



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA

**INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS**

sid.inpe.br/mtc-m21b/2015/04.22.16.03-TDI

## **CONTROLE DE ATITUDE COM VEÍCULO EM ALTA ROTAÇÃO**

Daniel Vieira

Dissertação de Mestrado do Curso de Pós-Graduação em Engenharia e Tecnologia Espaciais/Mecânica Espacial e Controle, orientada pelo Dr. Waldemar de Castro Leite Filho, aprovada em 22 de maio de 2015.

URL do documento original:

<<http://urlib.net/8JMKD3MGP3W34P/3JCADL2>>

INPE  
São José dos Campos  
2015

**PUBLICADO POR:**

Instituto Nacional de Pesquisas Espaciais - INPE

Gabinete do Diretor (GB)

Serviço de Informação e Documentação (SID)

Caixa Postal 515 - CEP 12.245-970

São José dos Campos - SP - Brasil

Tel.:(012) 3208-6923/6921

Fax: (012) 3208-6919

E-mail: pubtc@sid.inpe.br

**COMISSÃO DO CONSELHO DE EDITORAÇÃO E PRESERVAÇÃO  
DA PRODUÇÃO INTELECTUAL DO INPE (DE/DIR-544):****Presidente:**

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

**Membros:**

Dr. Gerald Jean Francis Banon - Coordenação Observação da Terra (OBT)

Dr. Amauri Silva Montes - Coordenação Engenharia e Tecnologia Espaciais (ETE)

Dr. André de Castro Milone - Coordenação Ciências Espaciais e Atmosféricas  
(CEA)

Dr. Joaquim José Barroso de Castro - Centro de Tecnologias Espaciais (CTE)

Dr. Manoel Alonso Gan - Centro de Previsão de Tempo e Estudos Climáticos  
(CPT)

Dr<sup>a</sup> Maria do Carmo de Andrade Nono - Conselho de Pós-Graduação

Dr. Plínio Carlos Alvalá - Centro de Ciência do Sistema Terrestre (CST)

**BIBLIOTECA DIGITAL:**

Dr. Gerald Jean Francis Banon - Coordenação de Observação da Terra (OBT)

Clayton Martins Pereira - Serviço de Informação e Documentação (SID)

**REVISÃO E NORMALIZAÇÃO DOCUMENTÁRIA:**

Simone Angélica Del Duca Barbedo - Serviço de Informação e Documentação  
(SID)

Yolanda Ribeiro da Silva Souza - Serviço de Informação e Documentação (SID)

**EDITORAÇÃO ELETRÔNICA:**

Marcelo de Castro Pazos - Serviço de Informação e Documentação (SID)

André Luis Dias Fernandes - Serviço de Informação e Documentação (SID)



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA

**INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS**

sid.inpe.br/mtc-m21b/2015/04.22.16.03-TDI

## **CONTROLE DE ATITUDE COM VEÍCULO EM ALTA ROTAÇÃO**

Daniel Vieira

Dissertação de Mestrado do Curso de Pós-Graduação em Engenharia e Tecnologia Espaciais/Mecânica Espacial e Controle, orientada pelo Dr. Waldemar de Castro Leite Filho, aprovada em 22 de maio de 2015.

URL do documento original:

<<http://urlib.net/8JMKD3MGP3W34P/3JCADL2>>

INPE

São José dos Campos

2015

Dados Internacionais de Catalogação na Publicação (CIP)

---

Vieira, Daniel.

V673c Controle de atitude com veículo em alta rotação / Daniel  
Vieira. – São José dos Campos : INPE, 2015.  
xx + 61 p. ; (sid.inpe.br/mtc-m21b/2015/04.22.16.03-TDI)

Dissertação (Mestrado em Engenharia e Tecnologia  
Espaciais/Mecânica Espacial e Controle) – Instituto Nacional de  
Pesquisas Espaciais, São José dos Campos, 2015.

Orientador : Dr. Waldemar de Castro Leite Filho.

1. Controle de precessão. 2. Controle de nutação. 3. Manobra  
de apontamento. 4. Simulação matlab/simulink. 5. atuadores  
ON/OFF. I.Título.

CDU 629.7062.2

---



Esta obra foi licenciada sob uma Licença [Creative Commons Atribuição-NãoComercial 3.0 Não Adaptada](https://creativecommons.org/licenses/by-nc/3.0/).

This work is licensed under a [Creative Commons Attribution-NonCommercial 3.0 Unported License](https://creativecommons.org/licenses/by-nc/3.0/).

Aprovado (a) pela Banca Examinadora  
em cumprimento ao requisito exigido para  
obtenção do Título de **Mestre** em  
**Engenharia e Tecnologia Espaciais/Mecânica  
Espacial e Controle**

Dr. Hélio Koiti Kuga

---

Presidente / INPE / SJC Campos - SP

Dr. Waldemar de Castro Leite Filho

---

Orientador(a) / IAE/CTA / SJC Campos - SP

Dr. Euler Gonçalves Barbosa

---

Convidado(a) / IAE/DCTA / São José dos Campos - SP

*Este trabalho foi aprovado por:*

maioria simples

unanimidade

**Título:** " Controle de atitude com veículo em alta rotação".

Aluno (a): **Daniel Vieira**

São José dos Campos, 22 de Maio de 2015



*“If no mistakes you have made, yet losing you are, a different game  
you should play.”*

YODA



*A meus pais Frederico e Leonícia, a minha irmã Patrícia e a  
meu irmão Danilo*



## AGRADECIMENTOS

Ao INPE, seu corpo docente e administração pela oportunidade de viver um aprendizado de qualidade, marcado pela confiança e ética aqui presentes.

Aos meus amigos Alexandre, Derso, Gustavo, João Marcos que sempre me mostraram motivos para continuar fazendo o que é correto.

A todo o pessoal da ASE-ECI, companheiros de trabalhos e irmãos na amizade que fizeram parte da minha formação e que vão continuar presentes em minha vida com certeza.

Ao meu orientador Dr. Waldemar, pelo suporte no pouco tempo que lhe coube, pelas suas correções e incentivos.

A meus pais e meus irmãos, pelo amor, incentivo e apoio incondicional.

E a todos que direta ou indiretamente fizeram parte da minha formação, o meu muito obrigado.



## RESUMO

Esse trabalho tem o objetivo de simular e analisar o algoritmo da manobra de precessão usada no veículo experimento SHEFEX 2. Essa manobra foi controlada por um sistema de controle não-linear com atuadores de bocais de gás frio. O experimento é feito em um veículo lançador de 2 estágios, o primeiro não controlado estabilizado por rotação e o segundo que deve seguir uma trajetória planejada. É durante a fase balística, quando termina a queima do primeiro estágio e antes de iniciar o segundo estágio, que o veículo faz a manobra de apontamento para que se faça a trajetória planejada. A simulação do algoritmo da manobra de precessão é feita no ambiente SIMULINK<sup>®</sup> e para a análise do algoritmo foi desenvolvido um algoritmo de controle próprio que serve tanto como uma maneira de estudar mais profundamente o problema como uma forma de referência para comparação. A condição de ambiente usada na simulação é de um veículo simétrico, girando em torno de seu eixo de menor inércia com atuação de jatos acionados na lateral de seu corpo de forma discreta (ON/OFF). O movimento é considerado livre de torques externos e as equações de movimento usadas são da biblioteca de funções PROPAT. Ao fim do trabalho é feita uma comparação entre os algoritmos de controle 1 e o algoritmo desenvolvido mostrando comparativamente seus desempenhos. Para estudos futuros, sugere-se a inclusão das dinâmicas dos atuadores e dos sensores, inserir erros de montagem e desalinhamentos dos sensores e também sugere-se um estudo mais aprofundado no algoritmo de controle 2 que tem muita margem para melhoras.

Palavras-chave: Controle de precessão. Controle de nutação. Manobra de apontamento. Simulação matlab/simulink. Atuadores ON/OFF. Dinâmica do corpo rígido livre de torques externos.



# ATTITUDE CONTROL OF VEHICLE AT HIGH ROTATION SPEED

## ABSTRACT

This work aims to simulate and analyze the precession maneuver algorithm used in the experimental vehicle SHEFEX 2, this maneuver was controlled by a non-linear control system with cold gas nozzle actuators. The experiment is done in a launch vehicle of two stages, the first is a non-controlled stabilized by rotation and the second that must follow a planned trajectory. It is during the ballistic phase, when you complete the first stage burning and before starting the second stage, that the vehicle does a maneuver to correct its attitude so it follows the planned trajectory. The simulation of the precession maneuver algorithm is made on the environment SIMULINK<sup>®</sup> and for the analysis of the algorithm was developed a control algorithm that serves both as a way to study more deeply the problem as in the form of reference for comparison with the first algorithm. The ambient condition used in the simulation is of a symmetrical vehicle, rotating around its axis of lower inertia and using jets commanded on the side of the vehicle body discretely (ON / OFF). The motion is considered of being free of external torques and the equations of motion used comes from the PROPAT library of functions. At the end of the work is done a comparison between the control algorithms 1 and the developed algorithm, showing the comparison of their performances. For future studies, it is suggested the inclusion of the dynamics of the actuators and sensors, insert assembling errors and misalignments of the sensors, also it is strongly suggested a further study in the control algorithm 2 that has a lot of room for improvement.

Keywords: Precession control. Nutation control. Pointing maneuver. Matlab/simulink simulation. ON/OFF actuators. Torque free Rigid-Body Dynamics.



## LISTA DE FIGURAS

	<u>Pág.</u>
1.1 Posição dos atuadores no veículo . . . . .	2
2.1 Ilustração do cone do corpo formado pelo movimento rotacional . . . . .	8
2.2 Ilustração do cone do corpo e do espaço . . . . .	9
3.1 Simplificação da visualização do movimento rotacional . . . . .	11
3.2 Movimento principio fisico . . . . .	13
3.3 Comandos de aceleração e correção . . . . .	14
3.4 Planificação do movimento angular . . . . .	16
3.5 Ilustração do ângulo $c$ . . . . .	16
3.6 Ângulos $r1, r2$ e $r3$ . . . . .	18
3.7 Lógica direção dos pulsos . . . . .	19
3.8 Angulo inicial e final de acionamento . . . . .	20
3.9 Diagrama de funcionamento do algoritmo de controle . . . . .	20
3.10 Movimento precessional esperado . . . . .	21
3.11 Diagrama do controle de precessão . . . . .	23
3.12 Ilustração vetor torque de controle $T_c$ . . . . .	24
3.13 Região de acionamento dos jatos . . . . .	25
3.14 Transformação vetor velocidade angular . . . . .	26
3.15 Diagrama do controle de nutação . . . . .	26
4.1 Blocos da simulação em Simulink . . . . .	27
4.2 Blocos Simulink - Região 1 . . . . .	28
4.3 Sub-sistema Controle 1 . . . . .	29
4.4 Sub-sistema Controle Precessao Nutacao . . . . .	30
4.5 Blocos Simulink - Região 2 . . . . .	31
4.6 Parâmetros da máscara do bloco Modelo . . . . .	32
4.7 Blocos Simulink - Região 3 . . . . .	33
4.8 Movimento de precessão para o alvo $\theta: 13^\circ$ e $\psi: 15^\circ$ . . . . .	35
4.9 Movimento de precessão para o alvo $\theta: 13^\circ$ e $\psi: 15^\circ$ na região do alvo . . . . .	36
4.10 Ângulo de cone $\gamma$ para $\theta_{alvo} = 13^\circ$ e $\psi_{alvo} = 15^\circ$ - Controle 1 . . . . .	36
4.11 Movimento de precessão para o alvo $\theta: 15^\circ$ e $\psi: 10^\circ$ . . . . .	38
4.12 Movimento de precessão para o alvo $\theta: 15^\circ$ e $\psi: 13^\circ$ na região do alvo . . . . .	39
4.13 Ângulo de cone $\gamma$ para $\theta_{alvo} = 15^\circ$ e $\psi_{alvo} = 13^\circ$ - Controle 1 . . . . .	39
4.14 Ajuste de ângulo $a$ , (ETTL, 2009) . . . . .	40
4.15 Comportamento comando de correção . . . . .	41

4.16	Ângulo $a$ no tempo para o caso $\theta: 15^\circ$ e $\psi: 13^\circ$ . . . . .	41
4.17	Comportamento comando de aceleração para $a > 180^\circ$ . . . . .	42
4.18	Código para definição do tipo de controle . . . . .	43
4.19	Código para definição do tipo de controle com correção . . . . .	43
4.20	Movimento de precessão para o alvo $\theta: 15^\circ$ e $\psi: 10^\circ$ com correção do comando . . . . .	43
4.21	Movimento de precessão para o alvo $\theta: 13^\circ$ e $\psi: 15^\circ$ usando o algoritmo de precessão 2 . . . . .	45
4.22	Movimento de precessão para o alvo $\theta: 13^\circ$ e $\psi: 15^\circ$ na região do alvo usando o algoritmo de precessão 2 . . . . .	45
4.23	Ângulo de cone $\gamma$ para $\theta_{alvo} = 13^\circ$ e $\psi_{alvo} = 15^\circ$ . . . . .	46
4.24	Movimento de precessão para o alvo $\theta: 15^\circ$ e $\psi: 13^\circ$ usando o algoritmo de precessão 2 . . . . .	47
4.25	Movimento de precessão para o alvo $\theta: 15^\circ$ e $\psi: 13^\circ$ na região do alvo usando o algoritmo de precessão 2 . . . . .	47
4.26	Ângulo de cone $\gamma$ para $\theta_{alvo} = 15^\circ$ e $\psi_{alvo} = 13^\circ$ . . . . .	48
4.27	Comandos de acionamento para o caso $\theta = 15$ e $\psi = 13$ usando o controle 2	49

## LISTA DE TABELAS

	<u>Pág.</u>
4.1 Propriedades físicas do experimento . . . . .	33
4.2 Comparação entre algoritmo de Controle 1 e algoritmo de Controle 2 . .	50
4.3 Comparação combinada Tempo X Custo . . . . .	50



## SUMÁRIO

	<u>Pág.</u>
<b>1 INTRODUÇÃO</b> . . . . .	<b>1</b>
<b>2 CONCEITOS BÁSICOS</b> . . . . .	<b>3</b>
2.1 Equações de Movimento . . . . .	3
2.1.1 Cinemática . . . . .	3
2.1.2 Dinâmica . . . . .	5
2.2 Movimento de Rotação Livre de Torque . . . . .	6
<b>3 SISTEMA DE CONTROLE</b> . . . . .	<b>11</b>
3.1 Controle 1 . . . . .	11
3.1.1 Simplificação do Movimento . . . . .	11
3.1.2 Princípio Físico . . . . .	12
3.1.3 Parâmetros de Controle . . . . .	15
3.1.3.1 Parâmetros $a, b, c$ e $d$ . . . . .	15
3.1.3.2 Parâmetros $\gamma_{atual}, r_1, r_2$ e $r_3$ . . . . .	17
3.1.4 Lógica de acionamento . . . . .	18
3.2 Controle 2 . . . . .	21
3.2.1 Controle de Precessão . . . . .	21
3.2.2 Controle de Nutação . . . . .	25
<b>4 SIMULAÇÃO</b> . . . . .	<b>27</b>
4.1 Descrição dos blocos . . . . .	27
4.1.1 Região 1 . . . . .	27
4.1.2 Região 2 . . . . .	31
4.1.3 Região 3 . . . . .	32
4.2 Resultados da simulação . . . . .	33
4.2.1 Valores numéricos . . . . .	33
4.2.2 Controle 1 . . . . .	34
4.2.3 Controle 2 . . . . .	44
4.2.4 Comparação algoritmos de controle . . . . .	48
<b>5 CONCLUSÃO</b> . . . . .	<b>51</b>

REFERÊNCIAS BIBLIOGRÁFICAS . . . . .	53
ANEXO A - Código da função para montar dados de entrada da simulação . . . . .	55
ANEXO B - Código do algoritmo de controle 1. . . . .	57
ANEXO C - Código do algoritmo de controle 2 - precessão. . . . .	59
ANEXO D - Código do algoritmo de controle 2 - nutação. . . . .	61

# 1 INTRODUÇÃO

O SHEFEX (SHarp Edge Flight EXperiment) é um projeto conduzido pelo DLR (Centro Aeroespacial Alemão) com objetivo de investigar o comportamento de veículos com superfícies não convencionais que reentram na atmosfera (TURNER et al., 2009).

