real:

0 18 34

Versor de giro de erro de A para As(no ref. do sat.):

0.0399  
0.4827  
-0.8749

Rotação em torno de n pré-est./real:

1 49 3

Erro associado à atitude na direção do eixo do sensor(g,m,s):

0 18 18

Erro associado à atitude, perpendicular ao eixo do sensor(g,m,s):

0 3 5

Erro de rotação de atitude calculada/real(g,m,s):

0 18 34

Versor de apontamento pré-estimado do sensor:

-0.5228 0.6896 0.5012

Coordenadas de apontamento pré-estimado do sensor:

8.4777 30.0773

Versor de apontamento calculado do sensor:

-0.4949 0.7007 0.5140

Coordenadas de apontamento calculado do sensor:

8.3490 30.9284

Versor de apontamento real do sensor:

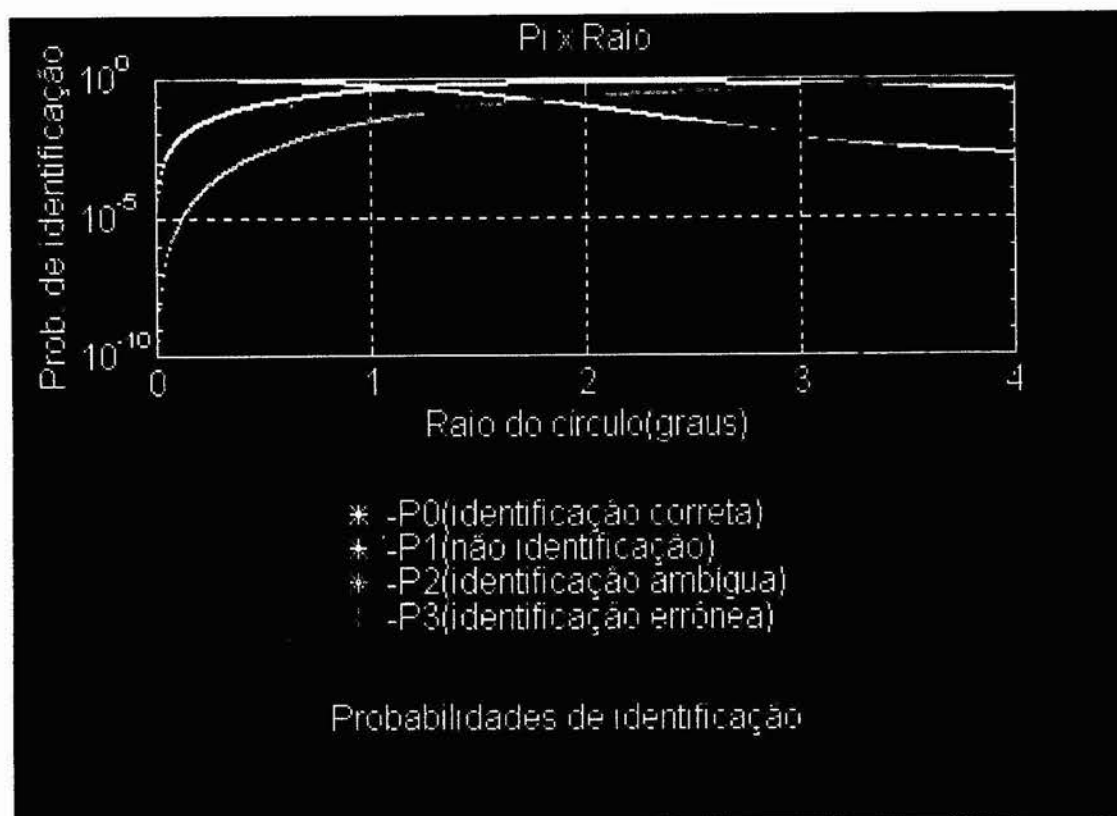
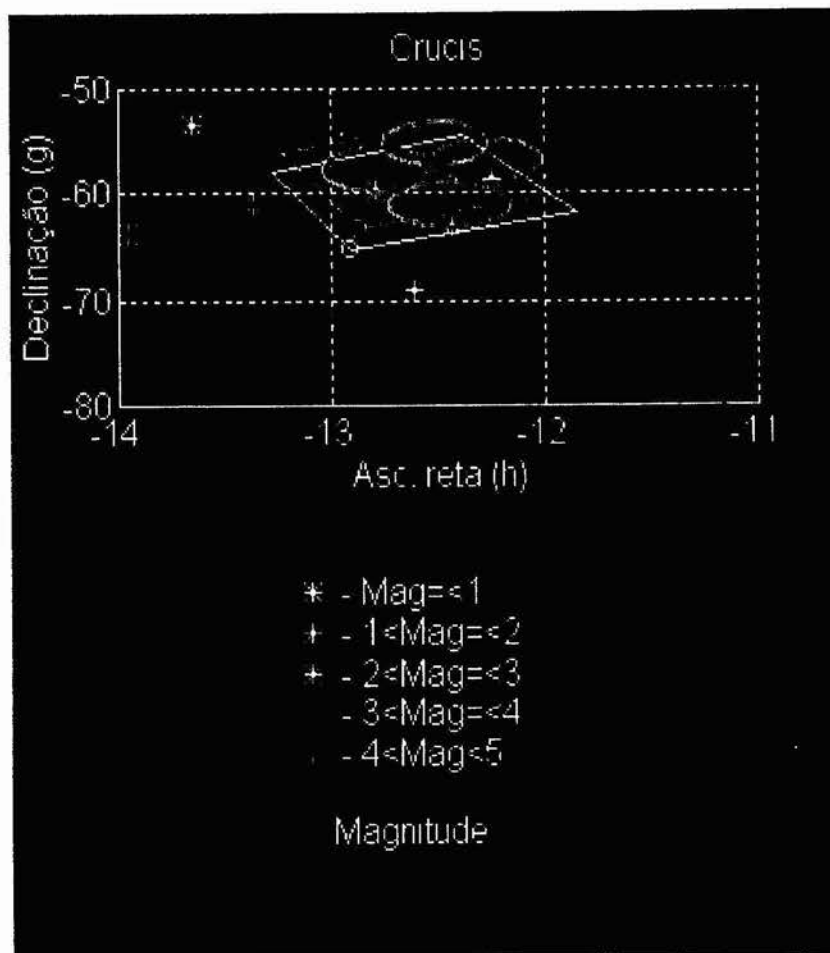
-0.4957 0.7030 0.5100

Coordenadas de apontamento real do sensor:

8.3458 30.6644

Número de estrelas identificadas:

4



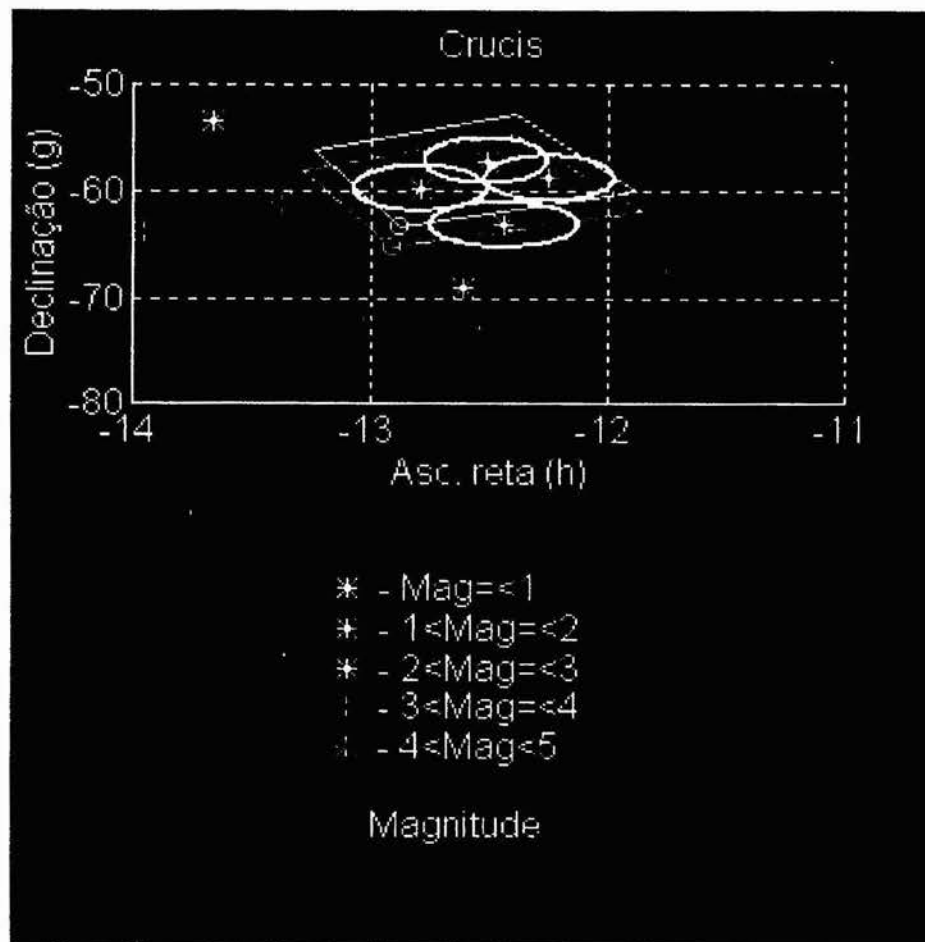
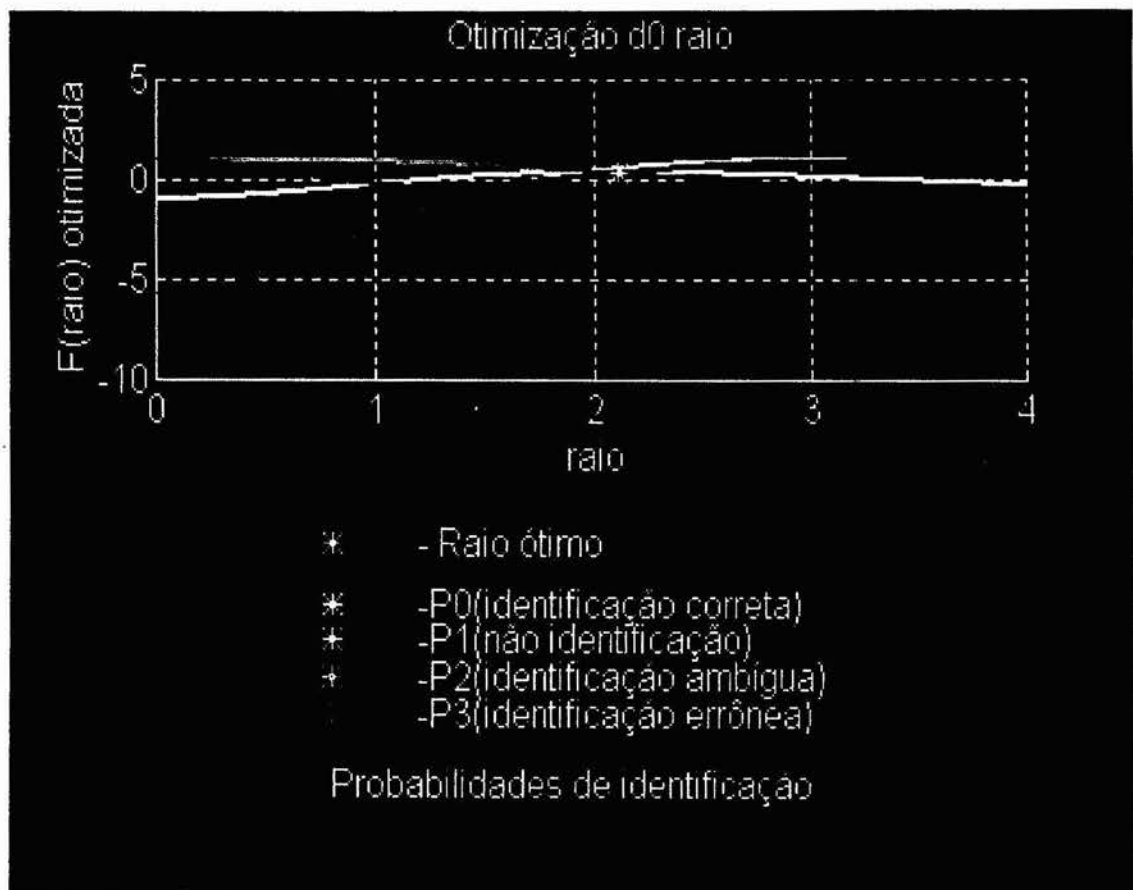


Fig. 37. - Representações gráficas dos resultados do primeiro teste.

## Condições 2:

Região observada: Crucis - 15 estrelas  
Cálculo dos desvios do centro da região: Dados fornecidos  
Precisão do sensor de pré-estimativa: 2°  
Precisão do sensor estelar: 3'  
Raio escolhido do círculo: 2.4400°  
Raio ótimo: 3.4400° (não identificou)  
Número de estrelas no campo: 3  
Densidade de estrelas no campo: 0.0369 estrelas / grau<sup>2</sup>

## Resultados:

(Ver fig. 38)

Análise com base de dados fornecidos pelo sensor de estrelas, sensor de atitude pré-estimada e simulador.

Ângulos de Euler321 associados à atitude pré-estimada:

176.6736 59.0221 29.8659

Ângulos de Euler321 associados à atitude calculados:

175.1312 59.3000 40.2700

Ângulos de Euler321 associados à atitude real simulada:

180.2970 57.1831 34.0000

Erros calculado/pré-est. nos ângulos de euler321(g,m,s):

-1 32 33

0 16 41

10 24 15

Erros calculado/real nos ângulos de euler321(g,m,s):

-5 9 57

2 7 1

6 16 12

Erros pré-est./real nos ângulos de euler321(g,m,s):

-3 37 24

1 50 20

-4 8 3

Matriz de atitude 321 calculada A321:

-0.5087 0.0433 -0.8599

-0.6186 -0.7131 0.3300

-0.5988 0.6997 0.3895

Matriz de atitude 321 pré-estimada A:

-0.5138 0.0299 -0.8574

-0.4765 -0.8410 0.2563

-0.7134 0.5403 0.4464

Matriz de atitude 321 real As:

-0.5419 -0.0028 -0.8404

-0.4656 -0.8315 0.3031

-0.6996 0.5556 0.4493

Versor de giro de erro de A321 para A(no ref. do sat.):

0.9971

-0.0264

-0.0707

Rotação em torno de n calc./pré-est.:

11 40 52

Versor de giro de erro de A321 para As(no ref. do sat.):

0.9513

-0.0247

-0.3074

Rotação em torno de n calc./real:

11 8 4

Versor de giro de erro de A para As(no ref. do sat.):

-0.3725

0.1990

-0.9064

Rotação em torno de n pré-est./real:

2 52 10

Erro associado à atitude na direção do eixo do sensor(g,m,s):

10 35 30

Erro associado à atitude, perpendicular ao eixo do sensor(g,m,s):

3 26 2

Erro de rotação de atitude calculada/real(g,m,s):

11 8 4

Versor de apontamento pré-estimado do sensor:

-0.5138 -0.4765 -0.7134

Coordenadas de apontamento pré-estimado do sensor:

14.8562 -45.5086

Versor de apontamento calculado do sensor:

-0.5087 -0.6186 -0.5988

Coordenadas de apontamento calculado do sensor:

15.3711 -36.7871

Versor de apontamento real do sensor:

-0.5419 -0.4656 -0.6996

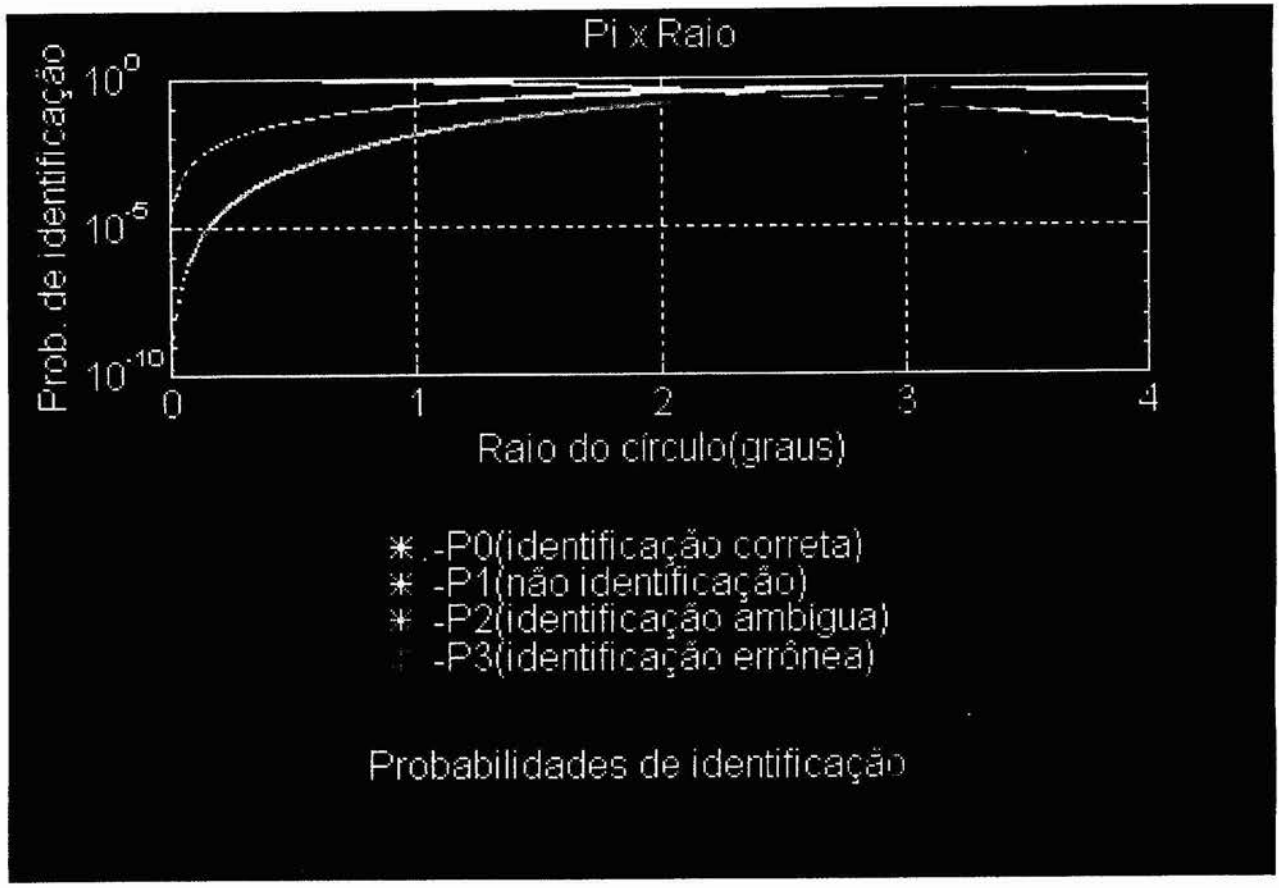
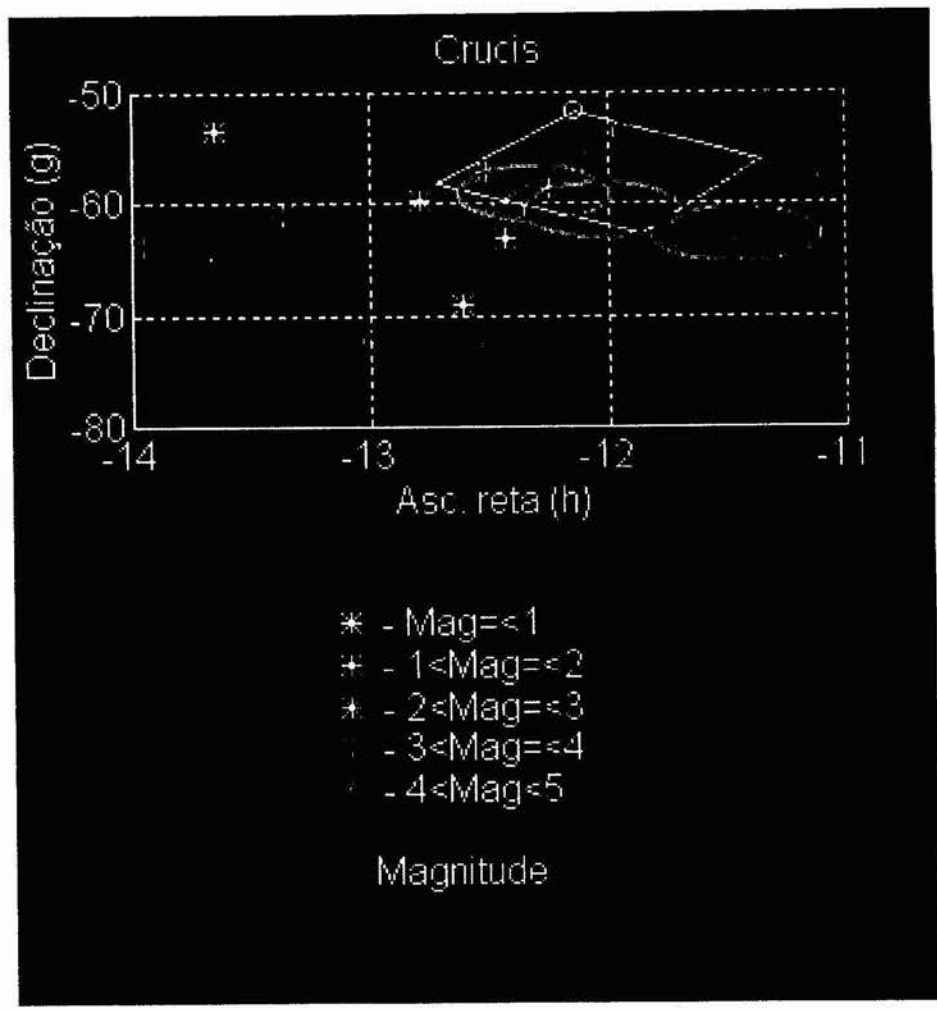
Coordenadas de apontamento real do sensor:

14.7113 -44.3964

Número de estrelas identificadas:

3

2



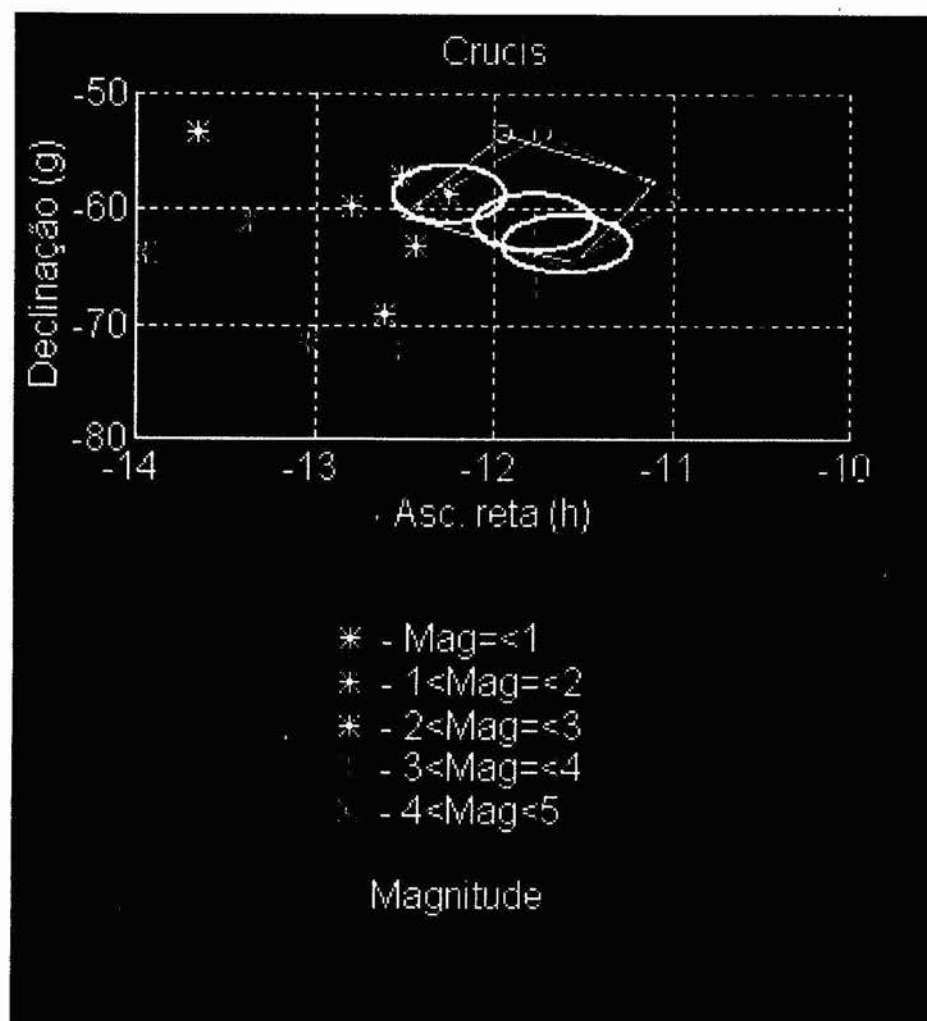
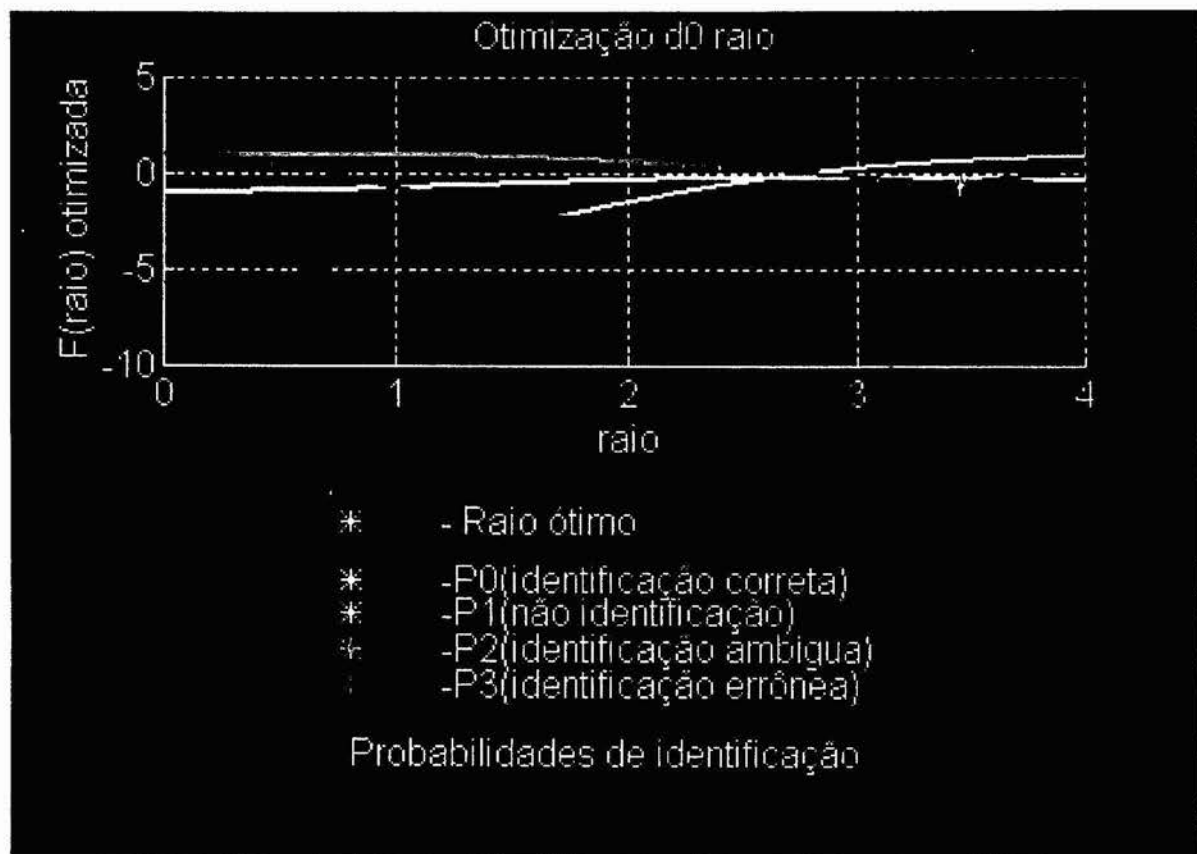


Fig. 38. - Representações gráficas dos resultados do segundo teste.



As condições alteradas nos dois testes, precisão do sensor de pré-estimativa e o raio de identificação das estrelas, são determinantes para a identificação, uma vez que são elas que levam a um cálculo mais, ou menos, preciso, segundo o número de estrelas identificadas pelo sistema. É claro também que a precisão do sensor estelar influi, mas em comparação com a pré-estimativa, a maior parte do desvio é devido a esta última.

As condições de atitude são diferentes para os dois testes, uma vez que realizou-se o segundo posicionando do campo do sensor diferentemente ao do resultado aleatório calculado no primeiro teste. Apesar da diferença, é possível notar as características do sistema, e ainda este fator ajuda na percepção da influência da disposição geométrica das estrelas no campo.

Podemos perceber claramente pelas figuras e pelos dados que a alteração dos valores de precisão e raio modificam drasticamente os resultados obtidos no cálculo e determinação da atitude através da identificação das estrelas observadas.

Enquanto o fator de precisão altera o posicionamento do campo visto pelo sensor com relação ao campo real e dessa forma influenciando no posicionamento das estrelas observadas, o raio de identificação atua na qualidade da identificação, aumentando ou diminuindo o número de estrelas identificadas. Perceba que ao aumentarmos o tamanho do raio, mais estrelas se enquadram dentro do círculo formado, aumentando a probabilidade de identificação ambígua, ou errônea dependendo da precisão utilizada que pode afastar muito o posicionamento do local real. Ao diminuirmos, diminui-se a probabilidade de identificações ambíguas e errôneas, mas aumenta-se a de não identificação, não enquadrando-se nenhuma estrela dentro do círculo formado. Para normalizar a pesquisa em cima do sistema a técnica dos múltiplos critérios [b15.] permite a escolha de um raio através de critérios estipulados, livrando os resultados de uma influência subjetiva por parte do usuário na escolha do raio. No entanto, em algumas circunstâncias o raio ainda é ditado pelo usuário (como é o caso do segundo teste onde o raio ótimo não foi capaz de promover uma identificação satisfatória para o cálculo da atitude).

Dessa forma, a importância na determinação de um raio não só ótimo como também adequado a cada situação em particular é vital ao bom funcionamento de todo o sistema para que possa fornecer os melhores resultados.

## 4.2. Resultados finais

Os resultados aqui descritos, como dito acima, foram obtidos de uma série de testes com o *software* e são apresentados em forma de gráficos que descrevem em forma de histogramas o comportamento do *software* com relação a vários tópicos de interesse. Através destes resultados pudemos analisar e estudar as características de desempenho e funcionamento do sistema montado., das quais podemos tirar proveitosas conclusões que nos possibilitam um melhor desenvolvimento de métodos mais avançados. Para este desenvolvimento dos testes foram adotados critérios de faixas ruins de 50 e 30% para a identificação correta e para a ambígua respectivamente, sendo em seguida constatado, num teste com a região crucis que as melhores faixas (requisitos ótimos) são 65 e 15% respectivamente.

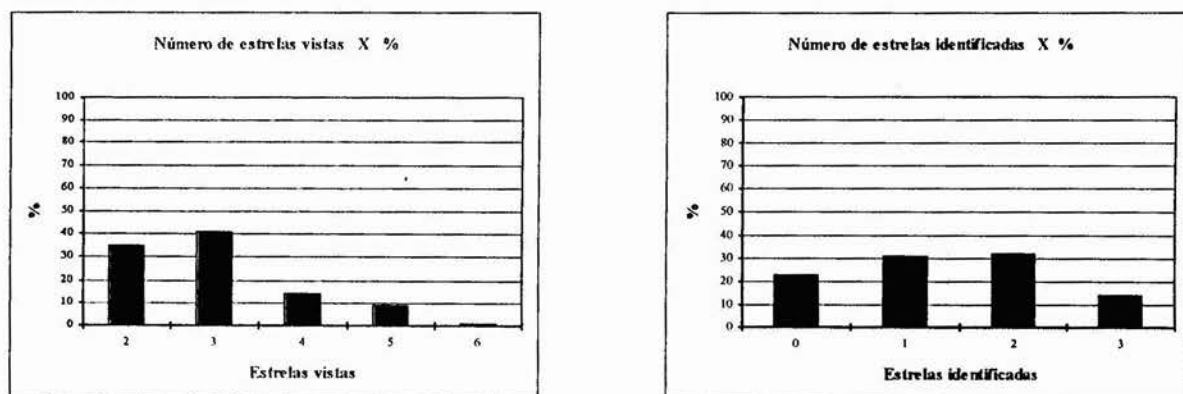
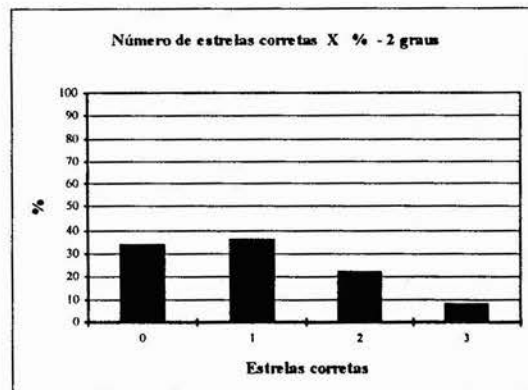
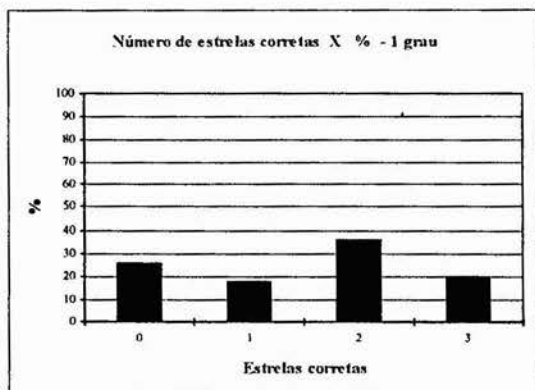
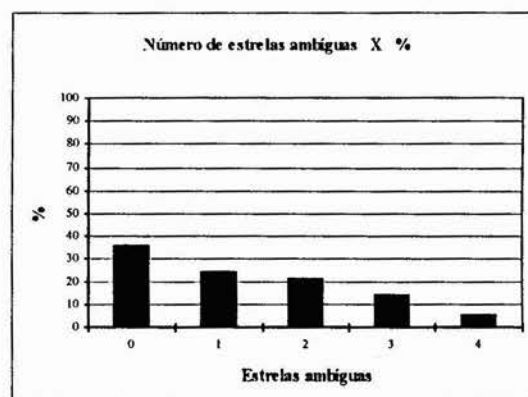
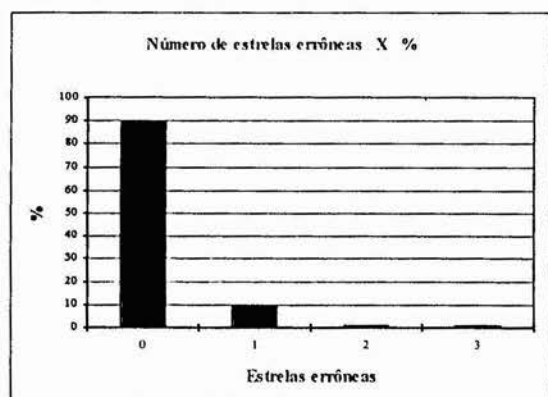


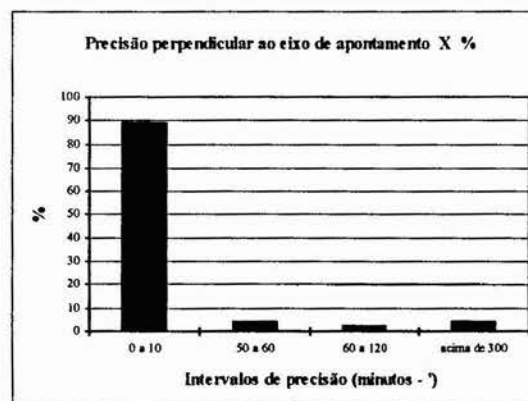
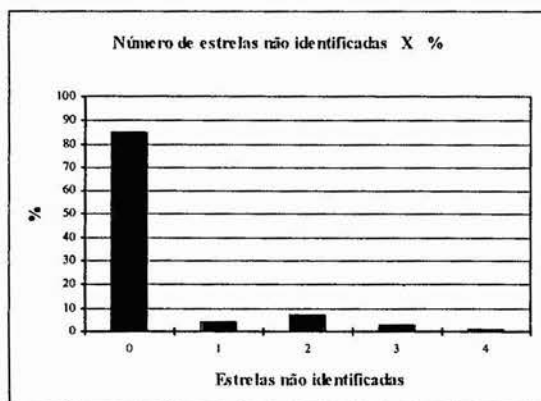
Fig.39 - Comparação entre % de estrelas vistas e % de estrelas identificadas.



**Fig.40 - Comparação entre % de estrelas identificadas corretamente para 1<sup>o</sup> e 2<sup>o</sup> de precisão do sensor de pré-estimativa.**



**Fig.41 - % de estrelas identificadas erroneamente e % de identificação ambígua(requisito de 30%).**



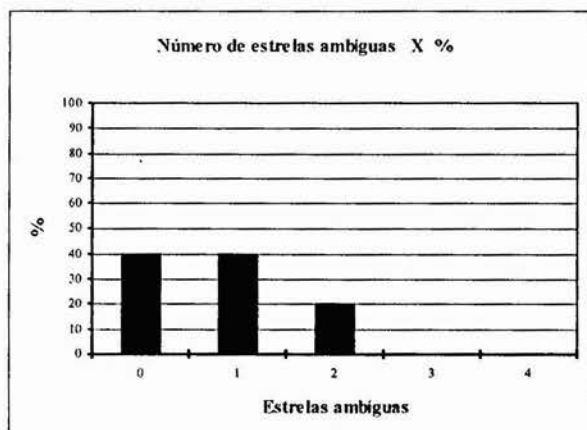
**Fig.42 - % de estrelas não identificadas e % de ocorrências de precisões obtidas em cálculo.**

### Conclusões:

Os histogramas acima e os testes numéricos realizados, fornecem diversas conclusões com respeito ao desempenho dos métodos desenvolvidos:

- O número de estrelas identificadas aproxima-se em, porcentagem máxima, ao número de estrelas vistas, o que é um bom resultado de desempenho (Ver fig. 39).
- A precisão de pré-estimativa é determinante, uma vez que sendo maior o erro (2<sup>o</sup>), a porcentagem de estrelas identificadas corretamente cai, ou seja, a atitude pré-estimada não está tão próxima da correta, influenciando a identificação (Ver fig. 40).