A primeira versão do experimento, SHEFEX 1, foi lançado em 2005 e seguiu uma trajetória parabólica, sendo possível analisar o veículo com velocidade de reentrada da ordem de Mach 7.

Para a segunda versão do experimento, lançado em 2012, o SHEFEX 2 tinha necessidade de velocidades maiores, da ordem de Mach 10, porém sem haver redução no tempo do experimento. Assim, foi planejada para esse experimento uma trajetória mais linear e usado um veículo lançador maior, o VS40.

O VS40 é um veículo lançador brasileiro composto de dois estágios. No primeiro estágio o veículo faz um voo não guiado estabilizado por rotação. Com o fim do primeiro estágio e antes de acionar o segundo estágio o veículo segue um movimento balístico e nessa fase do voo é preciso fazer uma manobra de apontamento para que o veículo siga a trajetória de reentrada programada.

O foco de estudo nesse trabalho é o controle de apontamento que acontece na fase balística do voo e nesse instante o veículo é praticamente constituído de um cilindro com o computador de bordo, sensores e atuadores.

As informações da atitude vêm do sensor inercial e alimentam o computador de bordo que executa os comandos de controle a 100Hz. Os atuadores são bocais de gás frio posicionados em um mesmo plano no corpo e distantes  $120^\circ$  um do outro como ilustra a Figura 1.1. Acionar um atuador significa abrir o bocal liberando gás e, assim, gerando o torque que é usado para corrigir a atitude do veículo.

É importante destacar que, com o veículo estabilizado por rotação no primeiro estágio e sem atuadores que gerem torque no eixo longitudinal, o apontamento deve ser feito com a rotação residual do primeiro estágio.

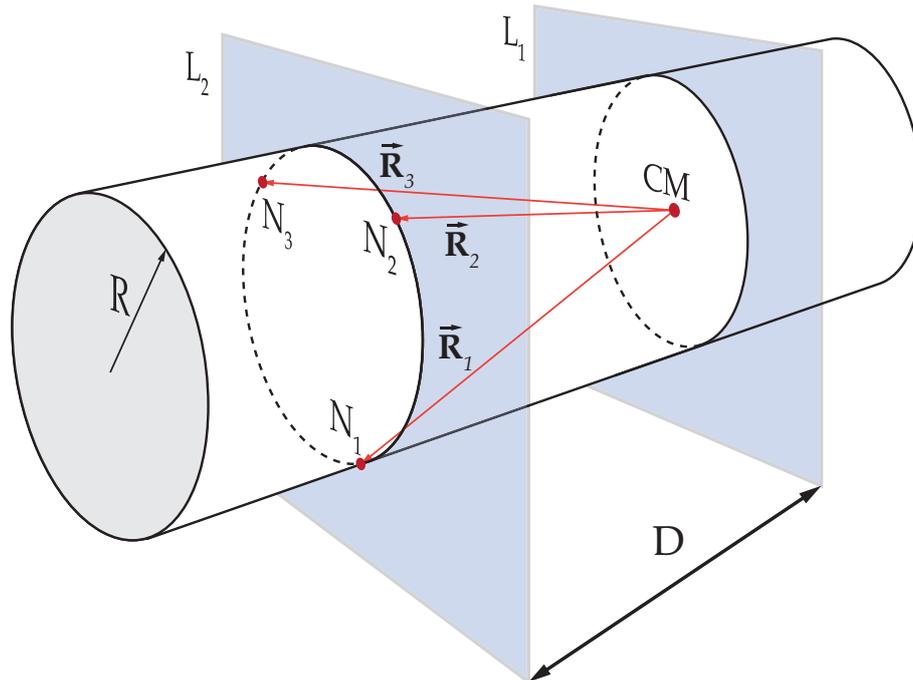


Figura 1.1 - Posição dos atuadores no veículo

Na missão realizada em 2012 foi utilizado um algoritmo de controle da manobra de precessão (ETTL, 2009) desenvolvido pela equipe do DLR. Para o projeto do SHEFEX 3 esse trabalho tratará de reproduzir o algoritmo usado em voo e analisá-lo quanto ao desempenho. Uma maneira de fazer a medida de desempenho do algoritmo de controle é por comparação com um outro algoritmo, então é proposto um controle de atitude baseado nos trabalhos encontrados na literatura clássica (SIDI, 1997) para ser usado como referência.

## 2 CONCEITOS BÁSICOS

### 2.1 Equações de Movimento

São chamadas equações de movimento o conjunto de equações diferenciais que precisam ser integradas para se obter a posição e atitude do veículo a cada instante e são agrupadas em duas categorias: cinemática e dinâmica.

#### 2.1.1 Cinemática

Considerando a sequência de rotação  $\mathbf{C}_1(\phi) \leftarrow \mathbf{C}_2(\theta) \leftarrow \mathbf{C}_3(\psi)$  de um triedro  $B$  para um triedro  $A$ , que simbolicamente é representado como

$$\mathbf{C}_3(\psi) : A' \leftarrow A, \quad (2.1)$$

$$\mathbf{C}_2(\theta) : A'' \leftarrow A', \quad (2.2)$$

$$\mathbf{C}_1(\phi) : B \leftarrow A''. \quad (2.3)$$

as derivadas no tempo dos ângulos de Euler,  $\phi$ ,  $\theta$  e  $\psi$ , são chamadas de  $\dot{\psi}$ ,  $\dot{\theta}$  e  $\dot{\phi}$ . Suas rotações sucessivas podem ser representadas como

$$\vec{\omega}^{A'/A} : A' \leftarrow A, \quad (2.4)$$

$$\vec{\omega}^{A''/A'} : A'' \leftarrow A', \quad (2.5)$$

$$\vec{\omega}^{B/A''} : B \leftarrow A''. \quad (2.6)$$

e os vetores de velocidade angular  $\vec{\omega}^{A'/A}$ ,  $\vec{\omega}^{A''/A'}$  e  $\vec{\omega}^{B/A''}$  são expressos como

$$\vec{\omega}^{B/A''} = \dot{\psi} \vec{a}_3 = \dot{\psi} \vec{a}'_3, \quad (2.7)$$

$$\vec{\omega}^{A''/A'} = \dot{\theta} \vec{a}'_2 = \dot{\theta} \vec{a}''_2, \quad (2.8)$$

$$\vec{\omega}^{A'/A} = \dot{\phi} \vec{a}''_1 = \dot{\phi} \vec{b}_1. \quad (2.9)$$

O vetor velocidade angular  $\vec{\omega}^{B/A}$  então se torna

$$\vec{\omega}^{B/A} = \vec{\omega}^{B/A''} + \vec{\omega}^{A''/A'} + \vec{\omega}^{A'/A} = \dot{\phi} \vec{b}_1 + \dot{\theta} \vec{a}''_2 + \dot{\psi} \vec{a}'_3, \quad (2.10)$$

que pode ser reescrito como

$$\vec{\omega}^{B/A} = [\vec{b}_1 \ \vec{b}_2 \ \vec{b}_3] \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + [\vec{a}''_1 \ \vec{a}''_2 \ \vec{a}''_3] \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + [\vec{a}'_1 \ \vec{a}'_2 \ \vec{a}'_3] \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix}, \quad (2.11)$$

e é sabido que

$$\begin{aligned} [\vec{a}''_1 \ \vec{a}''_2 \ \vec{a}''_3] &= [\vec{b}_1 \ \vec{b}_2 \ \vec{b}_3] \mathbf{C}_1(\phi), \\ [\vec{a}'_1 \ \vec{a}'_2 \ \vec{a}'_3] &= [\vec{b}_1 \ \vec{b}_2 \ \vec{b}_3] \mathbf{C}_1(\phi) \mathbf{C}_2(\theta). \end{aligned}$$

Como o vetor velocidade angular  $\vec{\omega} \equiv \vec{\omega}^{B/A}$  também pode ser representado como

$$\vec{\omega} = \omega_x \vec{b}_1 + \omega_y \vec{b}_2 + \omega_z \vec{b}_3 = [\vec{b}_1 \ \vec{b}_2 \ \vec{b}_3] \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}, \quad (2.12)$$

obtém-se

$$\begin{aligned} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} &= \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \mathbf{C}_1(\phi) \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \mathbf{C}_1(\phi) \mathbf{C}_2(\theta) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & \sin \phi \cos \theta \\ 0 & -\sin \phi & \cos \phi \cos \theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}. \end{aligned} \quad (2.13)$$

Fazendo a solução inversa da eq. 2.13 obtém-se

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \frac{1}{\cos \theta} \begin{bmatrix} \cos \theta & \sin \phi \sin \theta & \cos \phi \sin \theta \\ 0 & \cos \phi \cos \theta & -\sin \phi \cos \theta \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}, \quad (2.14)$$

que expandida dá origem às equações diferenciais:

$$\dot{\phi} = \omega_x + \omega_z \cos \phi \tan \theta + \omega_y \sin \phi \tan \theta, \quad (2.15)$$

$$\dot{\theta} = \omega_y * \cos \phi - \omega_z * \sin \phi, \quad (2.16)$$

$$\dot{\psi} = \frac{\omega_z * \cos \phi + \omega_y * \sin \phi}{\cos \theta}. \quad (2.17)$$

### 2.1.2 Dinâmica

Sabe-se que a variação do momento angular é igual à somatória dos torques externos atuando no corpo, expresso em notação matemática como

$$\dot{\vec{H}} = \vec{M}_b, \quad (2.18)$$

em que  $\vec{H}$  é o vetor momento angular e  $\vec{M}_b$  é o vetor do somatório dos torques externos.

Sabe-se também que o vetor momento angular é definido por

$$\vec{H} = \mathbf{I}\vec{\omega}_b, \quad (2.19)$$

em que  $\mathbf{I}$  é a matriz tensora de inércia e  $\vec{\omega}_b$  é o vetor velocidade angular do corpo.

Derivando a equação 2.19 obtém-se

$$\dot{\vec{H}} = \dot{\mathbf{I}}\vec{\omega}_b + \mathbf{I}\dot{\vec{\omega}}_b + \vec{\omega}_b \times \mathbf{I}\vec{\omega}_b. \quad (2.20)$$

A matriz  $\mathbf{I}$  é constante já que a única variação de massa no veículo acontece pela abertura dos bocal de gás frio. O veículo está em uma trajetória balística durante toda a janela de tempo em que funciona o controle. Então substituindo 2.20 em 2.18 e, como a variação de massa é desprezável em relação à massa total, eliminando o termo  $\dot{\mathbf{I}}$ , obtém-se

$$\dot{\vec{\omega}}_b = \mathbf{I}^{-1}(\vec{M}_b + \mathbf{I}\vec{\omega}_b \times \vec{\omega}_b). \quad (2.21)$$

Admitindo ainda que o corpo é simétrico e que por isso os produtos de inércia são nulos, obtém-se a seguinte matriz de inércia

$$\mathbf{I} = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix}. \quad (2.22)$$

Desenvolvendo a equação 2.21 obtém-se

$$\dot{\omega}_x = \frac{(I_y - I_z)\omega_y\omega_z}{I_x} + \frac{M_x}{I_x}, \quad (2.23)$$

$$\dot{\omega}_y = \frac{(I_z - I_x)\omega_x\omega_z}{I_y} + \frac{M_y}{I_y}, \quad (2.24)$$

$$\dot{\omega}_z = \frac{(I_x - I_y)\omega_x\omega_y}{I_z} + \frac{M_z}{I_z}. \quad (2.25)$$

## 2.2 Movimento de Rotação Livre de Torque

O veículo estudado nesse trabalho está em uma trajetória balística a uma altitude suficientemente grande para que as forças aerodinâmicas possam ser desprezadas. A única força atuando no corpo do veículo é a força gravitacional até que seja enviado um comando de controle para as válvulas. Como o gradiente de gravidade gera torques de grandezas muito menores que as envolvidas na manobra de apontamento do veículo, seu efeito é desprezado e, assim, a gravidade gera força apenas no centro de massa tornando o movimento livre de torque.

Retomando as equações 2.23, 2.24 e 2.25 e zerando os momentos nos eixos  $X$ ,  $Y$  e  $Z$  obtém-se

$$\dot{\omega}_x = \frac{(I_y - I_z)\omega_y\omega_z}{I_x}, \quad (2.26)$$

$$\dot{\omega}_y = \frac{(I_z - I_x)\omega_x\omega_z}{I_y}, \quad (2.27)$$

$$\dot{\omega}_z = \frac{(I_x - I_y)\omega_x\omega_y}{I_z}. \quad (2.28)$$

Como se trata de um corpo simétrico  $I_y = I_z = I_l$ , as equações de movimento passam a ser

$$\dot{\omega}_x = 0, \quad (2.29)$$

$$\dot{\omega}_y = \frac{(I_l - I_x)\omega_x\omega_z}{I_l}, \quad (2.30)$$

$$\dot{\omega}_z = \frac{(I_x - I_l)\omega_x\omega_y}{I_l}. \quad (2.31)$$

Da equação 2.29 tem-se que sem torque externo no corpo a velocidade de rotação é constante:

$$\omega_x = cte = \omega_0. \quad (2.32)$$

Agora, considerando a notação

$$\lambda = \frac{(I_l - I_x)\omega_x}{I_l}, \quad (2.33)$$

as equações 2.30 e 2.31 são reescritas como

$$\dot{\omega}_y - \lambda\omega_z = 0, \quad (2.34)$$

$$\dot{\omega}_z + \lambda\omega_y = 0. \quad (2.35)$$

Diferenciando a equação 2.34, obtém-se

$$\ddot{\omega}_y - \lambda\dot{\omega}_z = 0, \quad (2.36)$$

então, substituindo 2.36 na equação 2.35, tem-se

$$\ddot{\omega}_y + \lambda^2\omega_y = 0. \quad (2.37)$$

A solução desta equação diferencial é conhecida:

$$\omega_y = \omega_{yz} \sin \lambda t, \quad (2.38)$$

em que  $\omega_{yz}$  é uma constante de integração. Então, substituindo a equação 2.38 na equação 2.34 obtém-se a solução para  $\omega_z$ :

$$\omega_z = \frac{1}{\lambda} \frac{d\omega_y}{dt} = \frac{1}{\lambda} \frac{d}{dt}(\omega_{yz} \sin \lambda t) = \omega_{yz} \cos \lambda t. \quad (2.39)$$

As equações 2.32, 2.38 e 2.39 dão as componentes da velocidade angular  $\vec{\omega}$ :

$$\vec{\omega} = \omega_0 \hat{\mathbf{i}} + \omega_{yz} \sin \lambda t \hat{\mathbf{j}} + \omega_{yz} \cos(\lambda t) \hat{\mathbf{k}}. \quad (2.40)$$

Ou, segundo outra notação

$$\vec{\omega} = \omega_0 \hat{\mathbf{i}} + \vec{\omega}_\perp, \quad (2.41)$$

em que

$$\vec{\omega}_\perp = \omega_{yz} \sin \lambda t \hat{\mathbf{j}} + \omega_{yz} \cos(\lambda t) \hat{\mathbf{k}}. \quad (2.42)$$

$\vec{\omega}_\perp$  é a velocidade angular lateral, a componente de  $\vec{\omega}$  normal ao eixo X. Ela descreve um círculo de raio  $\omega_{yz}$  com velocidade angular  $\lambda$ , esse movimento descreve a superfície de um cone, denominado cone do corpo, que é ilustrado na Figura 2.1.