• O ótimo desempenho é notado pela grande porcentagem de identificações errôneas e não identificações nulas, acompanhada por uma tendência de identificações ambíguas nulas (maior %) (Ver fig. 41, 42 e 43). É notável que a distribuição de identificações ambíguas se apresentou esparsa (fruto dos requisitos adotados: faixa de 30%), o que nos permitiria num segundo instante um melhor desempenho se adotássemos como pré-estimativa a atitude calculada e reiterássemos com outros requisitos de probabilidade acentuando a identificação correta. Este processo de refinamento da precisão de pré-estimativa nos permitiria a identificação correta de muitas estrelas que em primeiro plano não foram identificadas, caindo numa classificação ambígua. Em prática, foi averiguada essa hipótese quando adotamos uma faixa mais restrita para a ambigüidade(15%) num teste com a região crucis, apresentando uma faixa muito mais ampla de identificações corretas e menores identificações ambíguas (Ver fig. 43).



(i)	Tipo de identificação	Bom %	Ruim %
-1	Errônea	0	5
0	Não identificação	0	15
1	Correta	100	65
2	Ambígua	0	15

**Fig.43. - % de estrelas ambíguas em cálculo (requisito de 15%) e faixas ótimas de aplicação das probabilidades.**

• Pelos testes fica clara a influência da distribuição e disposição (“entropia”) das estrelas no campo. Mesmo que a densidade no campo seja satisfatória, o fato de estarem aglomeradas piora o desempenho, enquanto que uma disposição mais esparsa é o ideal. É notável que algumas regiões seguem a hipótese inicial adotada de que seja seguida a distribuição de **Poisson**, enquanto que outras (ex.: orionis) fogem a esta regra notavelmente. É bastante evidente o melhor desempenho em regiões que seguem a hipótese adotada.

• A precisão com relação ao apontamento da atitude é melhor desenvolvida na direção perpendicular ao eixo de apontamento do sensor. Isto ocorre devido ao pequeno raio desenvolvido em torno deste eixo. Seria necessária uma rotação bem acentuada para que fosse significativa uma mudança de posição da estrela na referencial do satélite. Nos testes fica bem claro essa característica, estando a precisão 89% das vezes entre 0 e 10’, o que é bem satisfatório. Ou seja, verificou-se a previsão teórica inicial com relação a este tipo de precisão do sensor (Ver fig. 42).



# Capítulo Final

## 5. Comentários finais

Depois de descritos os conceitos e procedimentos que suportam o trabalho, passamos a descrever agora as atividades realizadas no decorrer do trabalho com opiniões a respeito de todo o contexto. Além disso, damos um breve discernimento da continuação do mesmo tentando atingir um objetivo final.

- Atividades realizadas durante o decorrer do trabalho:

Desde o início deste trabalho em agosto de 1995, foram realizadas algumas atividades em função do desenvolvimento do mesmo, ou sua complementação. A título de conhecimento ou informação simplesmente, essas atividades resumem-se em:

### a) Pesquisas bibliográficas:

Essas pesquisas referem-se em boa parte ao desenvolvimento do trabalho na obtenção de informações, mas também a uma ambientação aos recursos bibliográficos disponíveis no INPE e aquisição de conhecimentos em áreas alheias ao desenvolvimento propriamente dito.

### b) Visitas:

As visitas realizaram-se aos departamentos do INPE da área de controle e de testes de satélites, visando uma abrangência ampliada do processo desenvolvido em pesquisas espaciais com satélites no Brasil atualmente, além de uma visita a título de curiosidades e conhecimento geral ao centro de visitantes do INPE.

### c) Ambientação à área espacial de satélites:

Junto ao orientador e outros grupos de trabalho de outros tópicos de pesquisa foram realizadas breves reuniões onde discutiram-se tópicos básicos à pesquisa espacial fornecidos por uma apostila introdutória.

### d) Programação FORTRAN:

Foi desenvolvida uma breve tarefa em programação FORTRAN afim de revisar conceitos básicos e aplicações desta linguagem de programação em particular, devido ao fato de existirem muitas subrotinas prontas para o uso na instituição. No caso, foi desenvolvida uma rotina para o cálculo de trajetória orbital de uma espaçonave entre o planeta Terra e a Lua utilizando-se de uma subrotina previamente fornecida para a integração numérica..

### e) Ambientação com o equipamento de *hardware* do sensor:

Foram discutidas as capacidades, possibilidades e funcionamento de sensores já existentes e do sensor desenvolvido no INPE com as pessoas responsáveis pelo projeto do *hardware* junto ao orientador deste trabalho.

### f) Programação MATLAB:

Essa atividade foi um dos suportes básicos do trabalho, pois o *software* PC-MATLAB possibilita a implementação de todo o algoritmo. Dessa forma foi necessária uma ambientação ao *software* e às ferramentas por ele oferecidas.

g) Eventos:

Palestra no Observatório de São Carlos (CDA-CDCC-IFQSC-USP) sobre posicionamento, aquisição e determinação de atitude de satélites.

Participação no 2.º Seminário de Iniciação Científica com uma palestra de aproximadamente 20 minutos sobre o trabalho desenvolvido, além de assistir a muitas apresentações sobre os outros trabalhos de iniciação que se desenvolvem no INPE por outros estudantes.

• Opiniões gerais sobre a experiência:

No decorrer do desenvolvimento das atividades foi possível um amplo aproveitamento de todo o processo, apesar da distância entre o INPE e a instituição de origem EESC-USP-São Carlos.

Com o desenvolvimento deste trabalho de iniciação científica ficaram mais claros quais são os meios e processos utilizados numa pesquisa espacial, assim como todo o seu desenvolvimento, desmitificando um pré-conceito formado de que a pesquisa espacial é quase, ou senão impossível no nosso país.

Em opinião particular, o desenvolvimento deveria dar-se um pouco mais próxima da realidade da convivência com todo o equipamento de suporte e de forma mais amplificada, como conhecimento propriamente dito de um satélite desenvolvido no INPE, ou protótipo para uma pré-ambientação ao trabalho com o mesmo.

Em primeira visão, todo o desenvolvimento saiu-se perfeitamente bem, com mostras de uma ampla melhoria, além de sugerir idéias de um possível mestrado na área espacial pelo INPE.

• Continuação do trabalho:

a) Estudo de técnicas avançadas de identificação de estrelas:

Paralelamente à otimização do *software* serão estudadas e analisadas técnicas mais avançadas na identificação de estrelas afim de aumentar o desempenho no ponto particular da identificação, tornando o cálculo da atitude mais preciso.

Tais técnicas serão eventualmente pesquisadas através de uma ampla pesquisa bibliográfica, principalmente em artigos publicados recentemente, afim de permitir que tenhamos as mais avançadas em mãos, tendo em vista uma continuidade do trabalho numa possível prorrogação da iniciação científica e/ou numa pós graduação no INPE.

A pesquisa se estenderá através dos métodos de identificação sem necessidade de pré-estimativa de atitude como foi o caso deste trabalho inicial. Isto possibilita a utilização do sistema em condições mais especiais onde o satélite, ou nave, não tenha condições de utilizar outra referência senão as estrelas.

## 6. Referências bibliográficas e software

As referências para o desenvolvimento deste trabalho são baseadas em livros, artigos publicados, apostilas, catálogos e atlas de mapas estelares, onde toda a informação a respeito dos conceitos básicos, fundamentos teóricos e técnicas de aplicação foram obtidas.

Já os *softwares* utilizados baseiam-se em alguns títulos já consagrados pelo seu ótimo desempenho na realização de tarefas específicas para a editoração ou cálculo.

### a) Apostilas:

- 1. **Silvia Martins gonçalves, Lea.** *PC-MATLAB ver.3.1 - PC.* Usp - gráfica, São Carlos, SP, 1991;
- 2. *Matlab with simulink ver. 4.0 - help* impresso;
- 3. **Silva, Wilson Custódio Canesin.** *Movimento Orbital.* Publicação interna DMC - INPE, São José dos Campos, SP;
- 4. *Astronomia.* CDA-CDCC-IFQSC-USP, São Carlos, SP.

### b) Artigos e Livros:

- 1. **Shuster, Malcolm D.** *Introduction to spacecraft attitude determination.* The Johns Hopkins University, 1989;
- 2. **Baldini, M. Barni & A. Foggi; Benelli & A. Mecocci.** *A new star constellation matching algorithm for satellite attitude determination.* ESA Journal 1993, 17:185-198;
- 3. **Armstrong, R. W.; Staley, D. A.** *A survey of current solid state star tracker technology.* Journal of Astronautical Sciences 33:(4):341-352. Oct.-Dec., 1985;
- 4. **Allen.** *Astrophysics Quantities,* 523.023 A153a - 1973;
- 5. **Kosik, J. C.** *Star pattern identification aboard on inertially stabilized spacecraft.* Journal of Guidance, Control and Dynamics 14(2):230-235, Mar.- Apr., 1991;
- 6. **Strikwerda, T. E.; Fisher, H. L.; Kilgus, C. C. and Frank, L. J.** *Autonomous star identification and spacecraft attitude determination with CCD star trackers.* Proceedings of 1st ESA International Conference. ESTEC, Noordwijk, the Netherlands, 4-7 June, 1991, pp. 195-200;
- 7. **Undomkesmalee, S.; Alexander, J. W. and Tolivar, A. F.** *Stochastic star identification.* Journal of Guidance, Control and Dynamics, 17(6):1283-1287, Nov. - Dec., 1994;
- 8. **Willians, K. E.; Strikwerda, T. E.; Fisher, H. L.; Strohbehn, K. and Edwards, T. G.** *Design study: Parallel architectures for autonomous star pattern identification and tracking.* (AAS 93-012) *Spacecraft Mechanics* 1993, Vol. 82, Part I, *Advances in the Astronautical Sciences*, R. G. Melton; L. J. Wood; R. C. Thompson and S. J. Kerridge (eds.). Proceedings of AAS/AIAA Space Mechanics Meeting, Pasadena, CA, Feb. 22-24, 1993. (Univelt San Diego, CA, 1993)pp.43-58;
  - 9. **Wertz, J. R.(ed.).** *Spacecraft attitude determination and control.* D. Heidel, London, 1978. (Astrophysics and Space Sciences Library);
  - 10. **Hughes, P. C.** *Spacecraft attitude dynamics.* John-Wiley & Sons, New York, 1986;
  - 11. **Kane, T. R.; Linkins, P. W. and Levinson, D. A. (ed.).** *Spacecraft dynamics.* Mc Graw-Hill, New York, 1983;
  - 12. **Rinrott, F. P. J.** *Introductory attitude dynamics.* Spring Verlag, New York, 1989. (Mechanical Engineering Series);

- 13. **Freeman, H.** *Introduction to statistical inference*. Addison-Wesley Publishing Company, London, 1963. (Addison-Wesley Series in Statistics);
- 14. **Brownlee, K. A.** *Statistical theory and methodology in science and engineering*. John-Wiley & Sons, New York, 1967.
- 15. **Levine, Wilian S.** *The Control Handbook*. CRC Press, Inc., Boca Raton, Florida, 1996.

c) Catálogos:

- 1. *Efemérides astronômicas 1994*. Observatório Nacional do Rio de Janeiro, 1994.

e) Atlas e mapas estelares:

- 1. *Norton star atlas*.

f) Softwares:

- 1. *PC-MATLAB WITH SIMULINK 4.0*;
- 2. *Word for Windows 6.0*;
- 3. *Fortran for Windows*;
- 5. *Paint Shop Pro 1.4*;
- 6. *Excel for Windows 5.0*



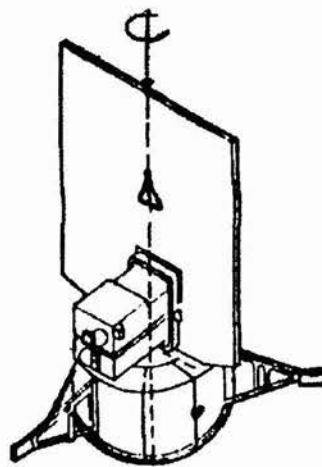
# Apêndices

## A - Tipos de estabilizações de satélite

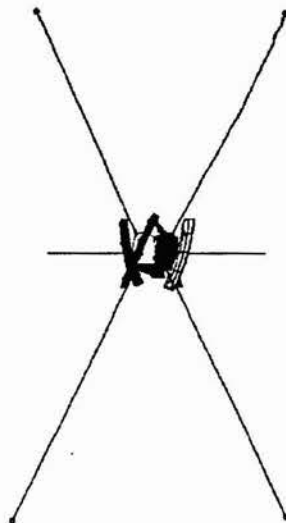
Os tipos de satélite e suas respectivas formas de estabilizações são em muito ligadas ao tipo de missão a ser desempenhada

Alguns tipos de missões são comuns devido às suas amplas aplicações e importâncias, como é o caso de satélites para telecomunicações, meteorologia, espionagem, posicionamento de veículos na Terra (gps), de recursos naturais e científicos.

Existem dois tipos básicos de estabilização de satélites, a triaxial e a rotativa (Ver fig. A1 e A2). Muitos modos são derivados da combinação destes dois, porém ainda sendo importante o modo de estabilização por gradiente gravitacional.



**Fig. A1. - Sonda OSO - 8: estabilização composta rotativa-triaxial.**



**Fig. A2. - Sonda RAE - 2: estabilização por gradiente gravitacional.**

Podemos chamar a estabilização triaxial de ativa, pois utiliza-se de sistemas de manutenção intensos, enquanto que a rotativa classifica-se como passiva por não usar nenhum sistema de monitoramento intenso, apenas a condição física da conservação do momento angular.

Satélites estabilizados triaxialmente mantêm-se fixos no espaço e são monitorados por sensores que não necessitam de nenhum movimento do satélite para funcionar. Já os satélites rotativos, mantêm o seu eixo de rotação fixo no espaço e utilizam-se de sensores que necessitam da rotação do satélite para funcionar.

Todos os tipos utilizam-se de posicionamentos de atitude com relação ao referencial inercial. Para isso, aponta-se dois versores para duas referências de interesse no caso de estabilização triaxial, ou o eixo de rotação no caso de estabilização por rotação. A atitude será o desvio do sistema do satélite com relação ao inercial.

## B- Missões, aquisição e controle de atitude e torques atuantes

São vários os tipos de missões de satélites. Cada qual utilizando-se de um tipo específico de estabilização que garanta as condições desejadas.

Satélites de comunicação utilizam órbitas geosíncronas equatoriais a 35880 Km de altitude (isto é, estacionárias acima de um ponto sobre a Terra). Há sérios problemas de superposição de órbitas de vários satélites de várias nacionalidades, uma vez que esta é a melhor órbita indicada aos fins de telecomunicação. No entanto, tais órbitas não são as únicas possibilidades, e ainda, não são adequadas as comunicações no pólos. Uma das soluções foi apresentada pelos satélites soviéticos *Molniya* que têm órbitas altamente elípticas com apogeu sobre o ex-território soviético, passando por ele devagar, e dessa forma permitindo um tempo mais amplo de operação. São utilizados vários satélites na mesma órbita, o que permite uma comunicação contínua, sempre existindo algum satélite sobre a área de interesse.

Satélites de sensoriamento remoto, destinados à observação da superfície terrestre para a avaliação de recursos naturais, podem funcionar em modo passivo (captação de luz refletida pela Terra) ou em modo ativo (captação de sinais enviados pelo satélite e refletidos pela Terra). As órbitas são heliocêntricas permitindo igual iluminação em horários iguais e facilitando a interpretação de sinais. Essas órbitas são obtidas utilizando-se perturbações orbitais para fazer com que o plano orbital rotacione, e deve ser retrógrada para que ocorra precessão.

Satélites de navegação e posicionamento constituem de uma constelação de dezoito satélites utilizados para fazer o posicionamento de qualquer coisa sobre a Terra, com órbitas entrelaçadas.

Estes são alguns exemplos dentre muitos outros utilizados em várias missões.

A missão pode ser dividida em três fases:

I - Lançamento: Atividades desde o chão até a órbita pré-determinada;

II - Acomodação: Manobras e checagem;

III - Operações: Finalidades da missão.

Alguns tópicos são fundamentalmente determinantes para o desempenho da missão e sua viabilidade. Esses tópicos englobam veículos lançadores reutilizáveis, processos de controle e atitude a bordo e avanços em comunicações com base em posicionamento gps.

Ônibus espaciais diminuem o custo da missão além de aumentar a capacidade dos satélites devido ao fato de ser possível aumentar seus tamanhos e formatos com instrumentos que antes eram problema para os foguetes.. Processos a bordo diminuem as falhas por transmissão de dados, e sistemas gps aumentam a precisão e velocidade de determinação da posição do satélite.

No entanto, há restrições quanto à avaliação de uma missão por completo. Devem ser analisados a fundo, não só os objetivos, como também as necessidades e requisitos para a realização, que sejam compatíveis com o tempo, tecnologia e orçamento disponíveis.

Podemos definir três estágios de projeto da missão. A caracterização resume e pesquisa conceitos alternativos para a aplicação e utilização de recursos. A definição de requisitos da missão estipula os níveis de funcionamento e margens de inter-relacionamento de cada elemento e a avaliação final verifica a satisfação de necessidades pelos conceitos escolhidos.

Passando a termos mais específicos da missão, um dos principais requisitos é a aquisição e manutenção de atitude para sustentar um perfeito funcionamento.

As manobras são um dos principais focos de planejamento da missão, uma vez que é a partir dela que vai se desenrolar todo o processo de uma boa atitude para um ótimo funcionamento. São em muito dependentes do tipo de estabilização do satélite, pois para cada caso deve ser executada uma manobra de aquisição em específico. A aquisição estende-se por todo o período entre a desacoplagem do veículo lançador e o estabelecimento prévio à atitude requerida, quando é feito um ajuste final com a determinação de atitude atual afim de chegar-se ao ideal.

Podemos ir de uma posição a qualquer outra através da combinação de manobras. No entanto, nem todas satisfazem algumas condições de atitude, de forma que o trajeto deve ser projetado para que o satélite não abandone uma determinada margem de atitude, o que chamamos de estratégia de controle.

Para a manutenção de uma atitude e a estabilização numa determinada posição, os torques de controle são intencionais e produzidos por sistemas de atuação, enquanto torques de distúrbio são produzidos pelo meio ou internamente indesejáveis.

Para o controle da atitude são aplicados diferentes tipos de sistemas, classificando-se conforme o tipo de seu funcionamento. Um sistema em malha aberta inclui interação humana enquanto que em malha fechada é auto-controlada. Um sistema ativo utiliza-se de recursos instalados para se reorientar, enquanto que um passivo utiliza-se de torques externos disponíveis.