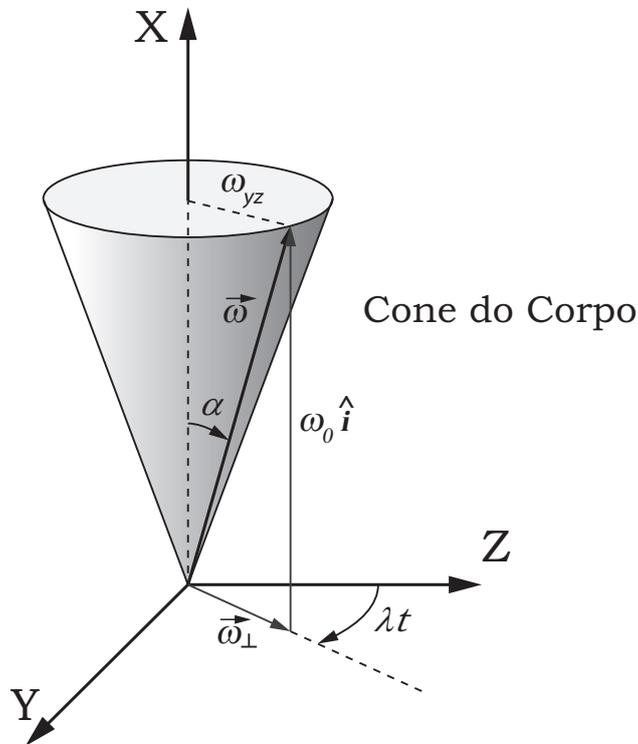


Figura 2.1 - Ilustração do cone do corpo formado pelo movimento rotacional

Se o corpo está em movimento rotacional causado por condição inicial e não está sobre influência de torques externos, então a norma da velocidade angular  $\vec{\omega}$  é constante. Como não há torques aplicados no corpo, o vetor momento angular  $\vec{H}$  é contante no espaço inercial e, com a condição de simetria nos eixos  $Y$  e  $Z$ , o vetor momento angular é calculado como

$$\vec{H} = I_x \vec{\omega}_x + I \vec{\omega}_\perp. \quad (2.43)$$

Os vetores  $\vec{\omega}$  e  $\vec{H}$  têm componentes nas mesmas direções ( $\vec{\omega}_x$  e  $\vec{\omega}_\perp$ ) implicando que  $\vec{H}$ ,  $\vec{\omega}$  e  $\vec{\omega}_x$  são coplanares. Como o vetor  $H$  é constante, sua direção é constante no espaço inercial, porém  $\vec{\omega}_\perp$  tem um movimento natural como mostra a Figura 2.1 e essa condição faz com que o vetor  $\vec{\omega}_\perp$  e o vetor  $\vec{\omega}$ , que são coplanares, rotacionem no plano  $YZ$  ao redor de  $\vec{H}$ .

Na Figura 2.2 é ilustrado o cone do espaço, onde o ângulo  $\beta$  entre o vetor  $\vec{H}$  e o vetor  $\vec{\omega}_x$  é chamado de ângulo de nutação,  $\alpha$  é o ângulo de cone do corpo e  $\gamma$  é o ângulo de cone do espaço.

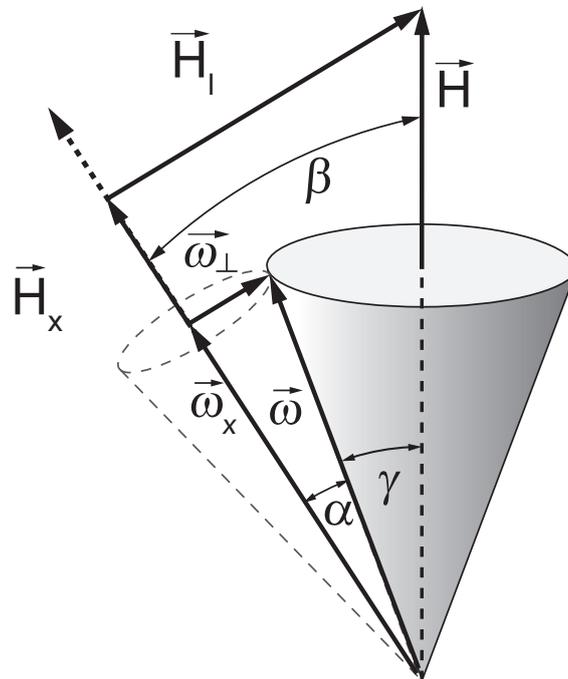


Figura 2.2 - Ilustração do cone do corpo e do espaço



### 3 SISTEMA DE CONTROLE

Esse capítulo descreve os algoritmos de controles testados no trabalho, que serão referenciados futuramente no texto como **Controle 1** e **Controle 2**.

#### 3.1 Controle 1

Esse é o algoritmo de controle usado no voo do SHEFEX 2 e essa sessão apresenta em maiores detalhes essa estratégia de controle descrita no documento de referência (ETTL, 2009).

##### 3.1.1 Simplificação do Movimento

O primeiro passo para entender o algoritmo é enxergar o movimento que o veículo faz durante a sua fase balística. Sua trajetória angular é simplificada e transformada em um movimento linear, um plano imaginário é criado e nele é projetada uma extensão do seu eixo longitudinal do veículo, então admite-se que o movimento é de deslocamentos angulares pequenos para aproximar o deslocamento angular da manobra a uma distância linear.

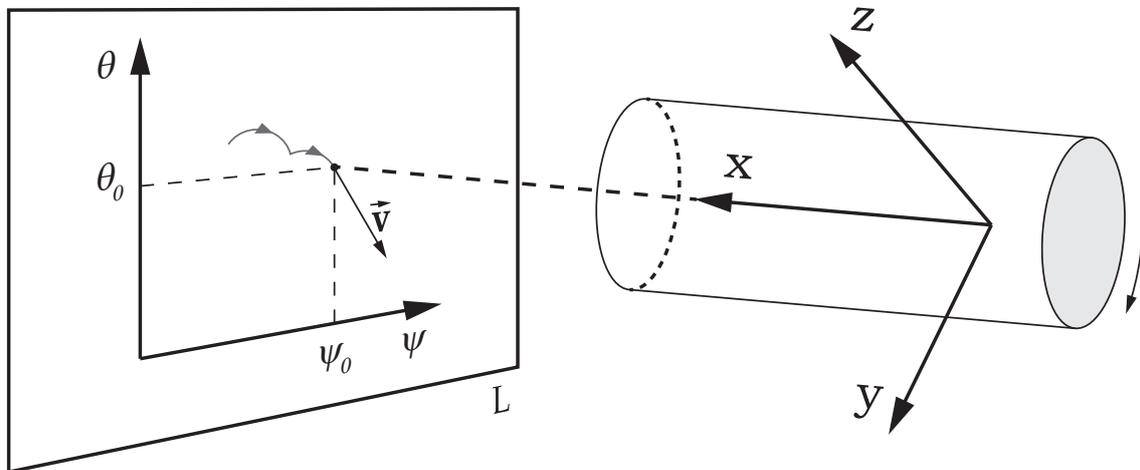


Figura 3.1 - Simplificação da visualização do movimento rotacional

Isso pode ser visto melhor através da figura 3.1 que mostra como a projeção do eixo **X** do cilindro descreve um caminho no plano **L**. Assim, o deslocamento angular é lido como uma distância linear e, da mesma forma, a velocidade angular é lida como uma velocidade linear  $\vec{v}$  com direção sempre tangencial ao caminho projetado do movimento.

### 3.1.2 Princípio Físico

Faz-se a consideração de que no início da simulação do veículo há apenas uma velocidade angular  $\omega_0$  no eixo longitudinal do veículo e portanto seu vetor momento angular inicial é

$$\vec{\mathbf{H}}_0 = I_x \omega_{x0} \hat{\mathbf{i}} = I_x \omega_0 \hat{\mathbf{i}}. \quad (3.1)$$

Nesse caso, sem torque externo no corpo, não há movimento de precessão até que seja aplicada a força por acionamento dos bocais de gás frio, quando o vetor momento angular  $\vec{\mathbf{H}}$  se desloca.

A Figura 3.2 ilustra o movimento do corpo, o eixo longitudinal do veículo rotaciona ao redor do vetor momento angular  $\vec{\mathbf{H}}_x$  e o seu movimento descreve a superfície do Cone do Corpo. Por sua vez o vetor  $\vec{\mathbf{H}}_x$  rotaciona ao redor do vetor momento angular total  $\vec{\mathbf{H}}$ , esse movimento descreve a superfície do Cone do Espaço. Usando a simplificação do movimento da seção anterior, percebe-se que a velocidade linear do caminho projetado no plano é sempre perpendicular ao vetor momento angular  $\vec{\mathbf{H}}_l$  já que a velocidade da projeção é tangente ao círculo formado pelo movimento de nutação e o vetor  $\vec{\mathbf{H}}_l$  aponta para o centro desse círculo.

O vetor  $\vec{\mathbf{H}}$  pode ser escrito como

$$\vec{\mathbf{H}} = \vec{\mathbf{H}}_x + \vec{\mathbf{H}}_l, \quad (3.2)$$

em que  $\vec{\mathbf{H}}_x$  e  $\vec{\mathbf{H}}_l$  são escritos como

$$\vec{\mathbf{H}}_x = I_x \omega_x \hat{\mathbf{i}}, \quad (3.3)$$

$$\vec{\mathbf{H}}_l = I_l \vec{\omega}_\perp. \quad (3.4)$$

E  $\vec{\omega}$  é dado por

$$\vec{\omega} = \omega_x \hat{\mathbf{i}} + \vec{\omega}_\perp, \quad (3.5)$$

em que  $\vec{\omega}_\perp$  é

$$\vec{\omega}_\perp = \omega_y \hat{\mathbf{j}} + \omega_z \hat{\mathbf{k}}. \quad (3.6)$$

O objetivo do controle é manter a velocidade da projeção do eixo  $X$ ,  $\vec{\mathbf{v}}$ , na direção do alvo, mas, como o corpo está em um movimento natural de nutação, é preciso fazer correções para manter a velocidade na direção correta.

As correções da trajetória são feitas com a abertura dos bocais de gás frio que geram

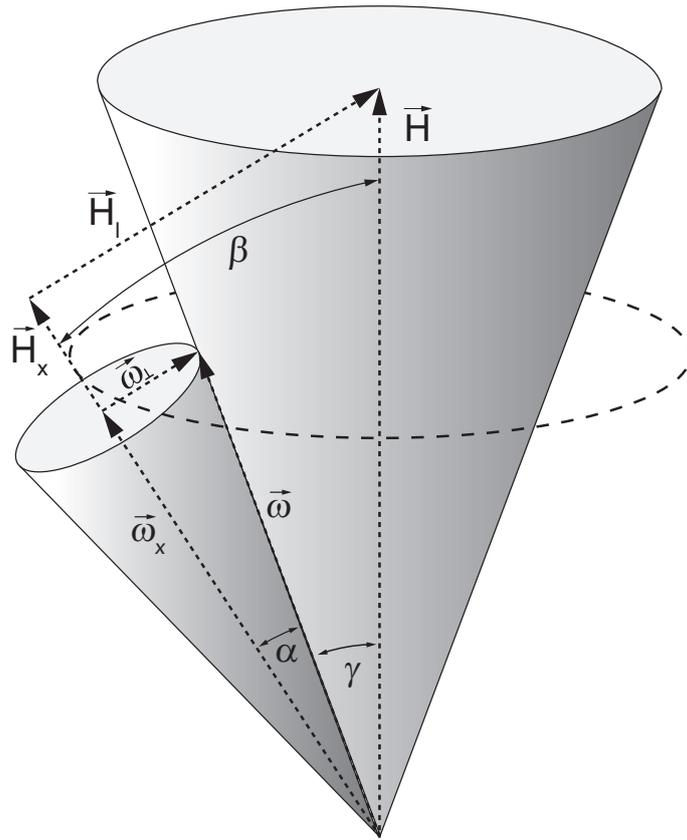


Figura 3.2 - Movimento principio fisico

torques no corpo que mudam seu vetor momento angular. Sabendo da relação de perpendicularidade entre  $\vec{H}_l$  e  $\vec{v}$ , os comandos são executados para manter o vetor  $\vec{H}_l$  perpendicular a linha que liga a posição projetada atual com a posição do alvo, fazendo com que a direção de  $\vec{v}$  esteja apontada para o alvo.

O algoritmo de controle trabalha com 3 modos de operação: aceleração, correção de direção e freio. O chaveamento entre os modos de operação depende das posições e velocidades angulares do veículo, a Figura 3.3a mostra uma situação em que acontece o comando de aceleração, um pulso é aplicado para gerar uma força  $\vec{F}_{pa}$  na direção do alvo, percebe-se que essa força gera também um torque  $\vec{T}_{pa}$  que muda o vetor  $\vec{H}_l$  em direção e módulo.

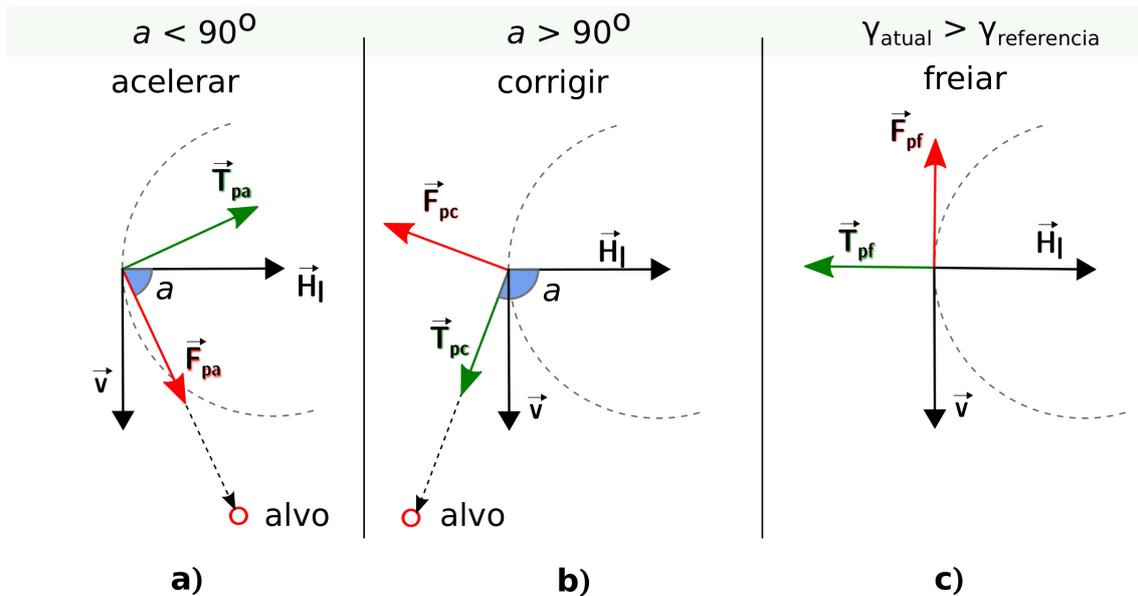


Figura 3.3 - Comandos de aceleração e correção

Na Figura 3.3b é ilustrada a situação em que o vetor  $\vec{v}$  está com a direção fora da região desejada, então é aplicado um pulso perpendicular à direção do alvo gerando a força  $\vec{F}_{pc}$  e o torque  $\vec{T}_{pc}$  que desloca a direção do vetor  $\vec{H}_l$  de volta para a região desejada.

Analisando os comandos de aceleração nota-se que para cada comando de aceleração há um aumento do módulo do vetor  $\vec{H}_l$  que implica em também aumentar o ângulo de cone do espaço. Como não é interessante aumentar indefinidamente esse ângulo de cone, há um valor limite de cone  $\gamma_{max}$  implementado no algoritmo e o comando de freio é acionado sempre que o cone do corpo for maior que esse valor. A Figura 3.3c ilustra a situação de um comando de freio, nesse comando a intenção é diminuir o módulo de  $\vec{H}_l$ , por isso é aplicado um pulso cuja força  $\vec{F}_{pf}$  gere um torque  $\vec{T}_{pf}$  de mesma direção, mas com sentido contrário a  $\vec{H}_l$ .

O ângulo de cone do corpo pode ser calculado por

$$\alpha = \arctan \frac{\omega_{\perp}}{\omega_x} \quad (3.7)$$

em que  $\alpha$  é o ângulo do cone do corpo,  $\omega_{\perp}$  é o módulo da velocidade angular perpendicular e  $\omega_x$  é o módulo da velocidade angular no eixo X.

O ângulo de cone do espaço é calculado por

$$\gamma = \beta - \alpha, \quad (3.8)$$

em que  $\gamma$  é o ângulo de cone do espaço e  $\beta$  é o ângulo formado entre os vetores  $\vec{\mathbf{H}}_x$  e  $\vec{\mathbf{H}}$ , calculado por

$$\beta = \arctan \frac{H_l}{H_x} = \arctan \frac{\omega_{\perp} I_l}{\omega_x I_x}. \quad (3.9)$$

### 3.1.3 Parâmetros de Controle

#### 3.1.3.1 Parâmetros $a$ , $b$ , $c$ e $d$

Usando a simplificação do movimento da Seção 3.1.1 é fácil criar relações geométricas que ajudem a definir o estado atual para calcular a direção do pulso de controle.

Na Figura 3.4 são ilustradas as relações geométricas usadas, o ponto  $P$  é a posição atual da projeção do eixo  $X$  e o ponto  $A$  é o alvo que deve ser alcançado, os dois pontos estão afastados por uma distância  $d$ .

O ângulo  $a$ , formado pelo vetor momento angular  $\vec{\mathbf{H}}_l$  com o segmento  $PA$ , o ângulo  $b$ , formado pelos segmentos  $P\theta'$  com o segmento  $PA$  e o ângulo  $c$ , formado pelo vetor momento angular  $\vec{\mathbf{H}}_l$  e o segmento  $P\theta'$  são as relações que descrevem a posição angular atual do veículo em relação à atitude alvo.

O ângulo  $b$  é calculado com a equação

$$b = \arctan \left( \frac{\psi_{alvo} - \psi_{atual}}{\theta_{alvo} - \theta_{atual}} \right). \quad (3.10)$$

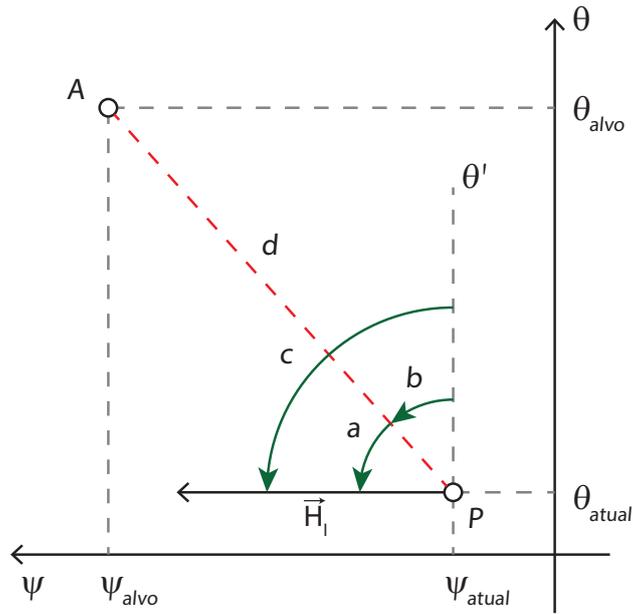


Figura 3.4 - Planificação do movimento angular

Para calcular o ângulo  $c$  é preciso usar mais uma relação geométrica, o ângulo  $c_v$  formado pela linha  $\theta'$  e o vetor velocidade da projeção. A Figura 3.5 ilustra como é formado o ângulo  $c_v$  e como ele se relaciona com o ângulo  $c$ .

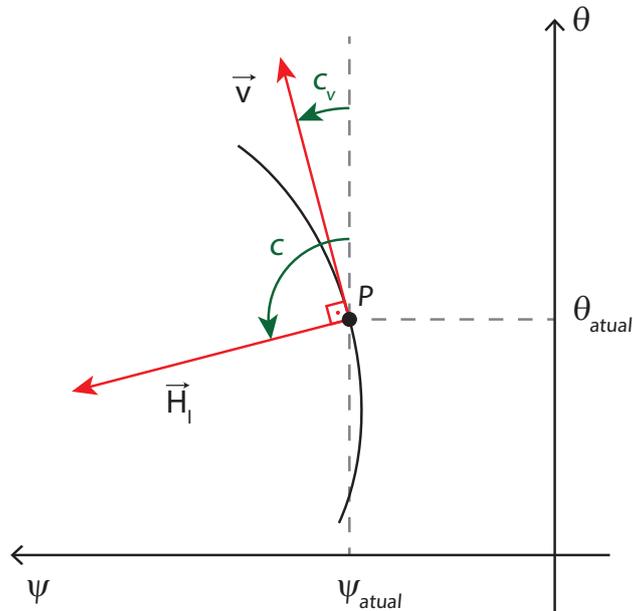


Figura 3.5 - Ilustração do ângulo  $c$

O valor de  $c_v$  pode ser calculado por

$$c_v = \arctan \left( \frac{\psi(t_n) - \psi(t_n - 1)}{\theta(t_n) - \theta(t_n - 1)} \right), \quad (3.11)$$

em que  $\psi(t_n)$  e  $\theta(t_n)$  são os ângulos de Euler atuais e  $\psi(t_n - 1)$  e  $\theta(t_n - 1)$  são os ângulos de Euler do passo anterior.

Como  $\vec{H}_l$  é perpendicular a  $\vec{v}$ ,  $c$  é calculado pela equação

$$c = c_v + 90^\circ. \quad (3.12)$$

Com o auxílio da Figura 3.4 percebe-se que o ângulo  $a$  pode ser calculado subtraindo  $b$  de  $c$ , escrito na forma de equação como

$$a = c - b. \quad (3.13)$$

A distância entre  $P$  e  $A$  é calculada na equação:

$$d = \sqrt{(\psi_{alvo} - \psi_{atual})^2 + (\theta_{alvo} - \theta_{atual})^2}. \quad (3.14)$$

### 3.1.3.2 Parâmetros $\gamma_{atual}$ , $r_1$ , $r_2$ e $r_3$

O parâmetro  $\gamma_{atual}$  representa o ângulo de cone do espaço e é usado na determinação do modo de acionamento a ser executado.

Os parâmetros  $r_1$ ,  $r_2$  e  $r_3$  são as posições angulares de cada bocal de gás frio em relação ao sistema inercial, mostrados na Figura 3.6. Como o eixo  $Z$  do sistema do corpo,  $Z_b$ , está posicionado em cima do bocal de gás frio  $N_1$ ,  $r_1$  é o ângulo formado entre o eixo  $Z$  do sistema de coordenadas inercial,  $Z_i$ , e o eixo  $Z_b$ . E como os bocais  $N_1$ ,  $N_2$  e  $N_3$  estão igualmente espaçados angularmente no corpo do veículo,  $r_2$  e  $r_3$  são calculados somando a  $r_1$  os ângulos de  $120^\circ$  e  $240^\circ$ , respectivamente.

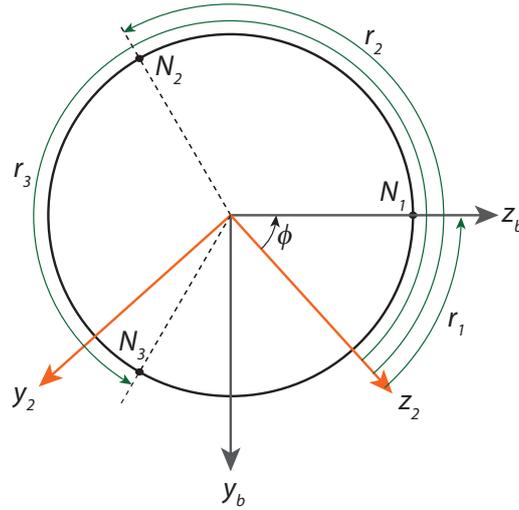


Figura 3.6 - Ângulos  $r_1$ ,  $r_2$  e  $r_3$

### 3.1.4 Lógica de acionamento

A lógica de acionamento é a aplicação do estudo do princípio físico para gerar os comandos de controle e apontar o veículo para a posição de referência e o resultado dessa lógica é a abertura do bocal de gás frio na posição e instante corretos.

Cada tipo de comando (aceleração, correção e freio) tem uma direção desejada de jato, visto anteriormente e ilustrado na Figura 3.3. Na lógica de acionamento é utilizada a convenção da Figura 3.4 para calcular o ângulo  $\alpha_{pulso}$ , que é o ângulo entre  $\vec{H}_I$  e a direção do pulso desejado, figura 3.3. Se o ângulo do cone do espaço de referência,  $\gamma_{ref}$ , for menor que o ângulo do cone do espaço atual,  $\gamma_{atual}$ , o comando pode ser de ACELERAÇÃO para o ângulo  $a$  menor do que 90 graus, com

$$\alpha_{pulso} = b + \pi, \quad (3.15)$$

ou o comando pode ser de CORREÇÃO para o ângulo  $a$  maior do que 90 graus, com

$$\alpha_{pulso} = b + \pi/2. \quad (3.16)$$

Se  $\gamma_{ref}$  for maior do que  $\gamma_{atual}$  o comando é de FREIO e o ângulo calculado é

$$\alpha_{pulso} = c - \pi/2. \quad (3.17)$$

A lógica do cálculo de  $\alpha_{pulso}$  é mostrada também na Figura 3.7.

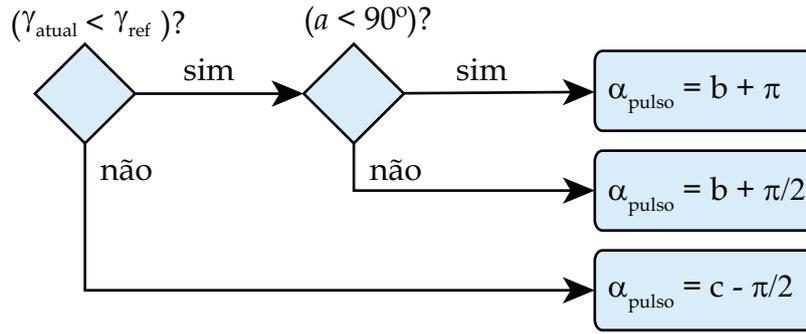


Figura 3.7 - Lógica direção dos pulsos

O ângulo do cone do espaço atual  $\gamma_{atual}$  é calcula pela Equação 3.8 e o ângulo do cone do espaço de referência  $\gamma_{ref}$  representa a distância da posição atual com a posição desejada e é calculado a partir da distância  $d$  com a equação

$$\gamma_{ref} = K_{cone}d, \quad (3.18)$$

em que  $K_{cone}$  é um ganho de controle.

É preciso calcular também o tempo de abertura do bocal para a aplicação da força no corpo, feito pela equação

$$t_{pulso} = K_{pulso}d/1000, \quad (3.19)$$

em que  $t_{pulso}$  é o tempo de abertura do bocal e  $K_{pulso}$  é um ganho de controle que relaciona a distância  $d$  com esse tempo. Como a Equação 3.19 pode gerar resultados de tempo de abertura menores que o tempo mínimo que os atuadores podem reproduzir é definido um tempo mínimo de acionamento.

A abertura de um bocal de gás frio gera uma força que continua agindo no corpo até que ele seja fechado e como o veículo está em rotação, a direção dessa força muda com o tempo. Para que essa força seja aplicada na direção desejada  $\alpha_{pulso}$ , calculada anteriormente, o controle deve adiantar seu acionamento para que o ponto médio do arco formado pelo movimento do bocal fique alinhado com  $\alpha_{pulso}$ . Nota-se que para arcos maiores há mais força sendo aplicada fora da direção desejada e como isso não é interessante, é definido um  $t_{max}$  que limita o comprimento do arco.

Na Figura 3.8 é ilustrado o movimento feito pelo bocal quando acionado. Em  $\phi_{inicial}$

o bocal é aberto e fechado quando chega em  $\phi_{final}$  varrendo um arco de comprimento  $\phi_{pulso}$  com velocidade  $\dot{\phi}$ . A direção desejada,  $\phi_{medio}$ , é definida pela bissetriz do arco formado.

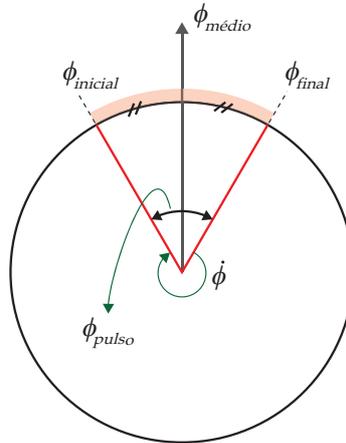


Figura 3.8 - Ângulo inicial e final de acionamento

Com o ângulo inicial e final de acionamento, o algoritmo da lógica de acionamento envia o comando de abertura para o bocal que estiver posicionado na região entre  $\phi_{inicial}$  e  $\phi_{final}$  e envia o comando de fechar para os bocais que estiverem fora desse intervalo. Um diagrama mostrando a sequência do algoritmo é visto na Figura 3.9.

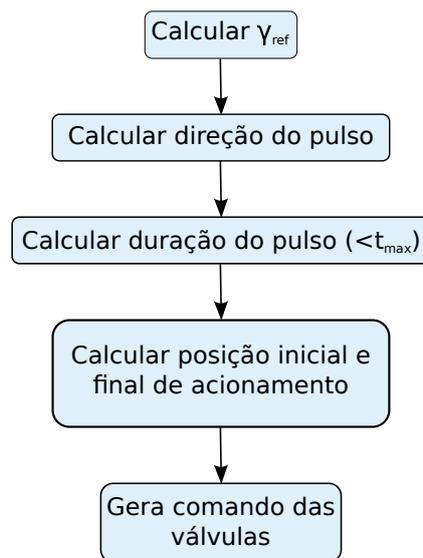


Figura 3.9 - Diagrama de funcionamento do algoritmo de controle

Por fim, o movimento esperado como resultado dessa manobra é mostrado na Figura 3.10.

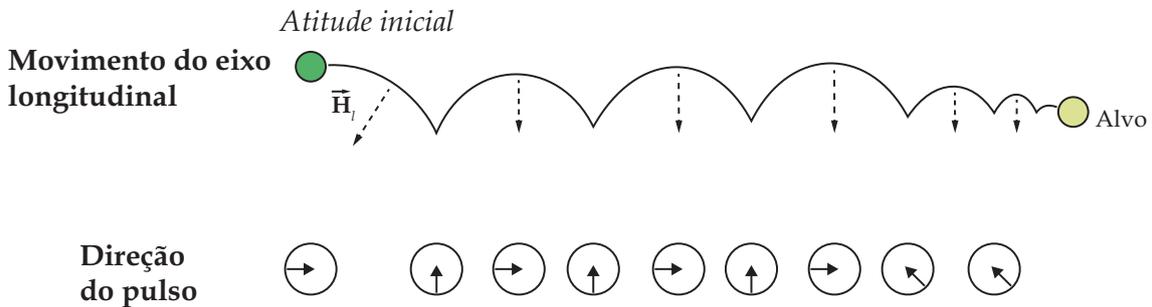


Figura 3.10 - Movimento precessional esperado

## 3.2 Controle 2

Esse algoritmo de controle é composto de duas partes que funcionam simultaneamente para poder corrigir a atitude do corpo. Uma primeira parte é o controle de precessão com o objetivo de levar o momento angular do veículo para as coordenadas do alvo. E uma segunda parte é o controle de nutação, já que cada acionamento de jato feito pelo controle de precessão induz no veículo um movimento de nutação que pode levar o sistema à instabilidade.

### 3.2.1 Controle de Precessão

É o controle responsável por levar o veículo ao alvo, assim a primeira parte do que se deve fazer é ler o erro de atitude do corpo em relação ao alvo desejado.

Usa-se aqui uma estratégia de calcular o erro de atitude através de matrizes de cossenos diretores (DCM - Direction Cosine Matrix) que é encontrado na literatura (SIDI, 1997). Esse método apresenta uma vantagem de não ser necessário ficar revendo e analisando o ângulo quanto a quadrante ou descontinuidade conseguindo ler diretamente o erro angular do veículo em relação ao alvo.