A decisão de usar controle ativo, passivo ou ambos, depende do ponto da missão, requisitos de estabilidade, características da missão e tempo de resposta.

Todo o controle da atitude é baseado em torques aplicados ao satélite, sendo de três origens distintas: internos, externos e aplicados por sistemas de controle.

I - Torques internos são provenientes de enrolamentos internos que causam campo magnético, ação-reação de mecanismos internos e atrito;

II - Os torques externos são devidos ao meio ambiente atuante. Os principais torques externos a serem levados em consideração são o aerodinâmico causado pelo atrito com a alta atmosfera, por radiação solar causada por partículas emitidas pela radiação do Sol, os magnéticos causados pelo campo da Terra e por gravidade do planeta.

III - Quanto a esses dois tipos, tenta-se otimizar o projeto com a minimização de suas atuações enquanto torna-se eficiente a aplicação de torques controladores através de jatos de gás, rodas de reação e enrolamentos magnéticos.

Podemos visualizar isto lembrando que a massa de um satélite não é perfeitamente distribuída fazendo com que o centro da gravidade se desloque do centro geométrico conforme mostra a figura (Ver fig. B1):

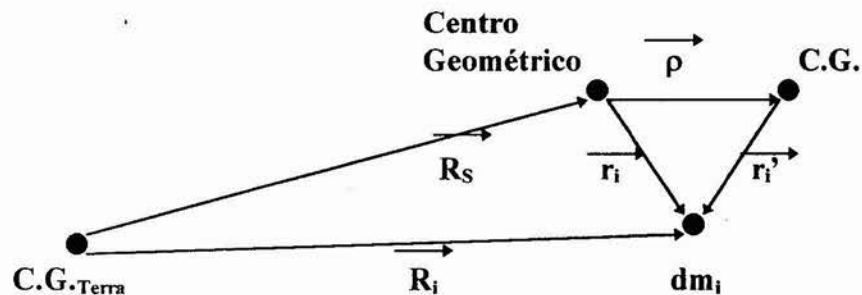


Fig. B1. - Forças atuantes num elemento de massa do satélite.

A força atuante num elemento de massa  $dm_i$  do satélite, distante  $\vec{R}_i$  do C.G. da Terra é dada por:

$$d\vec{F}_i = -G \cdot M_T \cdot \frac{dm_i \cdot \vec{R}_i}{R_i^3}$$

Essa força produz um torque no centro geométrico dado por:

$$d\vec{N}_i = \vec{r}_i \times d\vec{F}_i$$

e  $\vec{r}_i$  por soma vetorial é:  $\vec{r}_i = \vec{\rho} + \vec{r}_i'$  ;

Com relação ao C.G. da Terra:

$$\vec{R}_i = \vec{R}_S + \vec{\rho} + \vec{r}_i'$$

Substituindo na equação de torque, fazendo algumas simplificações e integrando em toda a massa:

$$\vec{N} = \frac{G \cdot M_T}{R_S^2} \cdot (\vec{R}_S \times \vec{\rho}) + \frac{3 \cdot G \cdot M_T}{R_S^3} \cdot \int_M (\vec{r}_i' \times \vec{R}_S) \cdot (\vec{r}_i' \cdot \vec{R}_S) \cdot dm_i$$

Notemos que se o C.G. coincidir com o centro geométrico,  $\rho = 0$ , e dessa forma o primeiro termo desaparece e teremos:

$$\vec{N} = 3 \cdot \frac{G \cdot M_T}{R_S^3} \cdot (\vec{R}_S \times (\vec{I} \cdot \vec{R}_S))$$

onde  $\vec{I}$  é o tensor momento de inércia do satélite.

Dessa forma, para satélites não esféricos ainda teremos torques, pois  $I \neq 0$ . No caso ideal de satélites perfeitamente esféricos,  $I = 0$ , ou  $N = 0$ .

## C- Vetores e matrizes

Uma importante ferramenta, quando falamos em termos de posicionamento e atitude, é a álgebra vetorial e matricial. Serão apresentados os conceitos que envolvem vetores, matrizes e que suportam em grande parte a matemática deste trabalho.

- Uma matriz é igual à outra se as duas tiverem a mesma ordem e os elementos correspondentes forem iguais.

- Definimos a matriz transposta como: se  $A = [a_{ij}]_{m \times n}$ , então sua transposta será  $A^T = [a_{ji}]_{n \times m}$  e é verdadeira a seguinte propriedade:  $(A^T)^T = A$ .

- Definimos a matriz adjunta como: se  $A = [a_{ij}]_{m \times n}$ , a transposta  $A^T = [a_{ji}]_{n \times m}$  e  $a_{ij}$  é um número complexo, então a matriz adjunta de A formada pelos conjugados complexos dos elementos da transposta:

$$A^+ = \overline{A^T} = [\overline{a_{ji}}]_{n \times m}$$

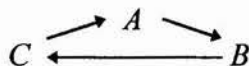
e é verdadeira a seguinte propriedade:  $(A^+)^+ = A$ .

Perceba que se todos os elementos de A forem reais, então  $A^T = A^+$ .

- O produto de matrizes não é comutativo, mesmo se as matrizes forem quadradas. O resultado não é necessariamente igual:  $A \cdot B \neq B \cdot A$ .

- Propriedade:  $(A \cdot B \cdot C \dots)^T = \dots C^T \cdot B^T \cdot A^T$ ,  $(A \cdot B)^+ = B^+ \cdot A^+$

- Definimos traço de uma matriz como a soma dos elementos da diagonal de uma matriz quadrada. O traço da multiplicação de matrizes é igual para uma permutação cíclica:  $tr(A \cdot B \cdot C) = tr(C \cdot B \cdot A)$ :



- Uma matriz  $A_{m \times n}$  é quadrada quando  $m = n$ .

- Classe K é o maior número de ordem  $m$  ou  $n$  de uma matriz, tirando as linhas e colunas nulas.

- Quanto ao determinante de uma matriz quadrada  $A_{m \times n}$ , são fornecidas algumas propriedades:

$$\det(A \cdot B) = \det A \cdot \det B;$$

$$\det(a \cdot A) = a^n \cdot \det A;$$

$$\det A^T = \det A^+;$$

$$\det A^+ = \overline{\det A};$$

Se o  $\det A \neq 0$ , então  $\det A^T \neq 0$  e  $\det A^+ \neq 0$  e ainda  $(A^T)^{-1} = (A^{-1})^T$  e  $(A^+)^{-1} = (A^{-1})^+$ .

Se  $\det A \neq 0$ ,  $\det B \neq 0$ ,  $\det C \neq 0$ , ..., então  $\det ABC \dots \neq 0$  e  $(A \cdot B \cdot C \dots)^{-1} = \dots C^{-1} \cdot B^{-1} \cdot A^{-1}$ .

O determinante de uma matriz triangular é a multiplicação dos elementos da diagonal de A.

- Matriz inversa  $A^{-1}$  é aquela que quando multiplicada pela matriz original A resulta numa matriz identidade I, podendo ser de três tipos:

I -  $K = m = n$  - inversa única:  $A \cdot A^{-1} = A^{-1} \cdot A = I$ ,  $\det A \neq 0$ . Se  $\det A = 0$ , A é chamada singular, ou seja, não tem determinante;

II -  $K = m < n$  - inversa à direita com infinitas soluções:  $A \cdot A^d = I$ ,  $A^d = A^+ \cdot (A \cdot A^+)^{-1}$ ;

III -  $K = n < m$  - inversa à esquerda com infinitas soluções:  $A^e \cdot A = I$ ,  $A^e = (A^+ \cdot A)^{-1} \cdot A^+$ .

- Matriz:

- simétrica é tal que  $A = A^T$ ;

- anti-simétrica é tal que  $A = -A^T$ ;

- heremita é tal que  $A = A^+$ ;

- ortogonal é tal que  $A^T = A^{-1}$ ,  $\therefore A^T \cdot A^{-1} = A^{-1} \cdot A^T = I$ ;

- unitária é tal que  $A^+ = A^{-1}$ ,  $\therefore A^+ \cdot A^{-1} = A^{-1} \cdot A^+ = I$ ;

- normal é tal que  $A^+ \cdot A = A \cdot A^+$

O produto de duas matrizes unitárias é uma matriz unitária, e o produto de duas matrizes ortogonais é uma matriz ortogonal.

Se A é:

- unitária, então  $|\det A|^2 = 1$ ;

- ortogonal, então  $(\det A)^2 = 1$ . Se  $\det A = 1$ , A é chamada própria, e se  $\det A = -1$ , A é chamada imprópria.

Para a transformação de matrizes, precisamos de uma multiplicação bilateral:

$A_S = B^{-1} \cdot A \cdot B$  - transformação similar;

$A_U = B^+ \cdot A \cdot B$  - transformação unitária;

$A_O = B^T \cdot A \cdot B$  - transformação ortogonal;

e ainda:  $tr A_S = tr A_U = tr A_O = tr A$ ,  $\det A_S = \det A_U = \det A_O = \det A$ ,  $A_U^+ = B^+ \cdot A^+ \cdot B$  e  $A_O^T = B^T \cdot A^T \cdot B$ .

• Vetores diferem de matrizes por terem apenas uma linha ou coluna, as vezes chamados de matriz linha ou coluna respectivamente. São comumente representados em sistemas ortonormais de coordenadas que são montados com base de versores, ou seja, vetores unitários, como na figura (Ver fig. C1):

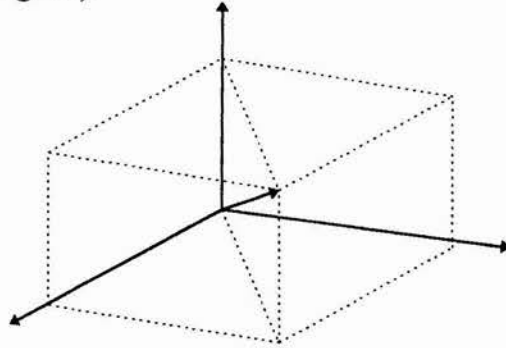


Fig. C1. - Sistema ortonormal de coordenadas.

Representamos  $\vec{r}$  através de coeficientes multiplicando os versores de direção:

$$\vec{r} = x \cdot \hat{i} + y \cdot \hat{j} + z \cdot \hat{k} \equiv (x, y, z);$$

$$\hat{i} \cdot \hat{j} = \hat{i} \cdot \hat{k} = \hat{j} \cdot \hat{k} = 0; \quad \hat{i} \times \hat{j} = \hat{k}, \quad \hat{j} \times \hat{k} = \hat{i}, \quad \hat{k} \times \hat{i} = \hat{j};$$

$$\hat{i} \cdot \hat{i} = \hat{j} \cdot \hat{j} = \hat{k} \cdot \hat{k} = 1.$$

Podemos montar uma condição do sistema:  $\hat{e}_i \cdot (\hat{e}_j \cdot \hat{e}_k) = \epsilon_{ijk}$ , para  $i, j, k = 1, 2, 3$ .

Assim:  $\epsilon_{123} = \epsilon_{231} = \epsilon_{312} = 1$ ,  $\epsilon_{132} = \epsilon_{321} = \epsilon_{213} = -1$  e demais termos nulos.

Estes parâmetros  $\epsilon$  são úteis para a definição algébrica de produto vetorial.

•  $m$  vetores são linearmente independentes quando a somatória de cada um deles multiplicados por um coeficiente  $a_i$  é zero se, e somente se, os coeficientes  $a_i$  forem nulos:

$$\sum_{i=1}^m a_i \cdot B_i = 0 \quad a_i = 0 \text{ para } i = 1, \dots, m.$$

• A multiplicação de matrizes por vetores fica:

$$A_{m \times n} \cdot \vec{V}_{n \times 1} = [(V_{n \times 1})^T \cdot (A_{m \times n})^T]^T.$$



Chamamos  $A_{m \times n} \cdot \vec{V}_{n \times 1} = \vec{y}$  e  $(V_{n \times 1})^T \cdot (A_{m \times n})^T = \vec{y}^T$ .

• Para dois vetores  $x = (a, b, \dots)_m$  e  $y = (c, d, \dots)_m$ , definimos como produto:

- escalar real:  $\vec{x} \cdot \vec{y}$ ;
- escalar complexo:  $\vec{x} \cdot \vec{y}^*$ .

O módulo é dado por:

$$|\vec{x}| = \sqrt{\vec{x} \cdot \vec{x}}$$

(ou  $|\vec{x}| = \sqrt{\vec{x} \cdot \vec{x}^*}$ , se  $x$  for complexo).

Definimos como produto externo:  $\vec{y} \cdot \vec{x}$  ou  $\vec{y}^* \cdot \vec{x}$

Definimos como produto vetorial:

$$\vec{x} \times \vec{y} = \sum_{k=1}^3 \left( \sum_{i=1}^2 \sum_{j=1}^1 x_i \cdot y_j \cdot \epsilon_{ijk} \right) \hat{e}_k, \text{ para o caso de vetores tridimensionais.}$$

Podemos também definir uma matriz  $[[\vec{x}]]$  tal que:  $[[\vec{x}]]_{ij} = \sum_{k=1}^3 \epsilon_{ijk} \cdot x_k$ ,

a qual chamamos função ômega, e multiplicá-la por  $\vec{y}$  e dessa forma:

$$\vec{x} \times \vec{y} = [[\vec{x}]] \cdot \vec{y}.$$

O produto vetorial ainda pode ser entendido como um vetor perpendicular ao plano formado pelos dois vetores que se multiplicam e que diferenciam angularmente de um ângulo  $\theta$ . Assim

$$u = \vec{x} \times \vec{y} = |\vec{x}| \cdot |\vec{y}| \cdot \text{sen}\theta,$$

onde  $\theta$  é o chamado de ângulo de rotação de  $\vec{x}$  para  $\vec{y}$  em torno de  $u$ .  $\theta$  também pode ser usado para calcular o produto escalar:

$$\vec{x} \cdot \vec{y} = |\vec{x}| \cdot |\vec{y}| \cdot \text{cos}\theta$$

A função ômega pode ser entendida para um número como a matriz que possui os elementos das posições cujo  $\epsilon = \pm 1$  iguais ao próprio número.

• Podemos ter dois sistemas ortogonais  $\epsilon$  e  $\epsilon'$  onde  $\epsilon = \{\hat{e}_1, \hat{e}_2, \hat{e}_3\}$  e  $\epsilon' = \{\hat{e}'_1, \hat{e}'_2, \hat{e}'_3\}$ , podendo representar vetores em ambos os sistemas que chamamos de base:  $x_\epsilon = \{x_1, x_2, x_3\}$  ou  $x_{\epsilon'} = \{x'_1, x'_2, x'_3\}$ .

Para relacionar  $\epsilon$  e  $\epsilon'$  obtemos uma matriz  $C = [c_{ij}]_{3 \times 3}$  chamada de matriz de mudança de base de  $\epsilon$  para  $\epsilon'$  onde:

$$\hat{e}'_i = \sum_{j=1}^3 c_{ij} \cdot \hat{e}_j$$

Para passar de  $\epsilon$  para  $\epsilon'$  temos  $C$ , e de  $\epsilon'$  para  $\epsilon$  temos  $C'$ , de forma que:

$$C \cdot C' = C' \cdot C = I, c_{ij} = \hat{e}'_i \cdot \hat{e}_j \text{ e } c'_{ij} = \hat{e}_i \cdot \hat{e}'_j \text{ e portanto } C' = C^T.$$

Como  $C' = C^T$  e  $C \cdot C' = C' \cdot C = I$ , então  $C^T = C^{-1}$  e portanto  $C$  é ortogonal.

Podemos utilizar  $C$  para passar as coordenadas de um vetor numa base  $\epsilon$  para uma  $\epsilon'$ :

$$\vec{x} = C \cdot \vec{y}.$$

Como  $C$  é ortogonal:  $(\det C)^2 = 1$ .

Assim  $C \cdot \vec{x} \cdot C \cdot \vec{y} = \vec{x} \cdot \vec{y}$  e  $(C \cdot \vec{x}) \times (C \cdot \vec{y}) = \det C \cdot C \cdot (\vec{x} \times \vec{y})$ .

Transformações ortogonais preservam o produto escalar, mas somente as próprias preservam o produto vetorial. Rotações são transformações ortogonais próprias.

Também transformações unitárias não alteram o produto escalar, ou seja, se A é unitária,

$$\vec{A} \cdot \vec{x} \cdot \vec{A} \cdot \vec{y} = \vec{x} \cdot \vec{y}.$$

• Podemos utilizar matrizes e vetores em sistemas de equações. Representamos um sistema da seguinte forma:

$$A \cdot \vec{x} = \vec{y}, \text{ onde } \vec{x} \text{ e } \vec{y} \text{ são vetores coluna.}$$

Se:

I -  $K = m = n$  - teremos solução única, pois o número de variáveis é igual ao número de equações;

II -  $K = m < n$  - número de variáveis maior que o número de equações e teremos infinitas soluções;

III -  $K = n < m$  - número de variáveis menor que o número de equações e não há soluções, mas podemos aplicar a solução com a inversa à esquerda para uma aproximação:  $x \cong A^{-1} \cdot y$ .

• Se  $A \cdot \vec{x} = \lambda \cdot \vec{x}$ , então  $\vec{x}$  é o autovetor e  $\lambda$  é o autovalor correspondente.

$\det(A - \lambda \cdot I) = 0$  é a equação característica de A.

Se  $\vec{x}$  é autovetor, então  $a \cdot \vec{x}$  também é.

Podemos montar uma matriz com todos os autovetores de A.

Pela equação de autovalor:

$$P = \begin{pmatrix} x_1^1 & \dots & x_1^n \\ x_2^1 & \dots & x_2^n \\ \vdots & & \vdots \\ x_n^1 & \dots & x_n^n \end{pmatrix} \quad A \cdot P = \begin{pmatrix} \lambda_1 \cdot x_1^1 & \dots & \lambda_n \cdot x_1^n \\ \lambda_1 \cdot x_2^1 & \dots & \lambda_n \cdot x_2^n \\ \vdots & & \vdots \\ \lambda_1 \cdot x_n^1 & \dots & \lambda_n \cdot x_n^n \end{pmatrix} = P \cdot \Lambda$$

Onde  $\Lambda$  é a matriz  $m \times n$  cujos elementos da diagonal são os  $n$  autovalores de A e os outros elementos nulos.

$\det P = 0$  se os autovetores forem linearmente independentes.

$\therefore \Lambda = P^{-1} \cdot A \cdot P$ . A é diagonalizável por transformação similar.

Se A for normal, podemos escolher vetores ortonormais, e assim P é unitária e A é diagonalizável por transformação unitária:  $\Lambda = P^+ \cdot A \cdot P$ .