Supondo um sistema de coordenadas para o corpo do veículo e um sistema de coordenadas para o alvo, ambos relativos a um mesmo sistema de referência, existe uma matriz DCM,  $\mathbf{A}_S$ , que transforma do sistema de referência para o sistema do corpo e existe também uma matriz DCM,  $\mathbf{A}_T$ , que transforma do sistema de referência para o sistema do alvo. Um vetor  $\mathbf{a} = [a_1, a_2, a_3]^T$  pode ser expresso no sistema de corpo como  $\mathbf{a}_S$  e no sistema do alvo como  $\mathbf{a}_T$  na seguinte forma:

$$\mathbf{a}_S = [\mathbf{A}_S] \mathbf{a}, \quad (3.20)$$

$$\mathbf{a}_T = [\mathbf{A}_T] \mathbf{a}. \quad (3.21)$$

Combinando as duas equações, tem-se

$$\mathbf{a}_S = [\mathbf{A}_S] [\mathbf{A}_T]^{-1} \mathbf{a}_T = [\mathbf{A}_S] [\mathbf{A}_T]^T \mathbf{a}_T = [\mathbf{A}_e] \mathbf{a}_T. \quad (3.22)$$

A matriz  $\mathbf{A}_e$ , como definida na equação 3.22, tem o seguinte significado: se as componentes dos dois vetores  $\mathbf{a}$  são idênticos, então os sistemas de coordenadas do corpo e do alvo são coincidentes e, conseqüentemente, o veículo está na atitude comandada. Para um melhor entendimento, a forma explícita da multiplicação da matriz é apresentada na equação 3.23.

$$\mathbf{A}_e = \begin{bmatrix} a_{11S} & a_{12S} & a_{13S} \\ a_{21S} & a_{22S} & a_{23S} \\ a_{31S} & a_{32S} & a_{33S} \end{bmatrix} \begin{bmatrix} a_{11T} & a_{21T} & a_{31T} \\ a_{12T} & a_{22T} & a_{32T} \\ a_{13T} & a_{23T} & a_{33T} \end{bmatrix} = \begin{bmatrix} a_{11E} & a_{12E} & a_{13E} \\ a_{21E} & a_{22E} & a_{23E} \\ a_{31E} & a_{32E} & a_{33E} \end{bmatrix}. \quad (3.23)$$

Nota-se que os elementos da última matriz da equação 3.23 são resultantes de produtos escalares entre eixos do sistema de coordenadas do corpo e eixos do sistema de coordenadas do alvo. Por exemplo:  $a_{12E}$  é o produto escalar entre os eixos  $\mathbf{X}$  do corpo do veículo e do eixo  $\mathbf{Y}$  do alvo, seguindo o raciocínio tem-se que

$$a_{12E} = \mathbf{X}_S \cdot \mathbf{Y}_T, \quad (3.24)$$

$$a_{13E} = \mathbf{X}_S \cdot \mathbf{Z}_T, \quad (3.25)$$

$$a_{23E} = \mathbf{Y}_S \cdot \mathbf{Z}_T \quad (3.26)$$

em que  $\mathbf{X}_S$  e  $\mathbf{Y}_S$  são os eixos  $\mathbf{X}$  e  $\mathbf{Y}$  do sistema do corpo e  $\mathbf{Y}_T$  e  $\mathbf{Z}_T$  são os eixos  $\mathbf{Y}$  e  $\mathbf{Z}$  do sistema do alvo.

Os valores de  $a_{12E}$ ,  $a_{13E}$  e  $a_{23E}$  são grandezas proporcionais ao erro angular dos eixos dos sistemas do corpo do veículo e do alvo, já que o resultado do produto escalar tem uma relação direta com o ângulo entre dois vetores.

Com esses valores proporcionais ao erro angular, uma lei de controle PD pode ser proposta com a seguinte forma:

$$\begin{aligned} T_{cy} &= K_p a_{13E} + K_d \omega_y, \\ T_{cz} &= K_p a_{12E} + K_d \omega_z \end{aligned} \quad (3.27)$$

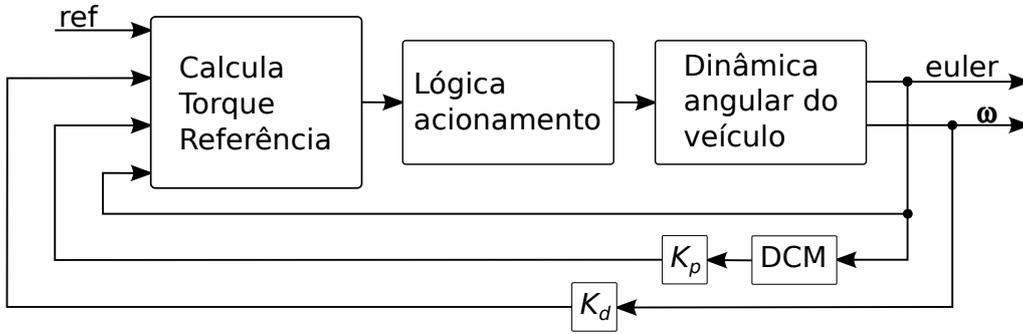


Figura 3.11 - Diagrama do controle de precessão

em que  $K_p$  é o ganho proporcional,  $K_d$  é o ganho derivativo e  $\omega_x, \omega_y, \omega_z$  as velocidades angulares do veículo. Como o veículo não possui sistema de atuação que gere torque em seu eixo longitudinal, o termo  $T_{cx}$  é desnecessário.

O controle de precessão funciona como mostra o diagrama da figura 3.11, são lidos os ângulos de Euler e montada a matriz DCM, os elementos da matriz são multiplicados pelo ganho  $K_p$  e a solução, juntamente com as velocidades angulares multiplicadas pelo ganho  $K_d$ , alimentam o bloco que calcula o vetor de torque de referência. Esse torque é usado na lógica de acionamento que gera os comandos de atuação das válvulas de gás frio.

Com o veículo sempre em rotação, o sistema do corpo está também girando, então a matriz DCM do alvo foi montada de forma a compensar essa rotação substituindo o ângulo de Euler do eixo  $\mathbf{X}$  do sistema do alvo pelo do sistema do corpo, assim a DCM do alvo é montada com  $\psi_T, \theta_T$  e  $\phi_S$  em que  $\psi_T$  e  $\theta_T$  são os ângulos de Euler dos eixos  $\mathbf{X}$  e  $\mathbf{Y}$  do sistema do alvo e  $\phi_S$  é o ângulo de Euler do eixo  $\mathbf{X}$  do sistema do veículo.

Os atuadores são válvulas ON/OFF e, sendo assim, não é possível gerar torques de forma proporcional ao erro aproveitando essa lei de controle diretamente, porém, é possível criar um vetor que aponta para a direção em que o torque precisa ser aplicado com os valores calculados de  $T_{cy}$  e  $T_{cz}$ .

A figura 3.12a ilustra a situação em que o corpo em uma posição  $\mathbf{P}$  enxerga o alvo

A, os erros angulares entre a posição atual e o alvo são  $\Delta\psi$  e  $\Delta\theta$  proporcionais a  $a_{12e}$  e  $a_{13e}$  que são elementos da matriz DCM de erro. Calculando os torques conforme a equação 3.27 é obtido o vetor de torque  $\vec{T}_c$ , então o acionamento do jato deve gerar a força  $\vec{F}_c$  mostrada na figura 3.12b.

Para que o acionamento do jato gere a força  $\vec{F}_c$  é preciso que ele seja comandado na posição oposta da força, já que o que acontece na realidade é uma força de reação quando o jato é ligado. Como acontece no caso do Controle 1, quando o acionamento é feito o jato faz uma força durante um período percorrendo um arco. Na figura 3.13a é criado um vetor  $\vec{A}_i$  no início do arco e um vetor  $\vec{A}_f$  no fim dele delimitando a região de acionamento e a figura 3.13b ilustra o movimento dos jatos até o instante de acionamento do jato  $J_3$ .

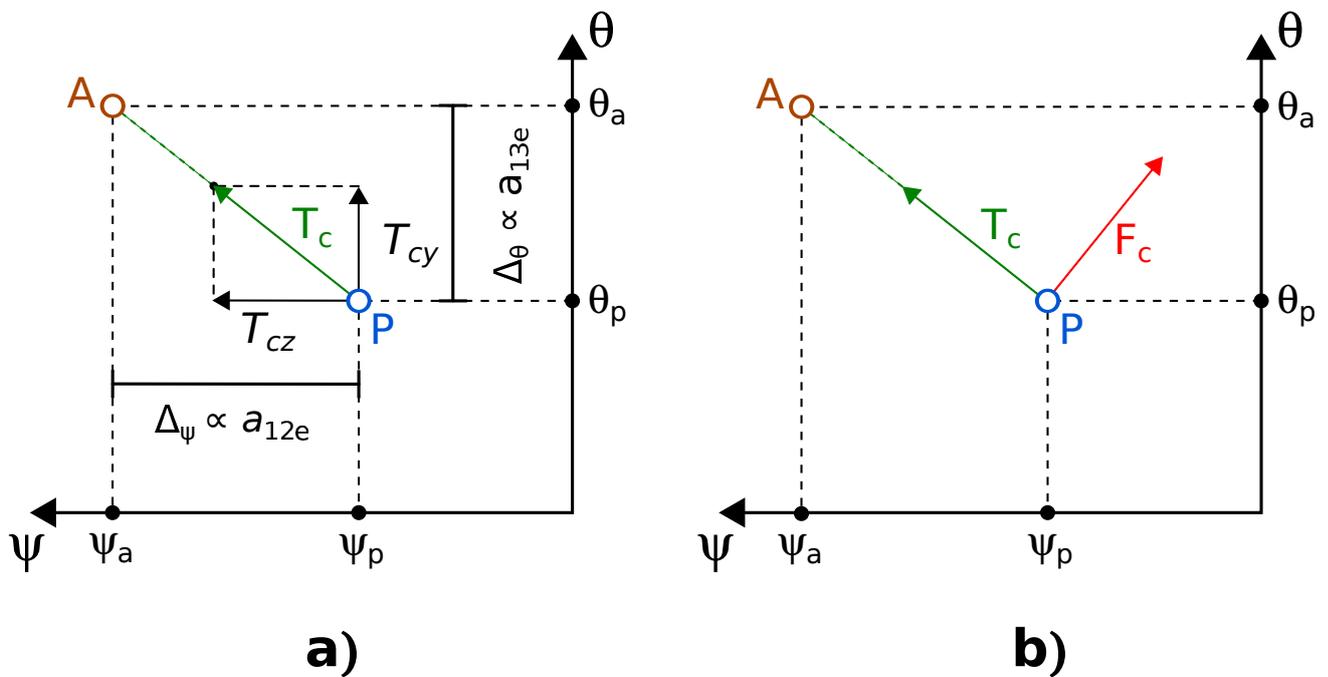


Figura 3.12 - Ilustração vetor torque de controle  $T_c$

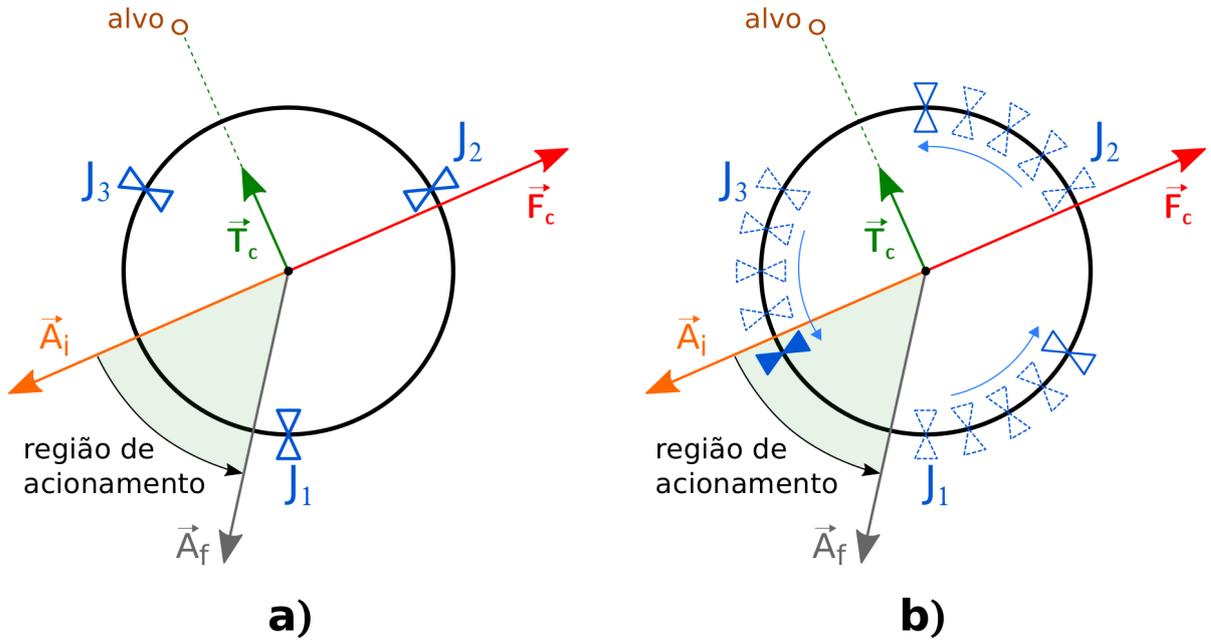


Figura 3.13 - Região de acionamento dos jatos

### 3.2.2 Controle de Nutação

Um corpo girando em torno de seu eixo de menor ou maior momento de inércia é dinamicamente estável na ausência de dissipação de energia interna. Porém, se há dissipação de energia então o corpo girando no eixo de menor momento de inércia se torna dinamicamente instável. Como para cada comando de jato há a dissipação de energia, é preciso fazer um controle de nutação para manter o movimento estável enquanto há comandos de manobra do controle de precessão.

Com o corpo girando em torno o eixo  $\mathbf{X}_S$ , uma lei de controle simples é limitar sua velocidade angular máxima. Nesse caso, são medidas as velocidades angulares e comandado um torque contrário ao movimento sempre que o valor limite for excedido. O comando é gerado por:

$$\begin{aligned} T_{cy} &= -F\Delta \operatorname{sign}(\omega_y), \\ T_{cz} &= -F\Delta \operatorname{sign}(\omega_z) \end{aligned} \tag{3.28}$$

em que  $\Delta$  é o braço do torque gerado pelo atuador. Como os atuadores não estão alinhados com os eixos  $\mathbf{Y}_S$  e  $\mathbf{Z}_S$  é preciso fazer uma transformação de coordenadas, calculando a projeção da velocidade angular na direção de cada atuador, como

mostrado na figura 3.14. Assim, a lei de controle passa a ser:

$$\begin{aligned} T_{J_1} &= -F\Delta \operatorname{sign}(\omega_{J_1}), \\ T_{J_2} &= -F\Delta \operatorname{sign}(\omega_{J_2}), \\ T_{J_3} &= -F\Delta \operatorname{sign}(\omega_{J_3}). \end{aligned} \quad (3.29)$$

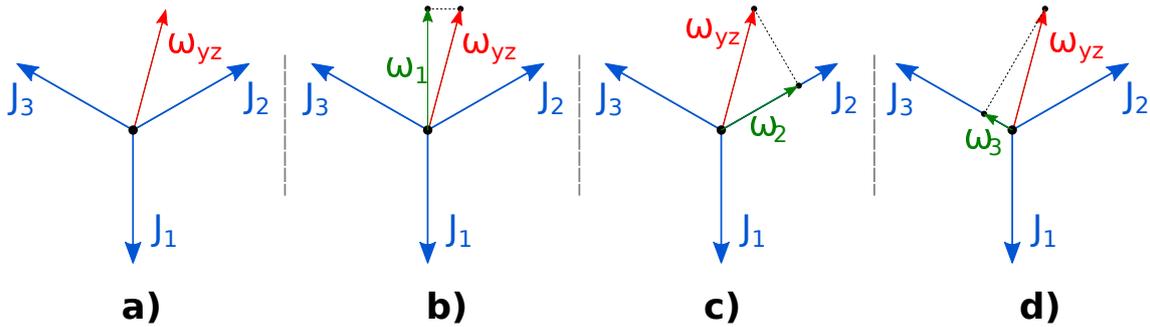


Figura 3.14 - Transformação vetor velocidade angular

Então o controle de nutação funciona como mostra a ilustração da figura 3.15, onde são lidas as velocidades angulares que alimentam o bloco com a lógica de acionamento. Os comandos executados geram o torque de correção para evitar que o ângulo de cone cresça indefinidamente.

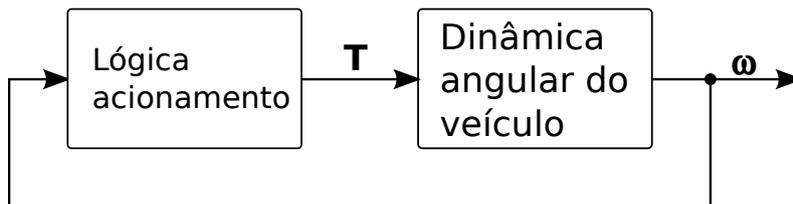


Figura 3.15 - Diagrama do controle de nutação

## 4 SIMULAÇÃO

Essa seção descreve os blocos desenvolvidos para o programa de simulação 3 DoF.

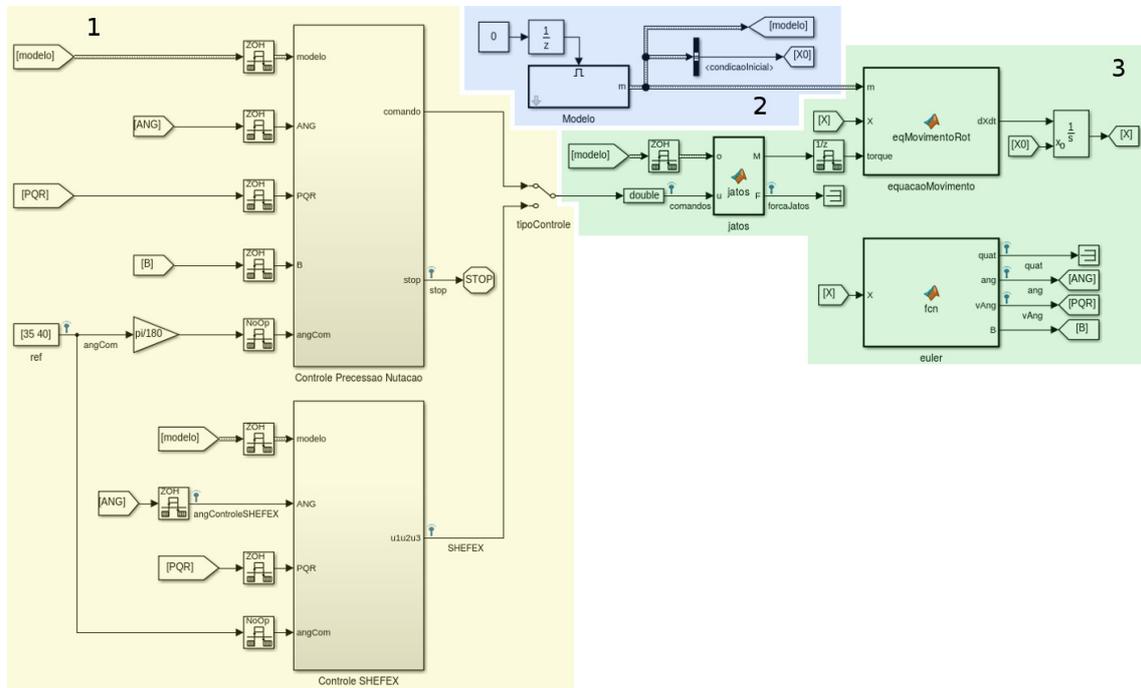


Figura 4.1 - Blocos da simulação em Simulink

### 4.1 Descrição dos blocos

A Figura 4.1 mostra a tela inicial do programa de simulação, destacando três regiões: a região 1 onde estão os blocos com os algoritmos de controle, a região 2 onde estão os blocos com a inicialização da simulação e a região 3 com os blocos da dinâmica e integração numérica.

#### 4.1.1 Região 1

A região 1 do programa de simulação, com imagem ampliada na figura 4.2, é onde estão programados os algoritmos de controle, separados por blocos de sub-sistema com saída padronizada. Para escolher qual algoritmo deve ser usado na simulação foi colocado um bloco *Manual Switch* com nome "tipoControle".

As entradas dos algoritmos de controle são semelhantes, ambos são alimentados com as variáveis "modelo", com os dados particulares da simulação, "ANG", com os valores dos ângulos de Euler  $\phi$ ,  $\theta$  e  $\psi$ , "PQR", com os valores de velocidades

angulares nos eixos  $X$ ,  $Y$  e  $Z$  e com o vetor de alvo com os valores que devem ser seguidos na manobra de apontamento.

No controle 2 há ainda uma entrada para a variável "B" que é a matriz de cossenos diretores para a iteração atual.

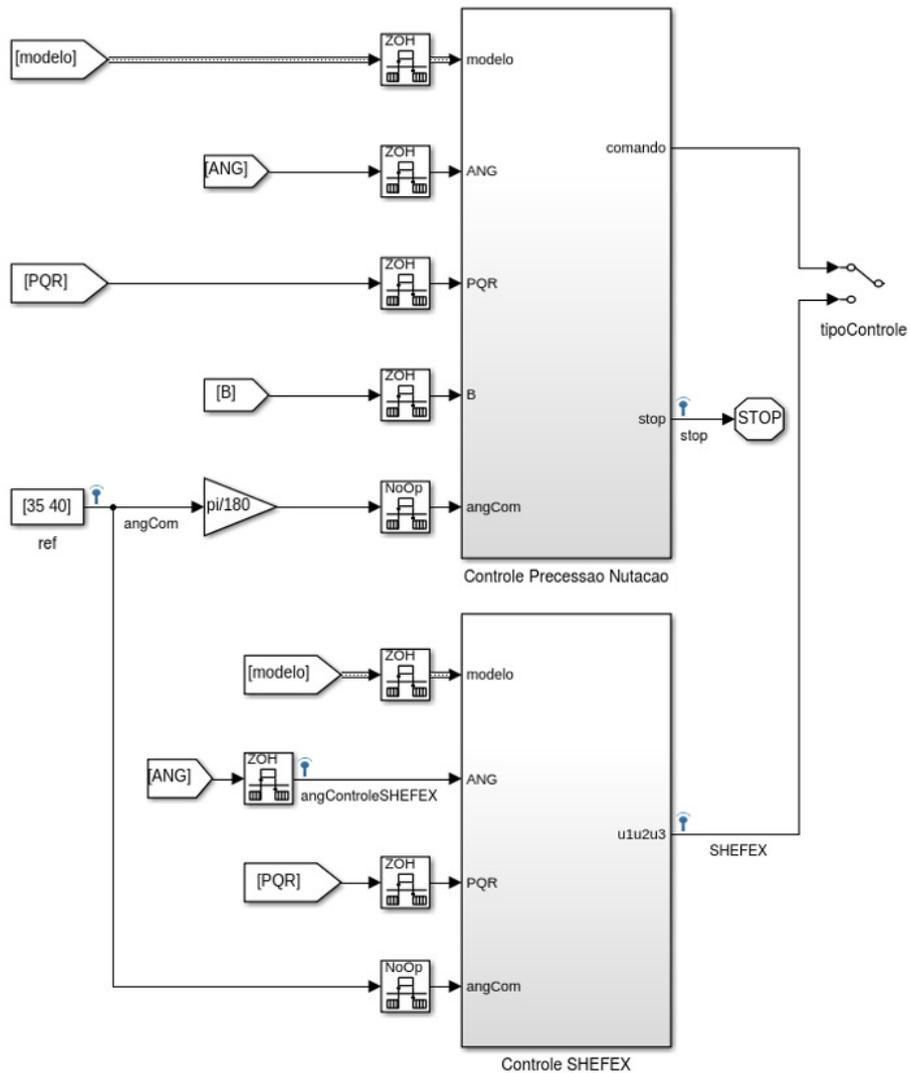


Figura 4.2 - Blocos Simulink - Região 1

O Controle 1 está programado no sub-sistema "Controle SHEFEX" e dentro do sub-sistema há dois blocos do tipo *MATLAB Function* chamados "Param\_controle" e "Controle", visto na figura 4.3.

Em "Param\_controle" são lidas as posições e velocidades angulares, as coordenadas angulares comandadas e as características de inercia do corpo. Nesse bloco são calculados os valores dos parâmetros  $a$ ,  $b$ ,  $c$  e  $d$  como mostrado na sessão 3.1.3.1, as posições angulares de cada atuador e o ângulo do Cone de espaço atual calculado pela equação 3.8.

No bloco "Controle" são recebidos os resultados de "Param\_controle" e ainda um barramento de dados "param" com os valores de tempo mínimo e máximo de acionamento do atuador, os ganhos  $K_{pulso}$  e  $K_{cone}$ , o valor máximo de ângulo de cone de espaço e a frequência de rotação do veículo que são usados no algoritmo de controle. O resultado desse bloco é um vetor "comando" de três posições com valores de 0 para não acionar ou 1 para acionar o jato, onde cada elemento representa um atuador. Todas as outras saídas do bloco são cálculos auxiliares que são usados na geração de figura e análise da simulação.

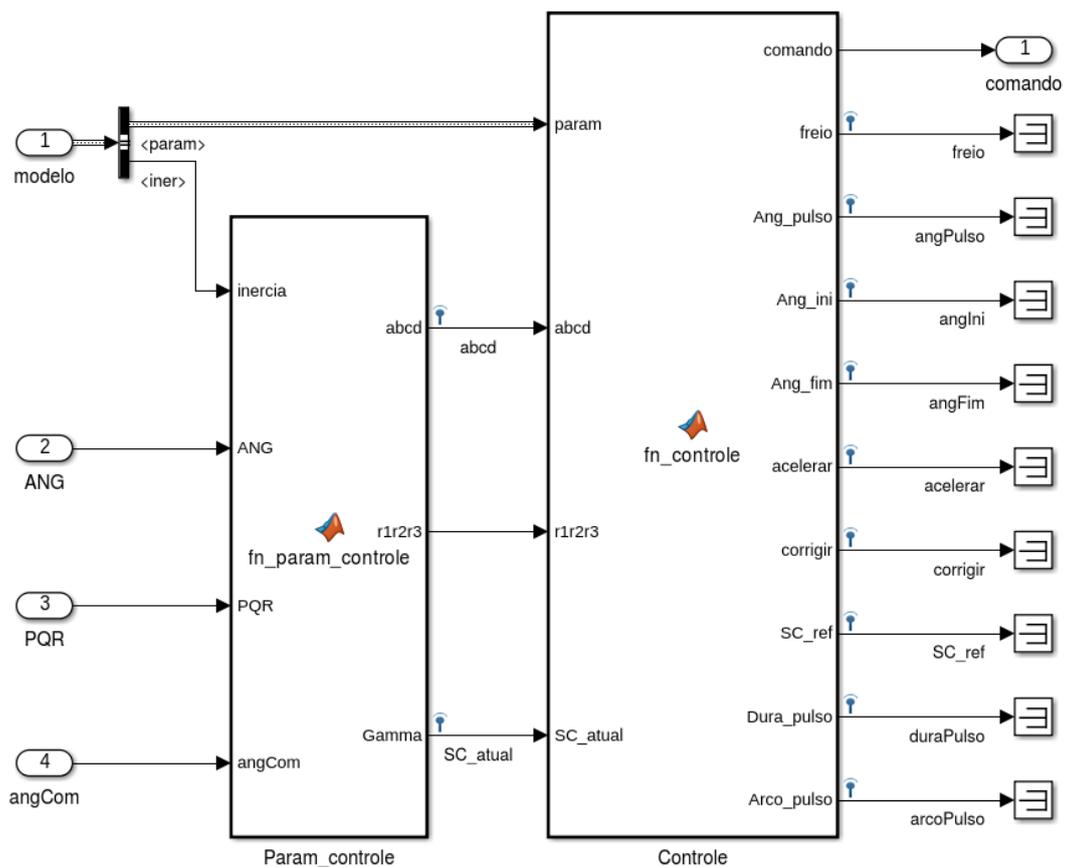


Figura 4.3 - Sub-sistema Controle 1

O algoritmo de controle apresentado na sessão 3.2 está implementado no bloco de sub-sistema "Controle Pecessao Nutacao", mostrado na figura 4.2, e expandindo esse sub-sistema é possível ver os blocos que o compõe, apresentados na figura 4.4.

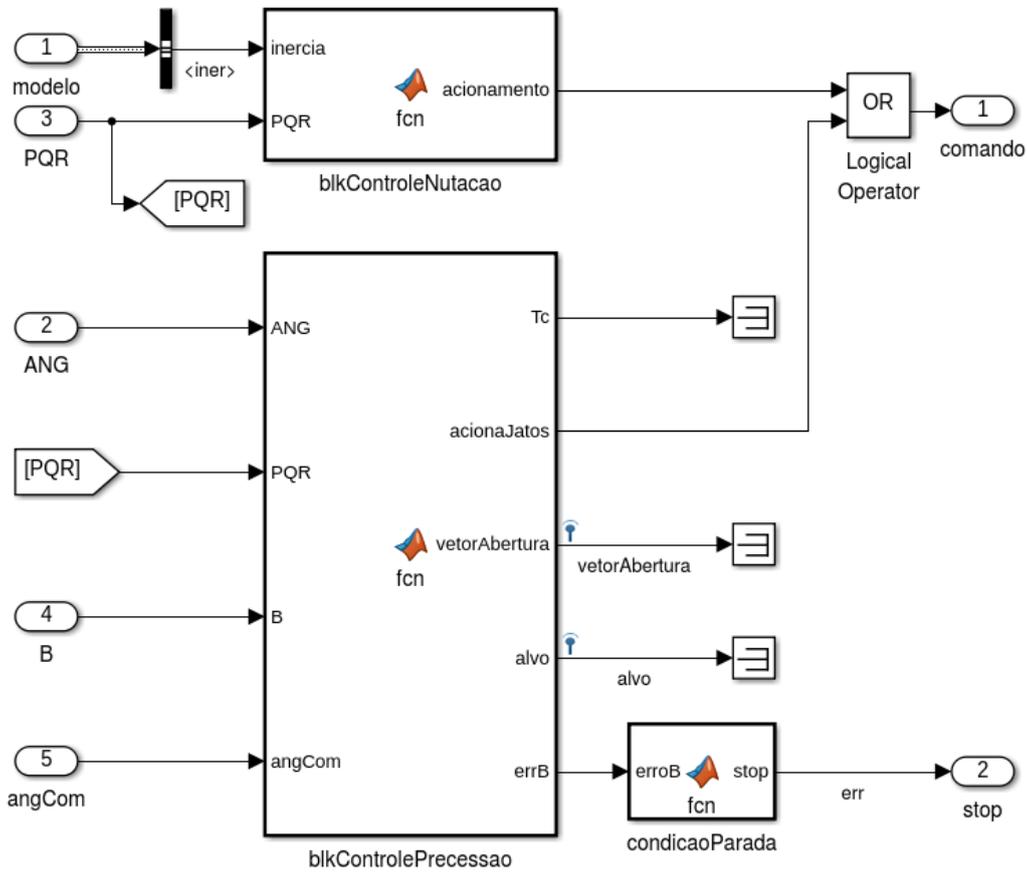


Figura 4.4 - Sub-sistema Controle Pecessao Nutacao

Da figura 4.2 nota-se que as entradas para os dois algoritmos de controle são idênticas, exceto pela entrada "B" no bloco "Controle Pecessao Nutacao", em que "B" é a matriz cossenos diretores que transforma um vetor no sistema de coordenadas inercial para o sistema de coordenadas do corpo. Isso acontece porque é preciso calcular a matriz de erro através da equação 3.23 e externamente ao subsistema a matriz "B" já está sendo calculada, então essa entrada extra no algoritmo serve apenas para aproveitar a variável já calculada.