Dessa forma:  $\det A = \lambda_1 \cdot \lambda_2 \dots \lambda_n$  e  $\text{tr} A = \sum_{i=1}^n \lambda_i$

Os autovalores de uma matriz real ou são reais ou ocorrem em pares conjugados complexos.

$$\text{Se } A \cdot \vec{x} = \lambda \cdot \vec{x}, \text{ então } A \cdot \overline{\vec{x}} = \overline{\lambda} \cdot \overline{\vec{x}}$$

• Função matriz é definida por  $f(A) = M$  onde os elementos de M nem sempre são funções  $f$  dos elementos de A.

$$f(A) = \sum_{n=0}^{\infty} a_n \cdot M^n \quad \text{ou} \quad f(M) = P \cdot \left( \sum_{n=0}^{\infty} a_n \cdot \Lambda^n \right) \cdot P^{-1}$$

- A normalização de vetores ocorre da seguinte forma:

$$\hat{V} = \frac{\vec{V}}{|\vec{V}|} \quad \text{onde} \quad |\vec{V}| = \sqrt{V_x^2 + V_y^2 + V_z^2}$$

- Em cálculo vetorial, um vetor  $\vec{x} = (x_1, \dots, x_n)$  pode ser utilizado por uma função  $\phi(x_1, x_2, \dots, x_n)$  multi-argumentos onde as derivadas parciais com relação a  $\vec{x}$  compõem o vetor gradiente de  $\phi$ :

$$\frac{\partial \phi}{\partial \vec{x}} = \left( \frac{\partial \phi}{\partial x_1}, \frac{\partial \phi}{\partial x_2}, \dots, \frac{\partial \phi}{\partial x_n} \right)$$

Definimos operador gradiente como:

$$\frac{\partial}{\partial \vec{x}} = \left( \frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \dots, \frac{\partial}{\partial x_n} \right)$$

O produto escalar do operador por um vetor:

$$\frac{\partial}{\partial \vec{x}} \cdot \vec{y} = \sum_{i=1}^n \frac{\partial y_i}{\partial x_i}$$

é chamado de divergente de  $\vec{y}$ .

## D- Cálculo da área da região observada e sua densidade de estrelas

Para o cálculo das probabilidades de ocorrência dos diversos tipos de identificação de estrelas com a distribuição de **Poisson** é necessário que tenhamos a densidade de estrelas  $\rho$  definida como:

$$\rho = \frac{\text{N}^\circ \text{ de estrelas no campo}}{\text{Área do campo}}$$

Para isso, é preciso o cálculo da área do campo que se processa através de integral. Através da figura (Ver fig. D1) observa-se um elemento de área na esfera associado às suas longitudes e latitudes (ou declinações e ascensões) e em seguida a codificação em coordenadas esféricas:

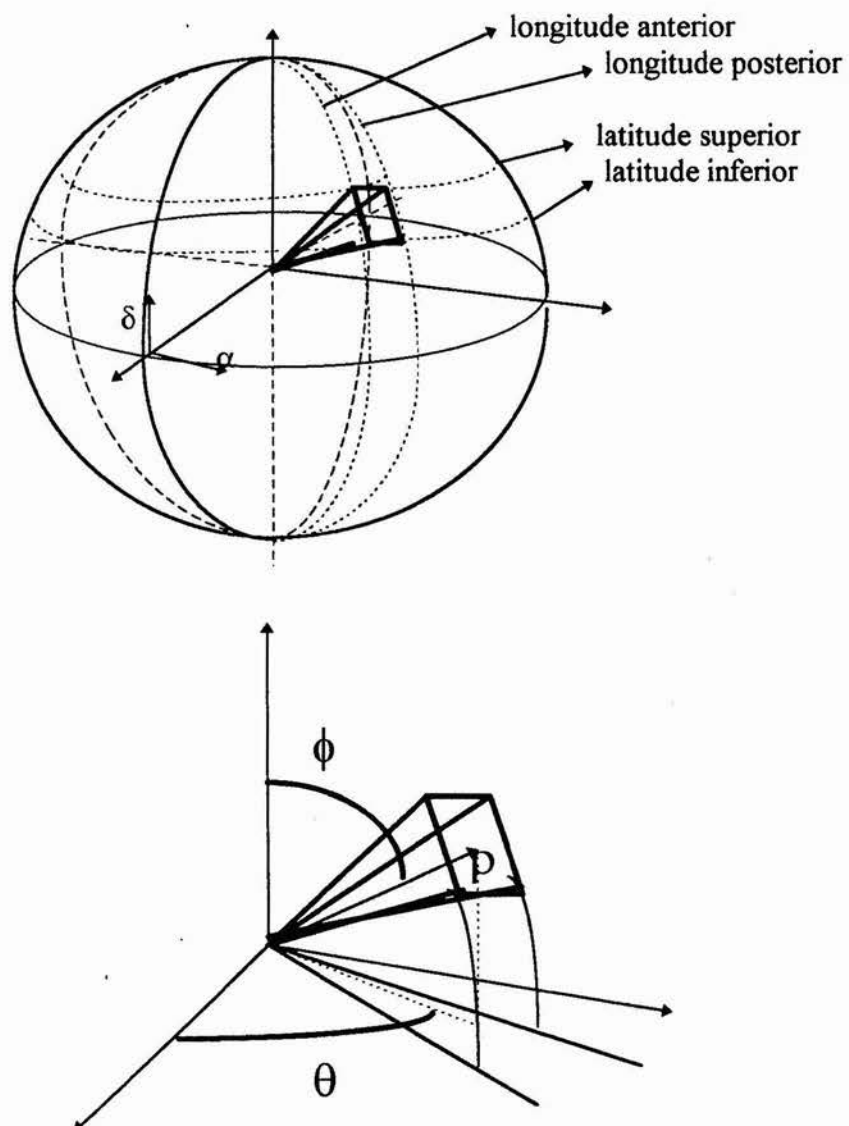


Fig. D1. - Elemento de área para a integração do cálculo de área da região

A área deste elemento infinitesimal pode ser aproximada pela área de um quadrado a uma distância  $\rho$  do centro da esfera, assim:

$$\Delta A = \rho \Delta\phi \cdot \rho \operatorname{sen}\phi \Delta\theta$$

Integrando dentro dos limites desejados:

$$A = \int_{\theta_1}^{\theta_2} \int_{\phi_1}^{\phi_2} \rho^2 \operatorname{sen}\phi \, d\phi \, d\theta$$

Como consideramos o raio da esfera celeste unitário  $\rho = 1$ , integrando:

$$A = (\cos\phi_1 - \cos\phi_2) \cdot (\theta_2 - \theta_1)$$

E assim transformamos as coordenadas de declinação e ascensão em coordenadas esféricas para que possamos aplicar a fórmula de área obtida:

$$\theta = \alpha_h / 15, \quad \phi = 90 - \delta$$

## E- Magnitudes

A magnitude de uma estrela é definida como o brilho que ela nos apresenta. No entanto, quanto mais perto, maior o brilho que percebemos, e dessa forma uma estrela com um brilho fraco pode apresentar-se mais brilhante que outra com um brilho mais forte, porém mais distante.

Assim convencionou-se magnitude absoluta como o brilho que um determinado objeto apresentaria se estivesse a 1 *parsec* ou 3,26 *anos-luz* de distância da Terra. A magnitude aumenta conforme o brilho enfraquece, sendo realizada numa escala logarítmica.

Porém, em termos práticos, a magnitude absoluta em nada ajuda. Usa-se o que chamamos de magnitude aparente, que inter-relaciona os brilhos de objetos celestes vistos da Terra.

No entanto, a resposta do sensor é diferente em termos de magnitude, pois é utilizada a magnitude instrumental. A magnitude instrumental é definida em termos da magnitude visual aparente  $v$  e da magnitude azul  $B$  que ocorre na faixa de 350 - 550 *nm* de comprimento de onda luminoso. A magnitude instrumental precisa ser calculada de forma a gerar um arquivo de estrelas não muito extenso, mas que inclua todas as que podem ser bem distinguidas. Também é necessária para o modelamento da resposta.

A magnitude instrumental é definida por:  $m_i = h \cdot v + (1 - h) \cdot B$ , onde  $v$  é a magnitude visual aparente,  $B$  é a azul e  $h$  é um fator empírico e é adaptado para que a distribuição instrumental melhor se encaixe à distribuição teórica de magnitudes estelares, ou até que se obtenha a máxima nitidez.

O sinal de saída  $I$  é relacionado com a magnitude instrumental da seguinte forma:

$$\text{Log } I = c_i + d_i \cdot m_i,$$

onde  $c_i$  e  $d_i$  são coeficientes dependentes do sensor e são geralmente fornecidos em sua calibração.

## F- Relações empíricas para o cálculo de $\phi$ e $\lambda$ em coordenadas da matriz eletrônica

Os coeficientes  $\phi$  e  $\lambda$  podem ser dados por expressões que levam em conta as distorções de diversas naturezas.

O coeficiente  $\phi$  é dado por:

$$\phi = c_0 + c_1 \cdot u + c_2 \cdot v + c_3 \cdot u^2 + c_4 \cdot u \cdot v + c_5 \cdot v^2 + c_6 \cdot u^3 + c_7 \cdot u^2 \cdot v + c_8 \cdot u \cdot v^2 + c_9 \cdot v^3$$

e  $\lambda$  é dado por:

$$\lambda = d_0 + d_1 \cdot u + d_2 \cdot v + d_3 \cdot u^2 + d_4 \cdot u \cdot v + d_5 \cdot v^2 + d_6 \cdot u^3 + d_7 \cdot u^2 \cdot v + d_8 \cdot u \cdot v^2 + d_9 \cdot v^3$$

onde  $u = U/f$  e  $v = V/f$ , sendo  $U$  a abcissa,  $V$  a ordenada,  $f$  a distância focal da lente e os coeficientes  $c_i$  e  $d_i$  são dependentes da temperatura.

## G- Tipos de representação de atitude e peculiaridades matemáticas

São muitas as formas de se representar uma atitude, no entanto, nos deteremos aos principais métodos que se consagram por qualidades relevantes que apresentam.

- Ângulos de Euler:

Se considerarmos três rotações consecutivas, uma matriz de rotação pode ser decomposta numa multiplicação três matrizes de rotação, uma para cada rotação do eixo em vigor.

Podemos representar da seguinte forma:

$$R(\hat{n}_1, \hat{n}_2', \hat{n}_3'', \theta_1, \theta_2, \theta_3) = R(\hat{n}_3'', \theta_3) \cdot R(\hat{n}_2', \theta_2) \cdot R(\hat{n}_1, \theta_1).$$

A rotação é sempre tomada em relação ao sistema em vigor, e não ao original.

$\hat{n}_1$ ,  $\hat{n}_2'$  e  $\hat{n}_3''$  são unitários e chamados eixos de Euler, enquanto  $\theta_1$ ,  $\theta_2$  e  $\theta_3$ , mais conhecidos por  $\varphi$ ,  $\theta$  e  $\psi$  respectivamente, são os ângulos de Euler que representam as rotações 1, 2 e 3 nos respectivos eixos citados. Deve-se ter  $n_1 \neq n_2'$  e  $n_2' \neq n_3''$ . São doze as combinações de formas de rotação, sendo as mais conhecidas a 313 e 321, assim:

$$R_{313}(\varphi, \theta, \psi) = R_3 \cdot R_1 \cdot R_3 \text{ e } R_{321}(\varphi, \theta, \psi) = R_3 \cdot R_2 \cdot R_1.$$

Qualquer matriz de rotação pode ser escrita em qualquer um dos arranjos dos ângulos de Euler. Porém, há problemas de ambigüidade. Em arranjos simétricos:

$R_{313}(\varphi, \theta, \psi) = R_{313}(\varphi + \pi, -\theta, \psi - \pi)$  e devemos ter que  $0 \leq \varphi < 2\pi$ ,  $0 \leq \theta \leq \pi$  e  $0 \leq \psi \leq \pi$   
Em arranjos assimétricos:

$$R_{321}(\varphi, \theta, \psi) = R_{321}(\varphi + \pi, \pi - \theta, \psi - \pi)$$

e devemos ter que  $0 \leq \varphi < 2\pi$ ,  $-\pi/2 \leq \theta \leq \pi/2$  e  $0 \leq \psi \leq \pi$ .

Para a determinação dos ângulos de Euler a partir de uma matriz de rotação dada, tomamos um dos modos de rotação, por exemplo 313, e determinamos a matriz  $R_{313}$  e igualamos aos termos da matriz dada.

Para determinarmos posicionamentos em atitude, algumas configurações de movimento são críticas, pois com movimentos infinitesimais em torno de alguns eixos, não garantem movimentos infinitesimais em torno de outros. Assim são necessários outros arranjos para a representação do movimento, porém que dão problemas em outras posições. Assim, para representar perfeitamente um sistema, são necessários no mínimo dois sistemas diferentes.

- Matriz de rotação:

Uma matriz de rotação nada mais é, que uma forma de escrever um vetor com coordenadas numa base  $\varepsilon$  para uma outra base  $\varepsilon'$ .

Suponhamos  $\varepsilon = \{\hat{e}_1, \hat{e}_2, \hat{e}_3\}$  e  $\varepsilon' = \{\hat{e}_1', \hat{e}_2', \hat{e}_3'\}$ .

Para simplificar, mostraremos apenas rotação bi-dimensional como na figura abaixo (Ver fig. G1):

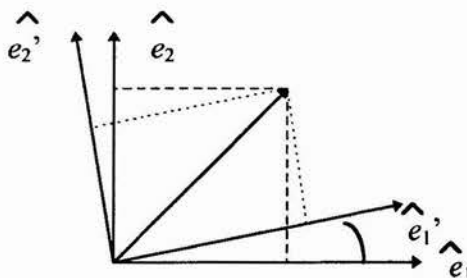


Fig. G1. - Rotação bi-dimensional de mudança de base.



$$\begin{pmatrix} \hat{e}_1' \\ \hat{e}_2' \\ \hat{e}_3' \end{pmatrix}_\varepsilon = \begin{pmatrix} \cos\theta & \text{sen}\theta & 0 \\ -\text{sen}\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \hat{e}_1 \\ \hat{e}_2 \\ \hat{e}_3 \end{pmatrix}_{\varepsilon'}$$

↓  
R

Dessa forma, R é a matriz de rotação de  $\varepsilon$  para  $\varepsilon'$ , e qualquer vetor que seja escrito como combinação linear dos elementos da base  $\varepsilon$ , pode ser escrito como combinação linear dos elementos da base  $\varepsilon'$  através da multiplicação com a matriz de rotação R.

• Eixo e ângulo de rotação:

Se temos um vetor  $\vec{V}$  qualquer e um eixo de rotação  $\hat{n}$ ,  $\vec{V}$  pode ser escrito em função de  $\hat{n}$  (Ver fig. G2):

$$\vec{V} = \underbrace{(\hat{n} \cdot \vec{V}) \cdot \hat{n}}_{\text{componente paralela a } \hat{n}} - \underbrace{\hat{n} \times (\hat{n} \times \vec{V})}_{\text{componente perpendicular a } \hat{n}}$$

Apenas a componente perpendicular a  $\hat{n}$  será afetada pela rotação. Se rotacionarmos o sistema com um ângulo  $\theta$ , o vetor no novo sistema será:

$$\vec{V}' = (\hat{n} \cdot \vec{V}) \cdot \hat{n} - \text{sen}\theta \cdot \hat{n} \times \vec{V} - \cos\theta \cdot \hat{n} \times (\hat{n} \times \vec{V})$$

que é chamada de fórmula de Euler.

Como  $\hat{n} \times (\hat{n} \times \vec{V}) = (\hat{n} \cdot \vec{V}) \cdot \hat{n} - \vec{V}$ ,  $(\hat{n} \cdot \vec{V}) \cdot \hat{n} = \hat{n} \cdot \hat{n}^T \cdot \vec{V}$  e  $\vec{V}' = R(\hat{n}, \theta) \cdot \vec{V}$  (mudança de coordenadas), podemos achar a matriz rotação e que pode ser escrita como:

$$R(\hat{n}, \theta) = I + \text{sen}\theta \cdot [[\hat{n}]] + (1 - \cos\theta) \cdot [[\hat{n}]]^2$$

e que tem as propriedades:  $R^{-1} = R^T = R(-\hat{n}, \theta) = R(\hat{n}, -\theta)$ , pois é ortogonal.

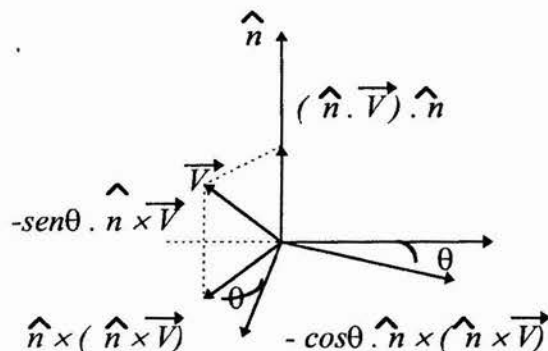


Fig. G2. - Representação de um vetor em ângulo e eixo de rotação.

## H- Geometria esférica

Apresentamos aqui alguns tópicos básicos resumidamente em geometria esférica aplicada.

Podemos encarar a geometria esférica como uma generalização, onde em particular, quando tratamos localmente e infinitesimalmente, tornamos à geometria plana.

- Um único grande círculo é determinado por dois pontos que não se distanciam de  $180^\circ$  na esfera, e cujo centro coincide com o centro da esfera. Todos os grandes círculos de uma esfera interceptam-se. Qualquer figura pode ser projetada de um lado oposto de um grande círculo que por ela passa, dando origem a uma imagem reversa.

- Em geometria esférica, a menor distância entre dois pontos é um arco de um grande círculo.

- Polígonos esféricos são construídos de arcos de grandes círculos.

- A soma dos ângulos internos de um triângulo excede  $180^\circ$ , e o excesso é proporcional à área do mesmo (Ver fig. H1).

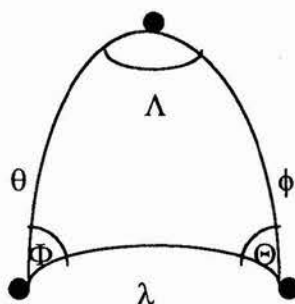


Fig. H1. - Triângulo esférico.

Algumas relações importantes do triângulo esférico:

Lei dos senos:  $\frac{\text{sen}\phi}{\text{sen}\Phi} = \frac{\text{sen}\theta}{\text{sen}\Theta} = \frac{\text{sen}\lambda}{\text{sen}\Lambda}$

Lei dos cossenos para os lados:  $\cos\theta = \cos\lambda \cdot \cos\phi + \text{sen}\lambda \cdot \text{sen}\phi \cdot \cos\Theta$

Lei dos cossenos para os ângulos:  $\cos\Theta = -\cos\Lambda \cdot \cos\Phi + \text{sen}\Lambda \cdot \text{sen}\Phi \cdot \cos\theta$

Um triângulo esférico é reto quando um dos ângulos for  $90^\circ$ , e quadrantal quando um dos lados for  $\frac{1}{4}$  de um grande círculo.

Quando ocorre triângulo esférico reto ou quadrantal, as leis trigonométricas simplificam-se.

- O estudo por geometria esférica é mais exato para a atitude, que por geometria plana aproximada. Um importante ponto a ser lembrado, é quando promove-se a representação plana de uma esfera. Quanto mais uma figura projetada estiver perto dos pólos, maior será a distorção sofrida, enquanto que no equador quase não há distorção.