Esse algoritmo foi dividido em dois blocos, um para fazer o controle da precessão e um bloco para fazer o de nutação e os seus códigos são mostrados nos anexos C e D.

Como as saídas dos dois algoritmos de controle foram padronizadas para facilitar a mudança entre um e outro na simulação, a resposta desse bloco também é um vetor de três elementos com valores de 0 ou 1. Porém, nesse caso há duas leis de controle agindo simultaneamente que são combinados através de um operador lógico "OR".

#### 4.1.2 Região 2

Na região 2, ampliada na figura 4.5, é computada a condição inicial, assim como toda a característica de massa e parâmetros gerais necessários para o controle. O bloco "Modelo" está configurada com uma máscara para que os dados sejam inseridos através de uma interface gráfica mais simples, figura 4.6, e seu código pode ser visto no anexo A .

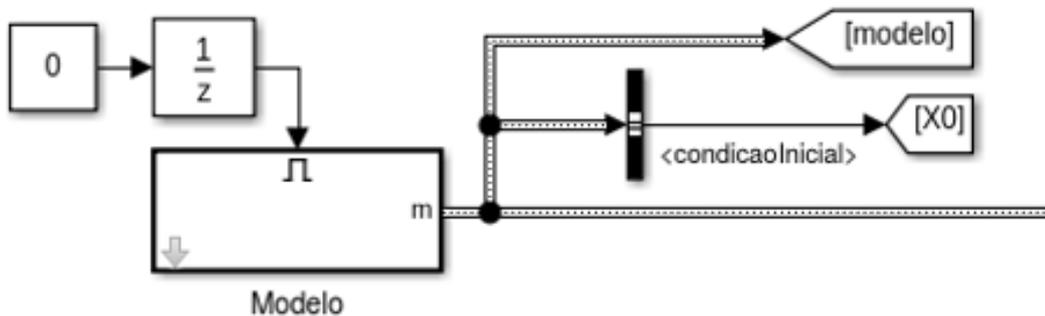


Figura 4.5 - Blocos Simulink - Região 2

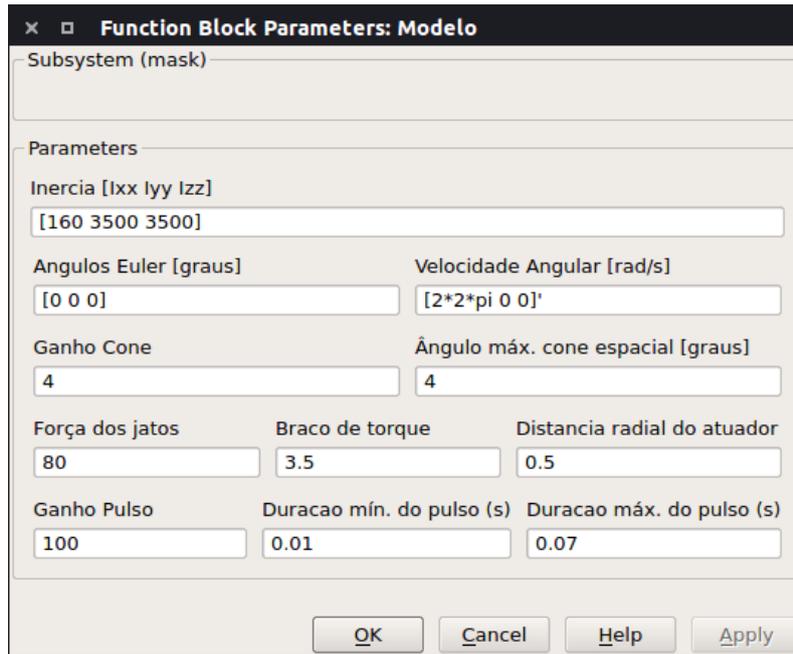


Figura 4.6 - Parâmetros da máscara do bloco Modelo

### 4.1.3 Região 3

Na região 3, figura 4.7, é onde os comandos de acionamento dos atuadores são transformados em Torques agindo no corpo. Isso alimenta o bloco "equacaoMovimento" que calcula as equações diferenciais da atitude do veículo, então o resultado desse bloco passa por um integrador numérico que calcula os estados para o próximo ciclo de simulação. O bloco "euler" existe simplesmente para separar os valores de posição e velocidade angular e calcular a matriz  $\mathbf{B}$  de cossenos diretores que transforma o vetor do sistema de coordenadas inercial para o do corpo.

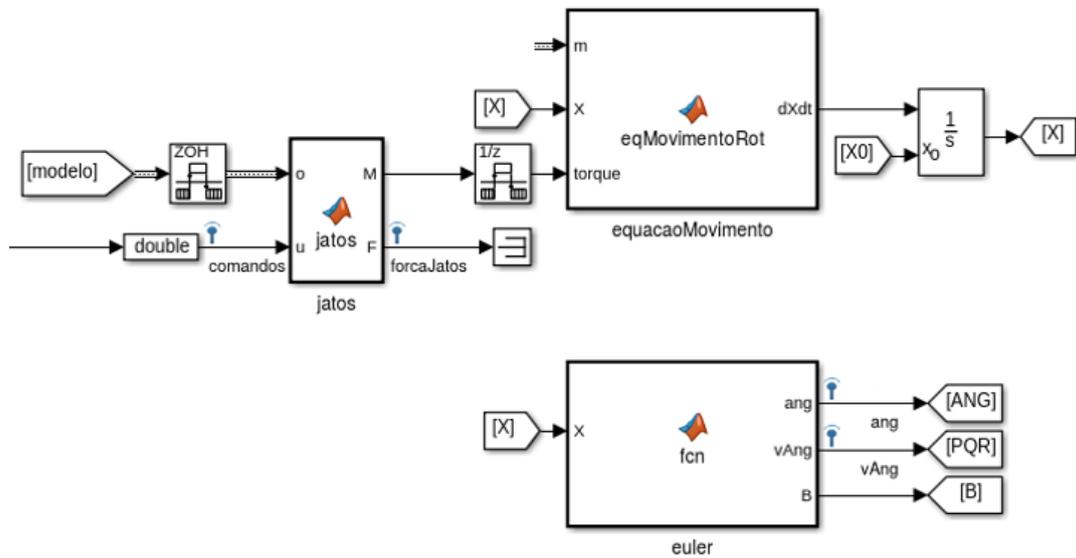


Figura 4.7 - Blocos Simulink - Região 3

## 4.2 Resultados da simulação

### 4.2.1 Valores numéricos

Para a simulação realizada foram utilizadas as propriedades físicas mostradas na Tabela 4.1, esses valores são inseridos diretamente no bloco "Modelo" através da janela da figura 4.6.

Tabela 4.1 - Propriedades físicas do experimento

Propriedade	Valor
Momento de inércia longitudinal ( $I_x$ )	160 $kgm^2$
Momento de inércia lateral ( $I_l$ )	3500 $kgm^2$
Raio do cilindro	0.5 $m$
Distância do CG ao plano dos Bocais	3.5 $m$
Empuxo gerado por atuador	80 $N$
Velocidade de rotação	2 $Hz$
Tempo mínimo de abertura do bocal	10 $ms$
Tempo máximo de abertura do bocal	70 $ms$
Ângulo de cone do espaço máximo	4 $graus$
Ganho $K_{cone}$	4
Ganho $K_{pulso}$	100

O *solver* do simulador funciona com um passo de integração de 0.0025 segundos ( $400Hz$ ) e período de discretização do bloco de controle é de 0.01 segundos para emular a taxa de atualização de  $100Hz$  do computador de bordo.

#### 4.2.2 Controle 1

Partindo a simulação na condição inicial

$$X_0 = \begin{bmatrix} \phi_0 \\ \theta_0 \\ \psi_0 \\ \omega_{x0} \\ \omega_{y0} \\ \omega_{z0} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 4\pi \\ 0 \\ 0 \end{bmatrix},$$

e definindo a posição do alvo como

$$\theta_{alvo} = 13^\circ \quad \text{e} \quad \psi_{alvo} = 15^\circ,$$

obtém-se, após 100 segundos de simulação, o resultado da figura 4.8, visto também na figura 4.9 com uma aproximação na região do alvo mostra o movimento angular do veículo até chegar ao alvo definido no início da simulação. Mostra-se no eixo  $X$  os valores de  $\psi$  e no eixo  $Y$  os valores de  $\theta$  e, na figura 4.8 há marcações com bolas pretas a cada 5 segundos de simulação para que se tenha a noção de tempo associada ao deslocamento. O rastro angular que a manobra faz está impresso na figura em 3 cores, cada cor para um modo de comando: aceleração é azul, correção é verde e freio é vermelho.

Esse resultado mostra que o alvo foi atingido, porém o movimento é diferente do esperado inicialmente. No início do movimento os ângulos  $\theta$  e  $\psi$  crescem quase linearmente e, conforme o veículo ganha velocidade a tendência de nutar do corpo faz com que a tendência do movimento não continue apontada para o alvo.

A Figura 4.10 mostra o ângulo de cone do espaço atual ( $\gamma_{atual}$ ) calculado durante os primeiros 40 segundos de simulação. O valor de  $\gamma_{atual}$  começa em zero e cresce até próximo ao valor máximo até começar a diminuir um pouco antes da marca de 5 segundos. Essa redução no valor acontece porque essa fase tem as características mostradas na figura 3.3, então estão sendo acionados os atuadores em modo de Correção que gera um torque que reduz o vetor momento angular  $\vec{H}_l$ .

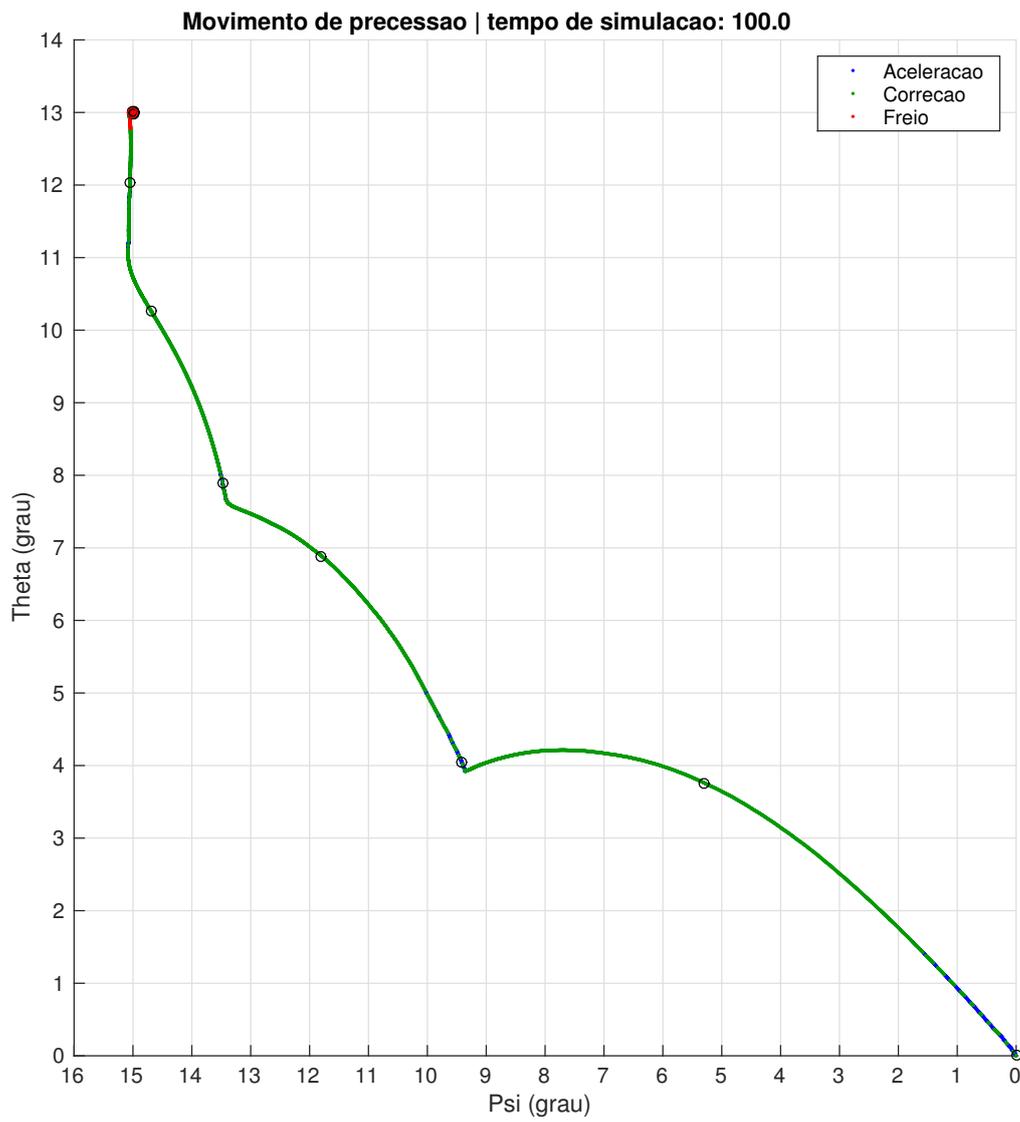


Figura 4.8 - Movimento de precessão para o alvo  $\theta: 13^\circ$  e  $\psi: 15^\circ$

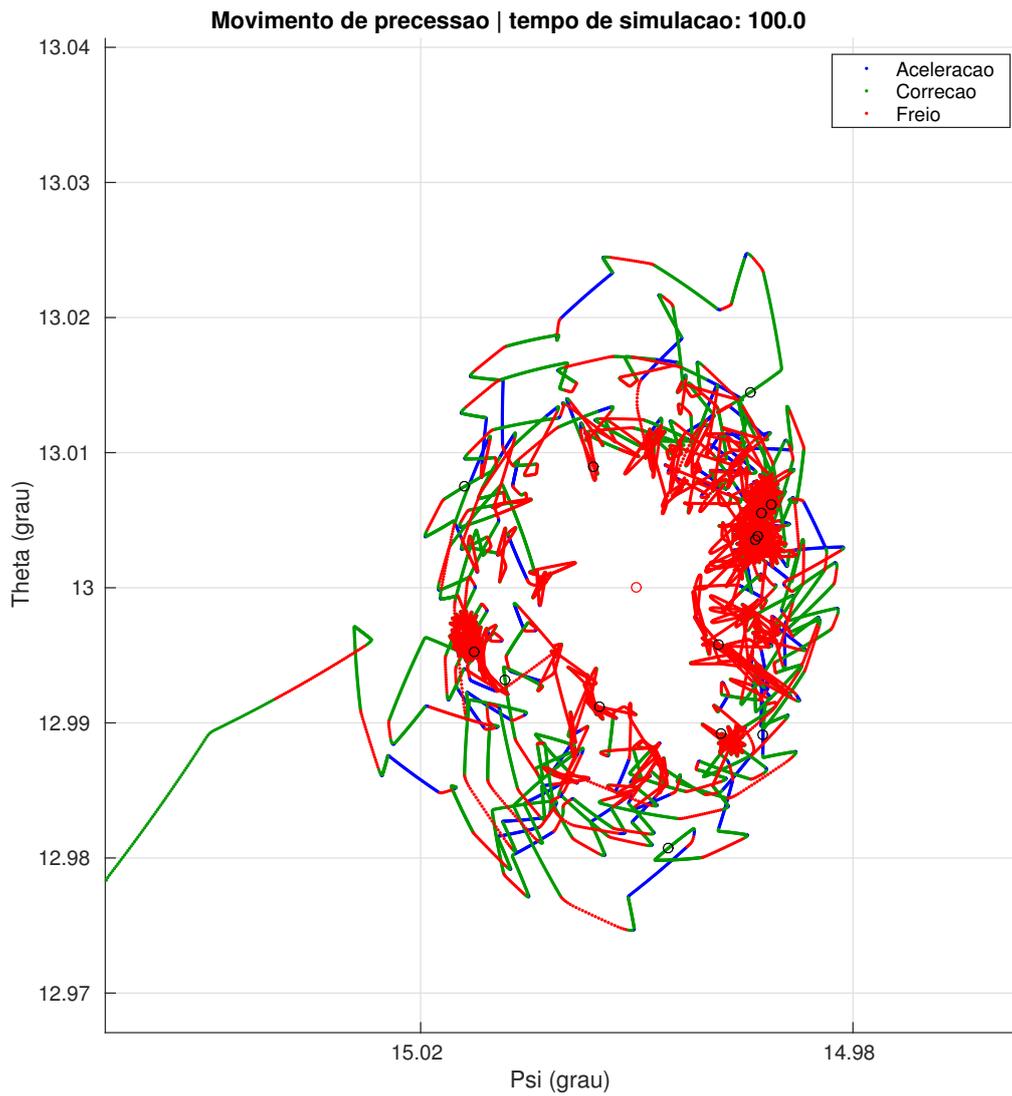


Figura 4.9 - Movimento de precessão para o alvo  $\theta: 13^\circ$  e  $\psi: 15^\circ$  na região do alvo