# I- Listagem de testes realizados para o *Software*

## Constelação: Crucis - requisitos 1 (testes iniciais) Precisão do sensor estelar: 3'

Sensor de pré-estimativa - Precisão: 1 <sup>o</sup>									
Tipos de identificação				Rotações (g,m,s)			Número de estrelas		Raio(g)
Correta	Errônea	Ambígua	Não identificação	Direção do sensor	Perpendicular ao sensor	Combinada	Vistas pelo sensor	Identificadas	////////// //
3	0	0	0	0,37,45	0,1,39	0,37,47	3	3	2.1100
3	0	0	0	0,17,57	0,2,0	0,18,4	3	3	2.1100
3	0	1	0	0,55,57	0,1,22	0,55,58	4	3	2.1100
3	0	1	0	0,1,22	0,2,41	0,3,11	4	3	2.1100
2	0	1	1	0,55,36	0,53,36	1,17,14	4	2	2.1100
2	0	1	0	0,4,28	0,3,29	0,5,40	3	2	2.1100
3	0	0	0	0,28,57	0,1,20	0,28,59	3	3	2.1100
2	0	0	0	-1,56,59	0,0,28	1,16,59	2	2	2.1100
2	0	0	0	0,29,14	0,2,34	0,29,21	2	2	2.1100
2	0	1	0	0,23,30	0,1,16	0,23,32	3	2	2.1100

Sensor de pré-estimativa - Precisão: 2 <sup>o</sup>									
Tipos de identificação				Rotações (g,m,s)			Número de estrelas		Raio(g)
Correta	Errônea	Ambígua	Não identificação	Direção do sensor	Perpendicular ao sensor	Combinada	Vistas pelo sensor	Identificadas	////////// //
1	0	4	0	-	-	-	5	1	3.4400
1	0	2	0	-	-	-	3	1	3.4400
0	0	3	0	-	-	-	3	0	3.4400
1	0	2	0	-	-	-	3	1	3.4400
0	1	3	0	-	-	-	4	1	3.4400
2	0	3	0	0,3,45	0,4,11	0,5,37	5	2	3.4400
1	0	2	0	-	-	-	3	1	3.4400
1	0	2	0	-	-	-	3	1	3.4400
0	1	3	0	-	-	-	4	1	3.4400
1	0	2	0	-	-	-	3	1	3.4400

## Constelação: Orionis Precisão do sensor estelar: 3'

Sensor de pré-estimativa - Precisão: 1 <sup>o</sup>									
Tipos de identificação				Rotações (g,m,s)			Número de estrelas		Raio(g)
Correta	Errônea	Ambígua	Não identificação	Direção do sensor	Perpendicular ao sensor	Combinada	Vistas pelo sensor	Identificadas	////////// //
0	0	3	0	-	-	-	3	0	2.1000
2	0	3	0	0,41,15	0,0,31	0,41,15	5	2	2.1000
1	0	3	0	-	-	-	4	1	2.1000
2	0	2	0	2,37,27	0,3,30	2,37,29	4	2	2.1000
2	0	0	0	0,23,50	0,1,14	0,23,52	2	2	2.1000
1	0	2	0	-	-	-	3	1	2.1000
0	0	0	2	-	-	-	2	0	2.1000
1	0	1	0	-	-	-	2	1	2.1000
0	1	0	4	-	-	-	5	1	2.1000
0	0	1	3	-	-	-	4	0	2.1000

Sensor de pré-estimativa - Precisão: 2 <sup>o</sup>									
Tipos de identificação				Rotações (g,m,s)			Número de estrelas		Raio(g)
Correta	Errônea	Ambígua	Não identificação	Direção do sensor	Perpendicular ao sensor	Combinada	Vistas pelo sensor	Identificadas	////////// //
2	0	1	0	0,3,26	0,3,21	0,4,48	3	2	3.4200
1	0	4	1	-	-	-	6	1	3.4200
1	0	2	0	-	-	-	3	1	3.4200
1	0	3	0	-	-	-	4	1	3.4200
1	0	2	0	-	-	-	3	1	3.4200
2	0	0	0	-1,30,37	0,9,0	1,31,4	2	2	3.4200
0	0	3	0	-	-	-	3	0	3.4200
0	1	0	3	-	-	-	4	1	3.4200
0	0	4	0	-	-	-	4	0	3.4200
2	0	1	0	1,18,5	0,5,15	1,18,16	3	2	3.4200

## Constelação: Pegasus Precisão do sensor estelar: 3'

Sensor de pré-estimativa - Precisão: 1 <sup>o</sup>									
Tipos de identificação				Rotações (g,m,s)			Número de estrelas		Raio(g)
Correta	Errônea	Ambígua	Não identificação	Direção do sensor	Perpendicular ao sensor	Combinada	Vistas pelo sensor	Identificadas	////////// //
0	0	0	2	-	-	-	2	0	2.1600
2	0	0	0	0,7,47	0,4,52	0,9,11	2	2	2.1600
3	0	0	0	0,23,24	0,3,58	0,23,44	3	3	2.1600
3	0	0	0	0,1824	0,1,18	0,18,27	3	3	2.1600
0	0	0	3	-	-	-	3	0	2.1600
2	0	0	0	0,24,20	0,2,60	0,24,31	2	2	2.1600
0	0	0	2	-	-	-	2	0	2.1600
0	0	2	0	-	-	-	2	0	2.1600
0	0	0	2	-	-	-	2	0	2.1600
2	0	0	0	0,18,59	0,2,28	0,19,9	2	2	2.1600

Sensor de pré-estimativa - Precisão: 2 <sup>o</sup>									
Tipos de identificação				Rotações (g,m,s)			Número de estrelas		Raio(g)
Correta	Errônea	Ambígua	Não identificação	Direção do sensor	Perpendicular ao sensor	Combinada	Vistas pelo sensor	Identificadas	////////// //
0	1	0	1	-	-	-	2	1	3.6800
2	0	1	0	0,20,51	0,3,32	0,21,9	3	2	3.6800
2	0	0	0	0,3,35	0,4,12	0,5,31	2	2	3.6800
3	0	0	0	0,1832	0,0,24	0,18,32	3	3	3.6800
2	0	0	0	0,30,11	0,2,14	0,30,13	2	2	3.6800
0	1	1	0	-	-	-	2	1	3.6800
3	0	0	0	0,7,53	0,2,40	0,8,19	3	3	3.6800
1	0	2	0	-	-	-	3	1	3.6800
2	0	0	0	0,5,38	0,4,11	0,7,1	2	2	3.6800
3	0	0	0	0,17,46	0,1,20	0,17,49	3	3	3.6800

## Constelação: Scorpii Precisão do sensor estelar: 3'

Sensor de pré-estimativa - Precisão: 1 <sup>o</sup>									
Tipos de identificação				Rotações (g,m,s)			Número de estrelas		Raio(g)
Correta	Errônea	Ambígua	Não identificação	Direção do sensor	Perpendicular ao sensor	Combinada	Vistas pelo sensor	Identificadas	////////// //
2	0	0	0	1,46,40	0,0,43	1,46,40	2	2	2.1700
1	0	1	0	-	-	-	2	1	2.1700
2	0	0	0	2,10,39	0,3,45	3,10,42	2	2	2.1700
3	0	0	0	0,2,18	0,0,42	0,0,24	3	3	2.1700
3	0	0	0	0,26,47	0,2,7	0,26,52	3	3	2.1700
2	0	1	0	0,45,43	0,2,2	0,45,46	3	2	2.1700
1	0	1	0	-	-	-	2	1	2.1700
2	0	1	0	1,26,3	0,4,10	1,26,10	3	2	2.1700
0	0	0	2	-	-	-	2	0	2.1700
0	0	1	2	-	-	-	3	0	2.1700

Sensor de pré-estimativa - Precisão: 2 <sup>o</sup>									
Tipos de identificação				Rotações (g,m,s)			Número de estrelas		Raio(g)
Correta	Errônea	Ambígua	Não identificação	Direção do sensor	Perpendicular ao sensor	Combinada	Vistas pelo sensor	Identificadas	////////// //
1	0	2	0	-	-	-	3	1	3.7100
0	0	2	0	-	-	-	2	0	3.7100
0	0	3	0	-	-	-	3	0	3.7100
2	0	0	0	0,8,49	0,1,43	0,8,59	2	2	3.7100
1	0	2	0	-	-	-	3	1	3.7100
0	0	2	0	-	-	-	2	0	3.7100
0	0	3	0	-	-	-	3	0	3.7100
1	0	1	0	-	-	-	2	1	3.7100
0	0	3	0	-	-	-	3	3	3.7100
0	1	1	0	-	-	-	2	1	3.7100

## Constelação: Tauri

### Precisão do sensor estelar: 3'

Sensor de pré-estimativa - Precisão: 1 <sup>o</sup>									
Tipos de identificação				Rotações (g,m,s)			Número de estrelas		Raio(g)
Correta	Errônea	Ambígua	Não identificação	Direção do sensor	Perpendicular ao sensor	Combinada	Vistas pelo sensor	Identificadas	////////// //
1	1	2	0	-21,38,18	1,1,16	27,39,45	4	2	2.070
2	0	3	0	0,2,28	0,2,48	0,3,44	5	2	2.070
0	0	0	2	-	-	-	2	0	2.070
1	0	1	0	-	-	-	2	1	2.070
3	0	2	0	0,5,12	0,3,15	0,6,8	5	3	2.070
0	2	0	0	-14,31,15	5,15,34	15,26,38	2	2	2.070
2	0	2	0	-1,14,34	0,3,19	1,14,38	4	2	2.070
2	0	3	0	1,52,32	0,2,32	1,52,34	5	2	2.070
1	0	1	0	-	-	-	2	1	2.070
1	0	1	1	-	-	-	3	1	2.070

Sensor de pré-estimativa - Precisão: 2 <sup>o</sup>									
Tipos de identificação				Rotações (g,m,s)			Número de estrelas		Raio(g)
Correta	Errônea	Ambígua	Não identificação	Direção do sensor	Perpendicular ao sensor	Combinada	Vistas pelo sensor	Identificadas	////////// //
1	0	2	0	-	-	-	3	1	3.2800
1	0	4	0	-	-	-	5	1	3.2800
2	0	0	0	0,57,58	0,2,1	0,58,0	2	2	3.2800
0	0	4	0	-	-	-	4	0	3.2800
1	0	3	0	-	-	-	4	1	3.2800
0	0	2	0	-	-	-	2	0	3.2800
2	0	3	0	0,46,34	0,2,11	0,46,37	5	2	3.2800
1	1	1	0	1,43,18	0,59,12	1,59,4	3	2	3.2800
0	3	0	0	19,12,31	6,10,37	20,10,39	3	3	3.2800
0	0	3	0	-	-	-	3	0	3.2800

## Constelação: Crucis - requisitos 2 (requisitos ótimos)

### Precisão do sensor estelar: 3'

Sensor de pré-estimativa - Precisão: 1 <sup>o</sup>									
Tipos de identificação				Rotações (g,m,s)			Número de estrelas		Raio(g)
Correta	Errônea	Ambígua	Não identificação	Direção do sensor	Perpendicular ao sensor	Combinada	Vistas pelo sensor	Identificadas	////////// //
2	0	0	0	-1,6,38	0,4,1	1,6,45	2	2	1.8200
1	0	0	2	-	-	-	3	1	1.8200
3	0	0	0	0,22,38	0,2,5	0,22,44	3	3	1.8200
3	0	0	0	0,0,7	0,1,50	0,1,50	3	3	1.8200
1	0	1	0	-	-	-	2	1	1.8200
3	0	1	0	-2,18,38	0,4,53	2,18,43	4	3	1.8200
1	0	0	1	-	-	-	2	1	1.8200
2	0	1	0	0,32,24	0,1,26	0,32,26	3	2	1.8200
4	0	0	0	0,31,36	0,1,17	0,31,38	4	4	1.8200
1	0	1	0	-	-	-	2	1	1.8200

Sensor de pré-estimativa - Precisão: 2 <sup>o</sup>									
Tipos de identificação				Rotações (g,m,s)			Número de estrelas		Raio(g)
Correta	Errônea	Ambígua	Não identificação	Direção do sensor	Perpendicular ao sensor	Combinada	Vistas pelo sensor	Identificadas	////////// //
3	0	1	0	0,56,21	0,2,56	0,56,26	4	3	2.9800
0	1	0	1	-	-	-	2	1	2.9800
1	0	2	0	-	-	-	3	1	2.9800
0	3	1	1	-3,4,34	5,18,20	0,7,58	5	3	2.9800
2	0	2	0	0,17,13	0,2,9	0,17,21	4	2	2.9800
2	0	0	0	0,30,25	0,4,3	0,30,41	2	2	2.9800
2	0	1	0	-1,12,19	0,3,25	1,13,4	3	2	2.9800
1	0	2	0	-	-	-	3	1	2.9800
1	0	2	0	-	-	-	3	1	2.9800
2	0	1	0	0,28,27	0,3,10	0,28,38	3	2	2.9800

## J- Listagem de rotinas

As rotinas abaixo encontram-se em ordem alfabética:

### Aneu321.m

```
function eulers = aneu321(A)

echo off
% function eulers = aneu321(A)
% aneu321 - retorna os ângulos de Euler fi, teta e psi em graus para
% rotação 3-2-1 para uma dada matriz de rotação A.

fi = atan2(A(1,2),A(1,1));
teta = -asin(A(1,3));
psi = atan2(A(2,3),A(3,3));
fi = degree(fi);
teta = degree(teta);
psi = degree(psi);
eulers = [fi teta psi];
return
```

### Aneuler.m

```
function eulers = aneuler(cregalfa,credelt,gama)

echo off
% function eulers = aneuler(cregalfa,credelt,gama)
% aneuler - retorna os ângulos de Euler correspondentes à rotação
% 321 a partir do referencial inercial para as coordenadas do centro
% da atitude.

fi=cregalfa*15;
teta=-credelt;
psi=gama;
eulers=[fi teta psi];
return
```

### Angr.m

```
function angl = angr(ang)

echo off
% function angl = angr(a)
% angr - transforma os valores dos ângulos de graus,minutos e segundos
% para valores reais em graus.

if(ang(1)==0)
    if(ang(2)==0)
```

```

if(ang(3)==0)
    sinal=0;
else
    sinal=ang(3)/abs(ang(3));
end
else
    sinal=ang(2)/abs(ang(2));
end
else
    sinal=ang(1)/abs(ang(1));
end
angl = abs(ang(3))/60+abs(ang(2));
angl = angl/60+abs(ang(1));
angl =angl*sinal;
return

```

### Areareg.m

```

function Ar=areareg(lim)

echo off
% function Ar=areareg(lim)
% areareg - calcula a área da região esférica do
% céu que está sendo observada.

alfa=lim(:,1);
teta=alfa*15;
teta=rad(teta);
delta=lim(:,2);
fi(1)=90-delta(2);
fi(2)=90-delta(1);
fi=rad(fi);
Ar=(cos(fi(1))-cos(fi(2)))*(teta(2)-teta(1));
return

```

### Asdecls.m

```

function ads=asdecls(MWien)

echo off
% function ads = asdecls(MWien)
% asdecl - converte as coordenadas das estrelas de coordenadas
% retangulares contidas na matriz MWien 3*n para ascensão reta e
% declinação contidas na matriz ads n*2.

as = vatanqc(MWien');
dec = asin(MWien(3,:))';
dec = degree(dec);
as = degree(as)/15;
s=size(as);

```

```

s=s(1);
for i=1:s;
    if(as(i)==24)
        as(i)=0;
    end
end
ads = [as dec];
return

```

### Atanqc.m

```

function ang = atanqc(x,y)

echo off
% function ang = atanqc(x,y)
% atanqc - retorna o valor do ângulo formado pelo vetor
% com componentes x e y e o eixo x, em rad.

ya=abs(y);
xa=abs(x);
tga=ya/xa;
teta=atan(tga);
if(y>0)
    if(x<0)
        teta=pi-teta;
    end
else
    if(x<0)
        teta=pi+teta;
    end
    if(x>0)
        teta=2*pi-teta;
    end
end
ang = teta;
return

```

### Atitud.m

```

% atitud - calcula a atitude de um satélite através de
% identificação de estrelas vistas por um sensor ccd
% pelo método de determinação de atitude determinístico
% triaxial otimizado.

figure(1);figure(2);figure(3);figure(4);figure(5);figure(6);
figure(1);
resp=0;
resp1=0;
resp2='s';
while (resp==0)

```



```

if(resp1==0)
    [eulers,adsc,ro]=simul;
else
    R=idreg(eulers);
    figure(1)
    plotreg(R);
    sensplo(eulers);
    if(resp2=='e')
        senspl1(eulerse);
    end
end
if(resp2=='s')
    [eulerse,sigma]=senspre(eulers);
    [coseret,prec]=sensor(adsc);
end
[adat,raio,resp]=softind(coseret,prec,eulerse,ro,sigma);
if(resp==0)
    disp('Identificação fracassada. Processo reinicializando...');
    disp('Diminua raio de precisão da estrela ou. ');
    disp('aumente precisão dos sensores');
    resp2=input('Deseja escolher outro raio(e),simular(s) novamente ou sair(q)?!','s');
    disp("");
    if(resp2=='q')
        disp('Cálculo de atitude fracassada!!');
        return
    end
    resp1=1;
end
end
A321=softat(coseret,adat,prec);
softana(A321,eulerse,adat,raio,eulers);

```

### Atreal.m

```

function [dalfa,ddelta,gama] = atreal(R)

echo off
% function [dalfa,ddelta,gama] = atreal(R)
% atreal - retorna deslocamentos do satélite com relação ao
% centro da região observada.

resp=input('Deseja gerar os desvios aleatoriamente?(s/n):','s');
if(resp=='s')
    i=1.1;
    while(i>1)
        i=randn;
    end
    dalfa =i*4.3/15;
    i=1.1;
    while(i>1)

```

```

    i=randn;
end
ddelta =i*4.3;
i=1.1;
gama =rand*360;
else
disp('Posicione na região a seguir o centro do campo de visada:')
subplot(211)
[alfa,delta]=ginput(1);
alfa=-alfa;
alfac=(R(1,1)+R(2,1))/2;
deltac=(R(1,2)+R(2,2))/2;
dalfa=alfa-alfac;
ddelta=delta-deltac;
gama=input('desvio angular(graus):');
end
return

```

### Atvis.m

```

function [eulers,ad] =atvis(R)

echo off
% function [eulers,ad] =atvis(R)
% atvis - retorna as coord. das estrelas da região R contidas
% num campo de visada de 8x8 graus do sensor com a atitude
% 321 ditada pelos ângulos eulers.

j=1;
while(j==1)
figure(1)
plotreg(R);
[dalfa,ddelta,gama]=atreal(R);
[cregalfa,credelt]=desreg(R,dalfa,ddelta);
eulers=aneuler(cregalfa,credelt,gama);
A321=mrot321(eulers);
MV=starvec(R);
MW=esmv(MV,A321);
adts=asdecls(MW);
[N,adsc,respsta]=starind(adts);
if(respsta==0)
disp('Nenhuma estrela detectada no campo aleatório...repita processo.');
```

```

else
if(respsta==1)
disp('Apenas uma estrela detectada no campo aleatório...repita processo.');
```

```

else
adsc(:,1)=adsc(:,1)/15;
adw=cretes(adsc);
adv=esmv(adw,A321);
ad=asdecls(adv);
n=size(N);
n=n(2);

```

```

disp('O número de estrelas presentes no campo de visada é:')
disp(n)
figure(1)
sensplo(eulers);
resp=input('recalcula?(s/n):','s');
if(resp=='s')
    j=1;
else
    j=2;
end
end
end
end
end
return

```

### Balstar.m

```

function [adat,ind,resp]=balstar(ad,ade,raio)

% function [adat,ind,resp]=balstar(ad,ade,raio)
% balstar - retorna as coordenadas inerciais das estrelas que
% tiveram única identificação dentro da circunferência centrada
% na estrela vista pelo sensor e de raio conhecido, e os índices
% referentes às coordenadas da matriz do sensor.

adv=cretes(ad);
adev=cretes(ade);
k=1;
s=size(ade);
s=s(1);
n=size(ad);
n=n(1);
for i=1:s;
    ni=0;
    for j=1:n;
        pe=adv(:,j)*adev(:,i);
        teta=acos(pe);
        sentet=sin(teta);
        if(sentet<raio)
            ni=ni+1;
            in=j;
        end
    end
    if(ni==1)
        ind(k)=i;
        adatv(:,k)=adv(:,in);
        k=k+1;
    end
end
end
s=size(adatv);
s=s(2);

```

```

resp=s;
if(resp==0 | resp==1)
    resp=0;
    return
else
    si=size(adatv);
    si=si(2);
    if(si==2)
        if(adatv(:,1)==adatv(:,2))
            resp=0;
            return
        end
    end
    adat=asdecls(adatv);
    for i=1:s;
        adat1(i,:)=adat(i,:);
        var=adat(i,:);
        for j=1:s;
            if(j==i)
                else
                    if(var==adat(j,:))
                        ind(j)=0;
                        adat(j,:)=[36 100];
                    end
                end
            end
        end
    end
    k=1;
    for i=1:s;
        if(adat(i,:)==[36 100])
            else
                ind1(k)=ind(i);
                a(k,:)=adat(i,:);
                k=k+1;
            end
        end
    end
    ind=ind1;
    adat=a;
end
return

```

### Cretes.m

```
function MVin=cretes(ad)
```

```

echo off
% function MVin=cretes(ad)
% cretes - converte as coordenadas das estrelas de ascensão reta e
% declinação contidas na matriz ad n*2 para coordenadas retangulares
% onde o eixo z coincide com o eixo da esfera celeste e o eixo x
% aponta na direção da referência da ascensão reta.Os vetores de
% saída(colunas de MVin) já são normalizados.

```

```

s=size(ad);
s=s(1);
for i=1:s;
    ad(i,2)=rad(ad(i,2));
    ad(i,1)=ad(i,1)*15;
    ad(i,1)=rad(ad(i,1));
end
for i=1:s;
    teta1=ad(i,1);
    teta2=ad(i,2);
    Vx=cos(teta2)*cos(teta1);
    Vy=sin(teta1)*cos(teta2);
    Vz=sin(teta2);
    V(:,i)=[Vx;Vy;Vz];
end
MVin=V;
return

```

### Degree.m

```

function Vang = degree(Vang)

echo off
% function Vang = degree(Vang)
% degree - retorna os valores dos elementos de Vang, de radianos
% para graus.

n=size(Vang);
l=n(1);
c=n(2);
cgr = 180/pi;
Vang=Vang*cgr;
return

```

### Desreg.m

```

function [cregalfa,credelt] = desreg(R,dalfa,ddelta)

echo off
% function [cregalfa,credelt] = desreg(R,dalfa,ddelta)
% desreg - retorna o centro da atitude deslocado do
% centro da região observada.

zalfreg = (R(2,1)+R(1,1))/2;
zdelreg = (R(2,2)+R(1,2))/2;
cregalfa = zalfreg+dalfa;
credelt = zdelreg+ddelta;
return

```

## Difang.m

```
function difads = difang(ads,adsc)

echo off
% function difads = difang(ads,adsc)
% difang - retorna a diferença dos ângulos elementos de
% ads com relação aos elementos de adsc.

difads=ads-adsc;
s=size(difads);
s=s(1);
for i=1:s;
    if(abs(difads(i,1))>10)
        sinal=adsc(i,1)/abs(adsc(i,1));
        difads(i,1)=sinal*(360-abs(difads(i,1)));
    end
end
return
```

## Eran321.m

```
function era321 = eran321(A,M)

echo off
% function era321 = eran321(A,M)
% eran321 - retorna os erros de ângulos de Euler em (g,m,s)
% da matriz A com relação a M em rotação 321.

ang = aneu321(M);
ang1 = aneu321(A);
era321 = difang(ang1,ang);
era321 = vgms(era321');
return
```

## Esmv.m

```
function MWin = esmv(MVin,A)

echo off
% function MWin = esmv(MVin,A)
% esmv - retorna os vetores MWin normalizados de estrelas no
% referencial do satélite a partir dos vetores no referencial
% inercial normalizados MVin e da matriz de rotação A.

MWin = A*MVin;
return
```

## Gms.m

```
function ang = gms(ang)

echo off
% function ang = gms(ang)
% gms - retorna o valor do ângulo em graus, minutos e segundos.

ang1=fix(ang);
mina=abs(ang-ang1)*60;
minal=fix(mina);
sega=abs(mina-minal)*60;
segal=round(sega);
ang=[ang1 minal segal];
return
```

## Idreg.m

```
function R=idreg(eulers)

% function R=idreg(eulers)
% indreg - identifica a região para a qual os
% ângulos de euler apontam.

asc=eulers(1)/15;
dec=-eulers(2);
load catal.mat
if(asc>crucis(1,1))
    if(asc<crucis(2,1))
        if(dec>crucis(1,2))
            if(dec<crucis(2,2))
                R=crucis;
            end
        end
    end
end
if(asc>scorpii(1,1))
    if(asc<scorpii(2,1))
        if(dec>scorpii(1,2))
            if(dec<scorpii(2,2))
                R=scorpii;
            end
        end
    end
end
if(asc>tauri(1,1))
    if(asc<tauri(2,1))
        if(dec>tauri(1,2))
            if(dec<tauri(2,2))
                R=tauri;
            end
        end
    end
end
```

```

    end
  end
end
end
if(asc>orionis(1,1))
  if(asc<orionis(2,1))
    if(dec>orionis(1,2))
      if(dec<orionis(2,2))
        R=orionis;
      end
    end
  end
end
end
if(asc>pegasus(1,1))
  if(asc<pegasus(2,1))
    if(dec>pegasus(1,2))
      if(dec<pegasus(2,2))
        R=pegasus;
      end
    end
  end
end
end
return

```

### **Ind.m**

```

function [Ni,star]=ind(eulers1,R)

% function [Ni,star]=ind(eulers1,R)
% ind - retorna as coordenadas das estrelas
% no ref. do satélite que encontram-se no
% campo de visada e os seus números no catálogo.

asds=starcor(R);
A=mrot321(eulers1);
svec=cretes(asds);
svecs=A*svec;
asdsa=asdecls(svecs);
[Ni,star]=starind(asdsa);
star(:,1)=star(:,1)/15;
return

```

### **Modulo.m**

```

function mV = modulo(V)

echo off
% function mV = modulo(x)
% modulo - Retorna o módulo do vetor V.

```



```

s = size(V);
if s(2) == 1
    mV = sqrt(V'*V);
else
    mV = sqrt(V*V');
end
return

```

### Mrot321.m

```

function A321 = mrot321(eulers)

echo off
% function A321 = mrot321(eulers)
% mrot - Retorna a matriz de rotação 321 em função dos ângulos de
% Euler nos eixos 3-2-1 fi,teta e psi em graus,apartir do vetor de
% ângulos eulers=[fi teta psi].

fi = eulers(1);
teta = eulers(2);
psi = eulers(3);
fi = rad(fi);
teta = rad(teta);
psi = rad(psi);
asym = [0 1;-1 0];
asym1 = [0 0 -1;0 0 0;1 0 0];
sym = [1 0 0;0 0 0;0 0 1];
sym1 = [0 0 0;0 1 0;0 0 0];
id = eye(2);
Rfi = id * cos(fi) + asym * sin(fi);
Rtet = sym * cos(teta) + sym1 + asym1 * sin(teta);
Rpsi = id * cos(psi) + asym * sin(psi);
Rfi = [Rfi zeros(2,1); zeros(1,2) 1];
Rpsi = [1 zeros(1,2); zeros(2,1) Rpsi];
A321 = Rpsi * Rtet * Rfi;
return

```

### Nesmv.m

```

function MWien = nesmv(MWin,prec)

echo off
% function MWien = nesmv(MWin,prec)
% nesmv - retorna os vetores de estrelas no referencial do satélite
% normalizados com uma precisão de medição prec em segundos de
% arco.

s=size(MWin);
s=s(2);
prec=prec/3600;

```

```

sigma = ones(s,1)*rad(prec);
MWine = MWin + randn(3,s)*diag(sigma);
MWien = normalmc(MWine);
return

```

### Normal.m

```

function Vn = normal(V)

echo off
% function Vn = normal(V)
% normal - retorna o vetor Vn normalizado.

s = size(V);
if s(2) == 1
    Vn = V/sqrt(V'*V);
else
    Vn = V/sqrt(V*V');
end
return

```

### Normalmc.m

```

function Mnc = normalmc(M)

echo off
% function Mnc = normalmc(M)
% normalmc - retorna as colunas da matriz Mnc normalizadas
% correspondentes às colunas da matriz M.

s = size(M);
for i=1:s(2)
    M(:,i) = normal(M(:,i));
end
Mnc = M;
return

```

### Omega.m

```

function W = omega(V)

echo off
% function W = omega(V)
% omega - retorna a matriz para operador produto vetorial
% onde V é o primeiro vetor coluna do produto.

W = [0 V(3) -V(2); -V(3) 0 V(1); V(2) -V(1) 0];
return

```

## Otim.m

```
function r=otim(matriz,qualif)

echo off
% function r=otim(matriz,qualif)
% otim - retorna o valor otimizado da
% variável listada na primeira coluna
% com relação às suas funções listadas
% nas demais colunas.

s=size(matriz,1);
n=size(matriz,2);
resp=0;
resp1=0;
while(resp==0)
    if(resp1==1)
        resp2=input('Aumenta(a) ou diminui(d)?:', 's');
        resp3=0;
        while(resp3==0)
            if(resp2=='a')
                disp('Posicione raio menor e raio maior');
                disp('respectivamente no gráfico:');
                subplot(211)
                [x,y]=ginput(2);
                rmin=x(1); rmax=x(2);
            else
                rmin=input('Raio menor(graus):');
                rmax=input('Raio maior(graus):');
            end
            intrmin=fix(rmin);
            difrmin=rmin-intrmin;
            intdif=fix(difrmin/0.01);
            rmin=intrmin+0.01*intdif;
            intrmax=fix(rmax);
            difrmax=rmax-intrmax;
            intdif=fix(difrmax/0.01);
            rmax=intrmax+0.01*intdif;
            if(rmin<matriz(1,1) | rmax>matriz(s,1))
                if(rmin<matriz(1,1))
                    disp('Raio mínimo fora da amplitude do gráfico');
                    disp('Selecione raio mínimo maior que');
                    disp(matriz(1,1));
                end
                if(rmax>matriz(s,1))
                    disp('Raio máximo fora da amplitude do gráfico');
                    disp('Selecione raio máximo menor que');
                    disp(matriz(s,1));
                end
            end
            resp4=input('Deseja escala original?(s/n):', 's');
            if(resp4=='s')
                r=-1;
            end
            return
        end
    end
end
```

```

        end
    else
        resp3=1;
    end
end
intmat=fix(matriz(:,1));
difmat=matriz(:,1)-intmat;
intdif=fix(difmat/0.01);
m2(:,1)=intmat+0.01*intdif;
for j=1:s;
    if(m2(j,1)==rmin)
        ind1=j;
    end
    if(m2(j,1)==rmax)
        ind2=j;
    end
end
clear mat
mat=matriz(ind1:ind2,:);
clear fx
clear m1
else
    mat=matriz;
end
s1=size(mat,1);
for i=1:s1;
    for j=2:n;
        aux=qualif(j-1,1)-qualif(j-1,2);
        m1(i,j-1)=(mat(i,j)-qualif(j-1,2))/aux;
    end
end
m11=m1';
fx=min(m11);
[fmax,in]=max(fx);
r=mat(in,1);
if(resp1==1)
    figure(5)
else
    figure(4)
end
clg
subplot(211)
hold
for j=1:s1;
    if(j==1)
        plot(mat(:,1),m1(:,j),'w.');
```

```

    end
    if(j==2)
        plot(mat(:,1),m1(:,j),'y.');
```

```

    end
    if(j==3)
        plot(mat(:,1),m1(:,j),'g.');
```

```

    end
end
```

```

if(j==4)
    plot(mat(:,1),ml(:,j),'r.');
```

end

```

end
plot(mat(:,1),fx,'m.','r,fmax,'c*');
```

grid

```

xlabel('raio');
ylabel('F(raio) otimizada');
title('Otimização do raio');
subplot(212)
hold
axis([-1 1 0 10]);
plot(-0.6,8.5,'c*');
text(-0.4,8.5,'- Raio ótimo');
plot(-0.6,6.5,'w*');
text(-0.4,6.5,'-P0(identificação correta)');
plot(-0.6,5.3,'y*');
text(-0.4,5.3,'-P1(não identificação)');
plot(-0.6,4.1,'g*');
text(-0.4,4.1,'-P2(identificação ambígua)');
plot(-0.6,2.9,'r*');
text(-0.4,2.9,'-P3(identificação errônea)');
text(-0.6,0.5,'Probabilidades de identificação');
axis off
respos=input('Deseja Zoom na otimização?(s/n):','s');
if(respos=='s')
    resp1=1;
else
    resp=1;
end
end
return
```

### Plostar.m

```

function raio=plostar(ads,A,ro,sigma)

%function raio=plostar(ads,A,ro,sigma)
% plostar - retorna as posições calculadas das
% estrelas como círculos de raio determinado.

fi=[0:3:360];
s=size(fi);
s=s(2);
W=cretes(ads);
V=A*W;
raio=proraio(ro,sigma);
figure(1)
raio=rad(raio);
n=size(V);
n=n(2);
for j=1:n;
```

```

y=V(2,j);
z=V(3,j);
r1=(y^2+z^2)^2;
if(r1==0)
    u=-omega([0 1 0]')*V(:,j);
    u=normal(u);
else
    u=-omega([1 0 0]')*V(:,j);
    u=normal(u);
end
v=normal(-omega(V(:,j))*u);
for i=1:s;
    p1=cos(raio)*V(:,j);
    p2=(u*cos(fi(i))+v*sin(fi(i)))*sin(raio);
    pi(:,i)=p1+p2;
end
pi=normalmc(pi);
star=asdecls(pi);
subplot(211)
plot(-star(:,1),star(:,2),'g.')
end
raio=degree(raio);
return

```

### Plostat.m

```
function plostat(ads,raio)
```

```

%function plostat(ads,raio)
% plostar - plota as posições calculadas das
% estrelas como círculos de raio determinado.

```

```

fi=[0:3:360];
s=size(fi);
s=s(2);
V=cretes(ads);
raio=rad(raio);
n=size(V);
n=n(2);
for j=1:n;
    y=V(2,j);
    z=V(3,j);
    r1=(y^2+z^2)^2;
    if(r1==0)
        u=-omega([0 1 0]')*V(:,j);
        u=normal(u);
    else
        u=-omega([1 0 0]')*V(:,j);
        u=normal(u);
    end
    v=normal(-omega(V(:,j))*u);
    for i=1:s;

```

```

p1=cos(raio)*V(:,j);
p2=(u*cos(fi(i))+v*sin(fi(i)))*sin(raio);
pi(:,i)=p1+p2;
end
pi=normalmc(pi);
star=asdecls(pi);
subplot(211)
plot(-star(:,1),star(:,2),'y.')
end
return

```

### Plotreg.m

```

function [mm1,m12,m23,m34,mm4]=plotreg(R)

echo off
% function [mm1,m12,m23,m34,mm4]=plotreg(R)
% plotreg - retorna a impressão da região R e os vetores com
% os números das estrelas que estão classificadas em classes
% de número de magnitude.

load catal.mat
clg
s=size(R);
s=s(1);
axis([-R(2,1) -R(1,1) R(1,2) R(2,2)])
im1=1; i12=1; i23=1; i34=1; im4=1;
subplot(211)
hold on
for i=3:s;
if(R(i,3)<1)|(R(i,3)==1)
plot(-R(i,1),R(i,2),'w*')
mm1(im1)=i-2;
im1=im1+1;
end
if(R(i,3)<2)|(R(i,3)==2)
if(R(i,3)>1)
plot(-R(i,1),R(i,2),'c*')
m12(i12)=i-2;
i12=i12+1;
end,end
if(R(i,3)<3)|(R(i,3)==3)
if(R(i,3)>2)
plot(-R(i,1),R(i,2),'y*')
m23(i23)=i-2;
i23=i23+1;
end,end
if(R(i,3)<4)|(R(i,3)==4)
if(R(i,3)>3)
plot(-R(i,1),R(i,2),'m*')
m34(i34)=i-2;
i34=i34+1;

```

```

end,end
if(R(i,3)>4)
    plot(-R(i,1),R(i,2),'b*')
    mm4(im4)=i-2;
    im4=im4+1;
end
end
grid
ylabel('Declinação (g)')
xlabel('Asc. reta (h)')
if (R(1,:)==crucis(1,:))
    title('Crucis')
end
if(R(1,:)==scorpii(1,:))
    title('Scorpii')
end
if(R(1,:)==pegasus(1,:))
    title('Pegasus')
end
if(R(1,:)==orionis(1,:))
    title('Orionis')
end
if(R(1,:)==tauri(1,:))
    title('Tauri')
end
subplot(212)
hold on
axis([0 3 0 8])
plot(.9,6.5,'w*')
text(1,6.5,' - Mag=<1')
plot(.9,5.5,'c*')
text(1,5.5,' - 1<Mag=<2')
plot(.9,4.5,'y*')
text(1,4.5,' - 2<Mag=<3')
plot(.9,3.5,'m*')
text(1,3.5,' - 3<Mag=<4')
plot(.9,2.5,'b*')
text(1,2.5,' - 4<Mag<5')
text(1,0.5,'Magnitude')
axis off
return

```

### Proraio.m

```

function raio=proraio(ro,sigma)

echo off
% function raio=proraio(ro,sigma)
% proraio - plota gráficos de probabilidade
% de identificação contra o raio do círculo
% de identificação e permite ao usuário a
% escolha de um raio mais adequado conforme

```



```

% os gráficos apresentados.
raio=-1;
while(raio== -1)
    resp=0;
    resp1=0;
    while(resp==0)
        if(resp1==0)
            r=[0.01:0.01:4];
        else
            resp2=input('Aumenta(a) ou diminui(d)?','s');
            if(resp2=='a')
                disp('Posicione raio menor e raio maior');
                disp('respectivamente no gráfico:');
                subplot(211)
                [x,y]=ginput(2);
                rmin=x(1);rmax=x(2);
            else
                rmin=input('Raio menor(graus):');
                rmax=input('Raio maior(graus):');
            end
            r=[rmin:0.01:rmax];
        end
        q=0.99;
        ep=rad(r);
        Ac=2*pi*(1-cos(ep));
        mi=Ac*q*ro;
        expmi=exp(-mi);
        miexpmi=mi.*expmi;
        expraio=exp(-(ep.^2)/(2*sigma^2));
        P0=(miexpmi./(1-expmi)).*(1-expraio)*q;
        P1=expmi.*(1-q+q*expyraio);
        P3=P1.*mi;
        P2=1-(P0+P1+P3);
        if(resp1==0)
            figure(2)
        else
            figure(3)
        end
        clg
        subplot(211)
        semilogy(r,P0,'w.',r,P1,'y.',r,P2,'g.',r,P3,'r. ');
        grid
        ylabel('Prob. de identificação');
        xlabel('Raio do círculo(graus)');
        title('Pi x Raio');
        subplot(212)
        hold on
        axis([0 3 0 8]);
        plot(.7,6.5,'w*');
        text(.8,6.5,'-P0(identificação correta)');
        plot(.7,5.5,'y*');
        text(.8,5.5,'-P1(não identificação)');
        plot(.7,4.5,'g*');
    end
end

```

```

text(.8,4.5,'-P2(identificação ambígua)');
plot(.7,3.5,'r*');
text(.8,3.5,'-P3(identificação errônea)');
text(.6,0.5,'Probabilidades de identificação');
axis off
respos=input('Deseja Zoom?(s/n):','s');
if(respos=='s')
    resp1=1;
else
    resp=1;
end
end
disp('Deseja escolher raio no gráfico(g,');
disp('entrar valor(e) ou otimizar o valor do raio');
resp2=input('automaticamente(o)?','s');
if(resp2=='g')
    disp('Selecione o raio adequado no gráfico:');
    subplot(211)
    [a,b]=ginput(1);
    raio=a;
end
if(resp2=='e')
    raio=input('Raio(graus):');
end
if(resp2=='o')
    s(1)=size(P0,2);
    s(2)=size(P1,2);
    s(3)=size(P2,2);
    s(4)=size(P3,2);
    s(5)=size(r,2);
    t=min(s);
    matriz=[r' P0(1:t)' P1(1:t)' P2(1:t)' P3(1:t)'];
    qualif=[100 50;0 15;0 30;0 5];
    %qualif=[100 65;0 15;0 15;0 5];
    disp('Qualificação das porcentagens de variação');
    disp('das probabilidades em sequência(P0,P1,P2,P3):');
    disp(' Bom Ruim --(%));
    disp(qualif);
    qualif=qualif/100;
    raio=otim(matriz,qualif);
end
end
disp('Raio selecionado:');
disp(raio);
pause
return

```

### Rad.m

```
function Vang = rad(Vang)
```

```
echo off
```

```

% function Vang = rad(Vang)
% rad - retorna os valores dos elementos de Vang, de graus para
% radianos.

```

```

n=size(Vang);
l=n(1);
c=n(2);
crad = pi/180;
for i=1:l;
    for j=1:c;
        Vang(i,j) = Vang(i,j)*crad;
    end
end
return

```

### Rotn.m

```

function [rot,n] =rotn(A,M)

echo off
% function [rot,n] =rotn(A,M)
% rotn - retorna o versor n e o ângulo de rotação rot
% da matriz A com relação a M em (g,m,s).

deltaA = A*M';
elem = deltaA - eye(3);
n = [elem(2,3);elem(3,1);elem(1,2)];
rot = degree(modulo(n));
if(rot>360)
    aux=rot/360;
    aux=fix(aux);
    aux1=aux*360;
    rot=rot-aux1;
end
if(rot>360)
    aux=rot/360;
    aux=fix(aux);
    aux1=aux*360;
    rot=rot-aux1;
end
if(rot<-360)
    aux=rot/(-360);
    aux=fix(aux);
    aux1=aux*(-360);
    rot=rot-aux1;
end
rot = gms(rot);
n = normal(n);
return

```

## Satccd.m

```
function adse=satccd(coseret)

% function adse=satccd(coseret)
% satccd - transforma as coordenadas da matriz do sensor
% para o sistema eq. de ref. no referencial do sensor.

coseret=-coseret;
coser=rad(coseret);
s=size(coser);
s=s(1);
for i=1:s;
    dec=degree(sin(coser(i,2)));
    asc=degree(sin(-coser(i,1)))/15;
    adse(i,:)=[asc dec];
end
for i=1:s;
    if(adse(i,1)<0)
        adse(i,1)=adse(i,1)+24;
    end
end
return
```

## Sensor.m

```
function [coseret,prec]=sensor(adsc)

% function [coseret,prec]=sensor(adsc)
% sensor - Faz o papel do sensor de estrelas. Retorna
% as cordenadas das estrelas vistas na matriz de
% visualização do sensor e a sua precisão.

prec=input('Precisão do sensor estelar(seg.):');
adscw=cretes(adsc);
adscwe=nesmv(adscw,prec);
adscwe=normalmc(adscwe);
s=size(adsc);
s=s(1);
for i=1:s;
    y=asin(adscwe(3,i));
    x=-asin(adscwe(2,i));
    coseret(i,:)=[x y];
end
coser=degree(coseret);
for i=1:s;
    if(abs(coser(i,1))<4)
        if(abs(coser(i,2))<4)
            coser1(i,:)=coser(i,:);
        end
    end
end
```

```

end
coseret=-coserl;
return

```

### Senspl1.m

```

function senspl1(eulers)

% function senspl1(eulers)
% sensplo - plota a região de observação do sensor na região
% observada no céu para a atitude calculada.

hold on
eulers=rad(eulers);
cregalfa=eulers(1);
credelt=eulers(2);
gama=eulers(3);
m1=[1 -1 -1 1 1;
    1 1 -1 -1 1];
R=[-1 0;0 1];
m1=R*m1;
m12=ones(1,5);
asym=[0 1;-1 0];
id=eye(2);
A=-(cos(gama)*id+sin(gama)*asym);
stsoma=rad(4)*A*m1;
ptsoma=[cregalfa;credelt]*m12;
for i=1:5;
    stsoma(1,i)=stsoma(1,i)/cos(stsoma(2,i)+ptsoma(2,i));
end
vert=ptsoma+stsoma;
vert(1,:)=degree(vert(1,:))/15;
vert(2,:)=degree(vert(2,:));
vert=vert';
subplot(211)
plot(-vert(:,1),vert(:,2),'g-')
plot(-vert(1,1),vert(1,2),'go')
vert=vert(1:4,1:2);
return

```

### Sensplo.m

```

function sensplo(eulers)

% function sensplo(eulers)
% sensplo - plota a região de observação do sensor na região
% observada no céu.

hold on

```

```

eulers=rad(eulers);
cregalfa=eulers(1);
credelt=eulers(2);
gama=eulers(3);
m1=[1 -1 -1 1 1;
    1 1 -1 -1 1];
R=[-1 0;0 1];
m1=R*m1;
m12=ones(1,5);
asym=[0 1;-1 0];
id=eye(2);
A=-(cos(gama)*id+sin(gama)*asym);
stsoma=rad(4)*A*m1;
ptsoma=[cregalfa;credelt]*m12;
for i=1:5;
    stsoma(1,i)=stsoma(1,i)/cos(stsoma(2,i)+ptsoma(2,i));
end
vert=ptsoma+stsoma;
vert(1,:)=degree(vert(1,:))/15;
vert(2,:)=degree(vert(2,:));
vert=vert';
subplot(211)
plot(-vert(:,1),vert(:,2),'w-')
plot(-vert(1,1),vert(1,2),'wo')
vert=vert(1:4,1:2);
return

```

### Senspre.m

```

function [eulerse,sigma]=senspre(eulers)

echo off
%function [eulerse,sigma]=senspre(eulers)
%senspre - Simula a atitude pré-estimada pelo sensor de pré
%estimativa, incluindo seus erros.

A=mrot321(eulers);
I=randn(3,2);
I=normalmc(I);
S=A*I;
prec=input('Precisão do sensor de pré estimativa(graus):');
prec=prec*60^2;
sigma=rad(prec/60^2);
Se=nesmv(S,prec);
Ape=triadop(I,Se,prec);
eulerse=aneu321(Ape);
fi=eulerse(1);
if(fi<0)
    eulerse(1)=fi+360;
end
clc

```

```

disp('Plotando atitude pré-estimada...');
figure(1)
senspl1(eulerse);
pause
clc
return

```

### Simul.m

```

function [eulers,adsc,ro]=simul

% function [eulers,adsc,ro]=simul
% simul - simula a atitude do satélite e as estrelas a
% serem vistas pelo sensor.

disp('Serão simuladas a atitude do satélite e as estrelas a');
disp('serem vistas pelo sensor');
save var.mat;
clear
load catal.mat;
who
R =input('Selecione a região do céu a ser observada:');
lim=R(1:2,1:2);
Ar=areareg(lim);
asde=starcor(R);
s=size(asde);
s=s(1);
ro=s/Ar;
rog=ro/degree(1)^2;
disp('Número de estrelas catalogadas na região escolhida:');
disp(s)
disp('Densidade de estrelas na região(estrelas/grau^2):');
disp(rog);
[eulers,ad]=atvis(R);
clc
disp('Plotando atitude simulada...');
figure(1)
plotreg(R);
sensplo(eulers);
pause
clc
adv=cretres(ad);
A=mrot321(eulers);
adw=A*adv;
adsc=asdecls(adw);
return

```

### Softana.m

```

function softana(A321,eulerse,adat,raio,eulers)

```

```

% function softana(A321,eulerse,adat,raio,eulers)
% softana - retorna as características principais
% da atitude calculada pelo sistema de identificação
% de estrelas e cálculo de atitude.

As=mrot321(eulers);
eulers1=aneu321(A321);
fi=eulers1(1);
if(fi<0)
    eulers1(1)=fi+360;
end
A=mrot321(eulerse);
era321=eran321(A321,A);
era321c=eran321(A321,As);
era321p=eran321(A,As);
[rot,n]=rotn(A321,A);
[rotc,nc]=rotn(A321,As);
rotcg=angr(rotc);
rotcrad=rad(rotcg);
ncv=nc*rotcrad;
[rotp,np]=rotn(A,As);
Vss=vsens(As);
asdss=asdecls(Vss);
Vs=vsens(A);
asds=asdecls(Vs);
Vs1=vsens(A321);
asds1=asdecls(Vs1);
ns=ncv*[1;0;0];
nps=sqrt(rotcrad^2-ns^2);
clc
disp('Análise com base de dados fornecidos pelo sensor de estrelas,');
disp('sensor de atitude pré-estimada e simulador.');
```

pause

```

clc
disp('Ângulos de Euler321 associados à atitude pré-estimada:');
disp(eulerse);
disp('Ângulos de Euler321 associados à atitude calculados:');
disp(eulers1);
disp('Ângulos de Euler321 associados à atitude real simulada:');
disp(eulers);
disp('Erros calculado/pré-est. nos ângulos de euler321(g,m,s):');
disp(era321);
disp('Erros calculado/real nos ângulos de euler321(g,m,s):');
disp(era321c);
disp('Erros pré-est./real nos ângulos de euler321(g,m,s):');
disp(era321p);
pause
clc
disp('Matriz de atitude 321 calculada A321:');
disp(A321);
disp('Matriz de atitude 321 pré-estimada A:');
disp(A);
disp('Matriz de atitude 321 real As:');
```



```

disp(As);
pause
clc
disp('Versor de giro de erro de A321 para A(no ref. do sat.):');
disp(n);
disp('Rotação em torno de n calc./pré-est.:');
disp(rot);
disp('Versor de giro de erro de A321 para As(no ref. do sat.):');
disp(nc);
disp('Rotação em torno de n calc./real:');
disp(rotc);
disp('Versor de giro de erro de A para As(no ref. do sat.):');
disp(np);
disp('Rotação em torno de n pré-est./real:');
disp(rotp);
pause ,clc
disp('Erro associado à atitude na direção do eixo do sensor(g,m,s):');
ns=degree(ns);
ns=gms(ns);
disp(ns);
disp('Erro associado à atitude, perpendicular ao eixo do sensor(g,m,s):');
nps=degree(nps);
nps=gms(nps);
disp(nps);
disp('Erro de rotação de atitude calculada/real(g,m,s):');
disp(rotc);
pause,clc
disp('Versor de apontamento pré-estimado do sensor:');
Vs=Vs';
disp(Vs);
Vs=Vs';
disp('Coordenadas de apontamento pré-estimado do sensor:');
disp(asds);
disp('Versor de apontamento calculado do sensor:');
Vs1=Vs1';
disp(Vs1);
Vs1=Vs1';
disp('Coordenadas de apontamento calculado do sensor:');
disp(asds1);
disp('Versor de apontamento real do sensor:');
Vss=Vss';
disp(Vss);
Vss=Vss';
disp('Coordenadas de apontamento real do sensor:');
disp(asdss);
num=size(adat);
num=num(1);
disp('Número de estrelas identificadas:');
disp(num);
R=idreg(eulers);
figure(6)
disp('Plotando a atitude pré-estimada...');
clg

```

```

plotreg(R);
sensplo(eulerse);
pause
figure(6)
disp('Plotando a atitude calculada...');
senspl1(eulers1);
pause
figure(6)
disp('Serão desenhados círculos amarelos em torno das estrelas');
disp('identificadas, não sendo estas parte do processo de identificação.');
```

```

ad=adat(:,1:2);
plostat(ad,raio);
return

```

### Softat.m

```
function A321=softat(coseret,adat,prec)
```

```

% function A321=softat(coseret,adat,prec)
% softat - calcula a atitude do satélite apartir dos dados
% do sensor e do software de identificação de estrelas e as
% coordenadas inerciais das estrelas vistas pelo sensor.

```

```

coseret=-coseret;
ind1=adat(:,3);
adat=adat(:,1:2);
s=size(ind1);
s=s(1);
for i=1:s;
    coser(i,:)=coseret(ind1(i,:));
end
adse=satccd(-coser);
MW=cretres(adse);
MV=cretres(adat);
A321=triadop(MV,MW,prec);
return

```

### Softind.m

```
function [adat,raio,resp]=softind(coseret,prec,eulerse,ro,sigma)
```

```

% function [adat,raio,resp]=softind(coseret,prec,eulerse,ro,sigma)
% softind - Faz o papel do software de identificação de estrelas
% vistas pelo sensor. Retorna as coordenadas inerciais para o
% cálculo da atitude com os referidos índices das coordenadas
% da matriz do sensor e o raio em torno das estrelas vistas.
% Ou retorna uma resposta de não identificação

```

```

coseret=-coseret;
adse=satccd(-coseret);

```

```

adsew=cretes(adse);
adsew=normalmc(adsew);
A=mrot321(eulerse);
adev=A'*adsew;
ade=asdecls(adev);
disp('Plotando estrelas vistas pelo sensor...');
raio=plostar(adse,A,ro,sigma);
raio=rad(raio);
R=idreg(eulerse);
ad=starcor(R);
[adat,ind,resp]=balstar(ad,ade,raio);
if (resp==0)
    adat=0;
    ind=0;
else
    adat=[adat ind'];
end
raio=degree(raio);
return

```

### Starcor.m

```

function asds=starcor(R)

% function asds=starcor(R)
% starcord - retorna somente as coordenadas
% das estrelas da região R.

s=size(R);
s=s(1)-2;
for i=1:s;
    asds(i,:)=[R(i+2,1) R(i+2,2)];;
end
return

```

### Starind.m

```

function [N,instar,respsta] = starind(adts)

echo off
% function [N,instar,respsta] = starind(adts)
% starind - retorna os números de identificação das estrelas
% que estão no campo de visada do sensor(8*8) e seus
% respectivos ângulos com relação ao referencial do satélite.

s=size(adts);
s=s(1);
j=0;
sinqu=sin(rad(4));
for i=1:s;

```

```

xalfa=sin(rad(adts(i,1)*15));
ydelta=sin(rad(adts(i,2)));
if(xalfa<sinqu & xalfa>-sinqu & ydelta<sinqu...
& ydelta>-sinqu)
    j=j+1;
    instar(j,:)=[adts(i,1) adts(i,2)];
    N(j,1)=i;
end
end
N=N';
if(j==0)
    respsta=0;
else
    if(j==1)
        respsta=1;
    else
        instar(:,1)=instar(:,1)*15;
    end
end
return

```

### Starvec.m

```

function MV = starvec(R)

echo off
% function MV = starvec(R)
% starvec - retorna todos os vetores associados às estrelas
% da região R.

tstar=starcor(R);
MV=cretes(tstar);
return

```

### Triad.m

```

function A = triad(V1,V2,W1,W2)

echo off
% function A = triad(V1,V2,W1,W2)
% triad - Three-axis attitude determination
% Retorna a matriz de atitude( ou rotação) a partir de
% dois vetores colunas medidos em duas bases diferentes
% V1 e V2, W1 e W2.

r2 = -omega(V1) * V2;
r2 = normal(r2);
r3 = -omega(V1) * r2;
Mr = [V1 r2 r3];
s2 = -omega(W1) * W2;

```

```

s2 = normal(s2);
s3 = -omega(W1) * s2;
Ms = [W1 s2 s3];
A = Ms * Mr';
return

```

### Triadop.m

```

function A = triadop(MVin,MWin,prec)

echo off
% function A = triadop(MVin,MWin,prec)
% triadop - three-axis attitude determination
% optimized. Retorna a matriz atitude( ou rotação) a partir
% de n vetores colunas(colunas das matrizes MVin e MWin de
% entrada 3xn) normalizados medidos em duas bases diferentes,
% Vn, Wn, com erros de medida prec, e com uma precisão de
% cálculo de 1' de arco.

V1 = MVin(:,1);
V2 = MVin(:,2);
W1 = MWin(:,1);
W2 = MWin(:,2);
A = triad(V1,V2,W1,W2);
s = size(MVin);
n = s(2);
angerr = 1/3600;
angerr = rad(angerr);
mdqsi = 2/3600;
mdqsi = rad(mdqsi);
while (mdqsi>angerr)
Mio = zeros(3);
Uio = zeros(3,1);
for i=1:n;
m = A*MVin(:,i);
u1 = MWin(:,i)-m;
u1 = u1/prec;
m1 = omega(m)/prec;
Mi = m1'*m1;
Ui = m1*u1;
Mio = Mi + Mio;
Uio = Ui + Uio;
end
qsi = inv(Mio)*Uio;
mqsi = omega(qsi);
mdqsi = modulo(qsi);
A = expm2(mqsi)*A;
end
return

```

### Vatanqc.m

```
function Vang=vatanqc(M)

echo off
% function Vang=vatanqc(M)
% vatanqc - retorna os valores dos ângulos em rad formados pelos n
% vetores com componentes x e y e o eixo x da matriz M n*3.

s=size(M);
s=s(1);
for i=1:s;
    vteta(i)=atanqc(M(i,1),M(i,2));
end
Vang=vteta';
return
```

### Vgms.m

```
function Vang = vgms(V)

echo off
% function Vang = vgms(V)
% V é um vetor coluna com valores reais de ângulos em graus. vgms(V)
% retorna o valor dos ângulos em graus, minutos e segundos
% dispostos em linhas em uma matriz n*3 .

n = size(V);
s=n(1);
for i=1:s
    Vang(i,:)=gms(V(i));
end
return
```

### Vsens.m

```
function Vs=vsens(A)

echo off
% function Vs=vsens(A)
% vsens - retorna o vetor de apontamento do sensor para uma dada
% matriz de rotação A, sendo o eixo 'x' a sua posição de referência.

Vs = A*[1;0;0];
return
```