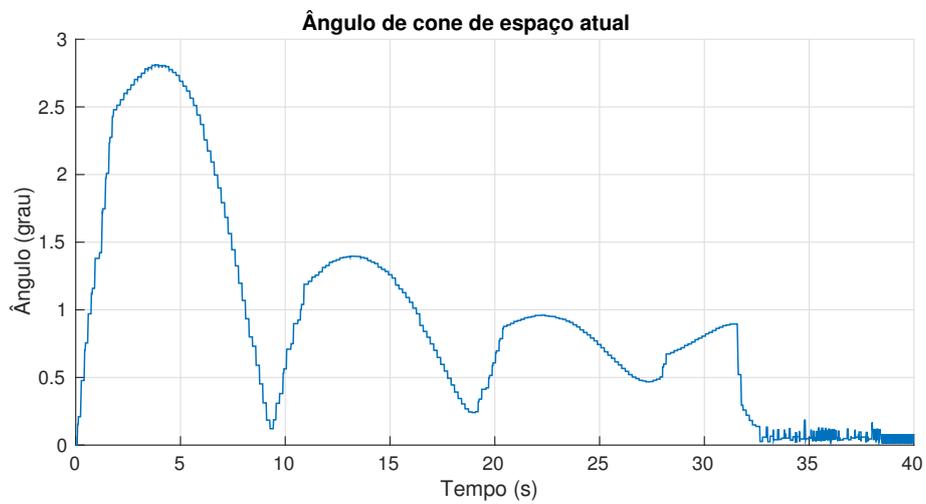


Figura 4.10 - Ângulo de cone  $\gamma$  para  $\theta_{alvo} = 13^\circ$  e  $\psi_{alvo} = 15^\circ$  - Controle 1

Um problema que foi encontrado ainda nas fases iniciais das simulações, (CAVALCANTE, 2013), e estudado mais profundamente é que para alguns alvos escolhidos o controle não é capaz de atingir a atitude desejada, um exemplo é o caso do alvo  $\psi_{alvo} = 13^\circ$  e  $\theta_{alvo} = 15^\circ$ .

O resultado, visto em gráficos nas figuras 4.11 e 4.12, para outros alvos, mostra que o comportamento no início da manobra é semelhante à simulação anterior, porém quando o alvo está próximo, os comandos deixam a atitude instável e os ângulos  $\theta$  e  $\psi$  divergem da região do alvo.

A Figura 4.13 mostra o resultado do cálculo do ângulo de cone de espaço para o alvo  $\theta_{alvo} = 15^\circ$  e  $\psi_{alvo} = 13^\circ$ . O comportamento desse caso é semelhante ao do anterior até o instante de aproximação do alvo, no tempo de simulação próximo a 30 segundos, quando o ângulo de cone deveria reduzir seu valor e permanecer próximo a 0. O que acontece porém é um aumento do ângulo de cone até o seu valor máximo de 4 graus.

O estudo mais aprofundado do problema mostrou que há uma condição de operação em que o algoritmo gera comandos que instabilizam o veículo. O comando de correção, figura 3.3, que tem o objetivo de realinhar o vetor momento angular lateral  $\vec{H}_l$ , pode mudar a direção desse vetor no sentido errado, fazendo com que a velocidade  $\vec{v}$  também fique na direção errada.

O ângulo  $a$ , figura 3.4, é calculado como mostra o material de referência, figura 4.14, onde os limites de  $a$  é estabelecido entre  $0^\circ$  e  $360^\circ$  e o comportamento anormal acontece quando  $a$  é maior que  $180^\circ$ .

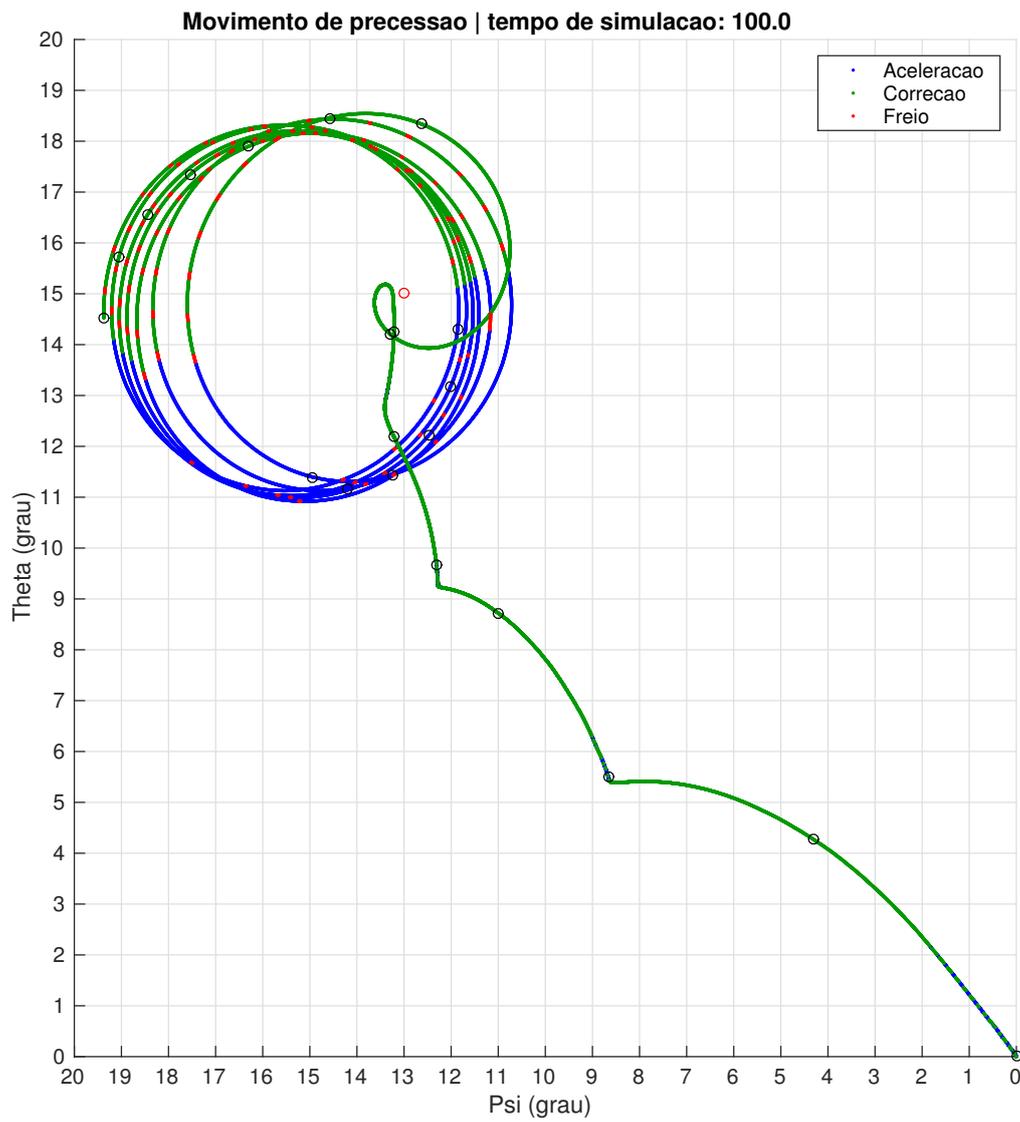


Figura 4.11 - Movimento de precessão para o alvo  $\theta: 15^\circ$  e  $\psi: 10^\circ$

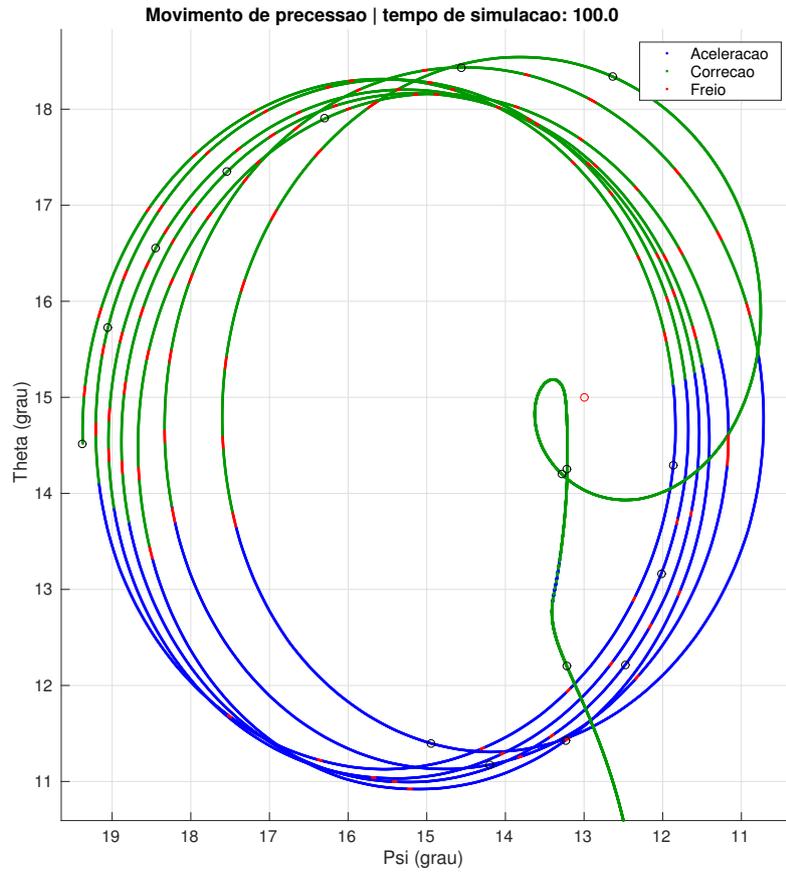


Figura 4.12 - Movimento de precessão para o alvo  $\theta: 15^\circ$  e  $\psi: 13^\circ$  na região do alvo

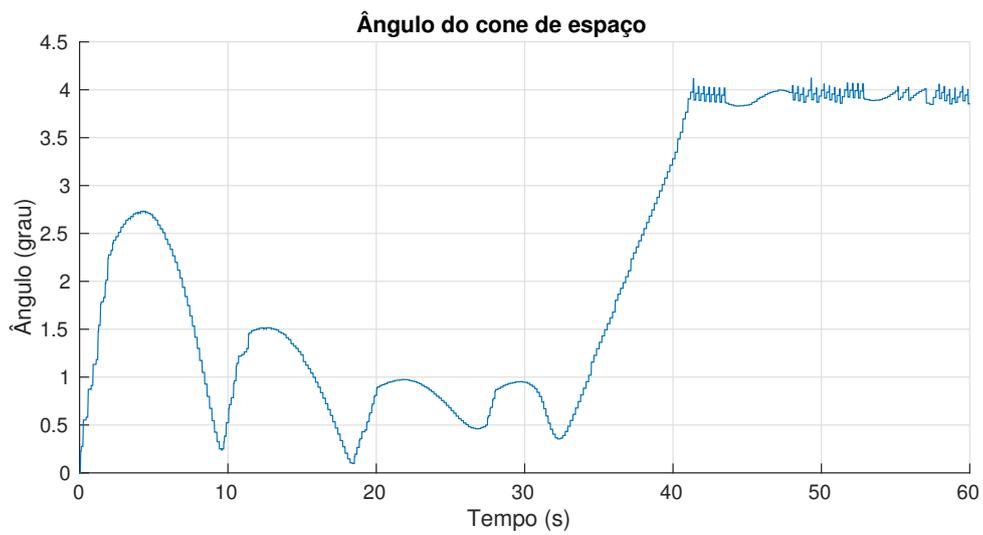


Figura 4.13 - Ângulo de cone  $\gamma$  para  $\theta_{alvo} = 15^\circ$  e  $\psi_{alvo} = 13^\circ$  - Controle 1

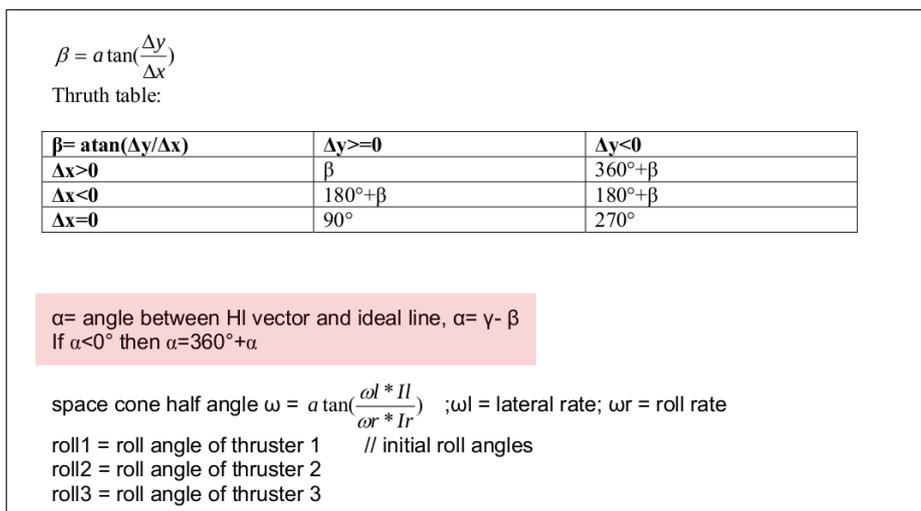


Figura 4.14 - Ajuste de ângulo  $a$ , (ETTL, 2009)

Para entender melhor esse comportamento, o comando de correção é dividido em 3 casos, ilustrados na figura 4.15 e descritos a seguir:

\* Caso **a**: com o ângulo  $a$  maior que  $90^\circ$  e menor que  $180^\circ$ , o acionamento do jato gera um torque que além de reduzir a magnitude do vetor momento angular lateral  $\vec{H}_l$  também o direciona para o sentido de reduzir o ângulo  $a$  e retorná-lo ao valor de  $90^\circ$ .

\* Caso **b**: com o ângulo  $a$  maior que  $180^\circ$  e menor que  $270^\circ$ , o acionamento do jato gera um torque que, como o caso **a**, reduz a magnitude de  $\vec{H}_l$ , porém o momento angular é direcionado para um sentido que faz o ângulo  $a$  crescer mais e não voltar para  $90^\circ$  como desejado.

\* Caso **c**: com o ângulo  $a$  maior que  $270^\circ$  e menor que  $360^\circ$ , o acionamento do jato gera um torque que aumenta a magnitude de  $\vec{H}_l$  e ainda faz com que o ângulo  $a$  aumente.

Assim, quando o veículo está na região do ângulo de referência  $a$  entre  $180^\circ$  e  $270^\circ$ , os comandos de Correção mudam a direção do vetor  $\vec{H}_l$  no sentido contrário do desejado e quando o ângulo  $a$  tem um valor maior que  $270^\circ$  os acionamentos aumentam a magnitude de  $\vec{H}_l$  aumentando também o ângulo de cone do espaço  $\gamma$  e o movimento de nutação instabilizando o veículo e criando a trajetória vista na figura 4.12.

O ângulo  $a$  ao longo dos 100 segundos de simulação é mostrado na figura 4.16 com

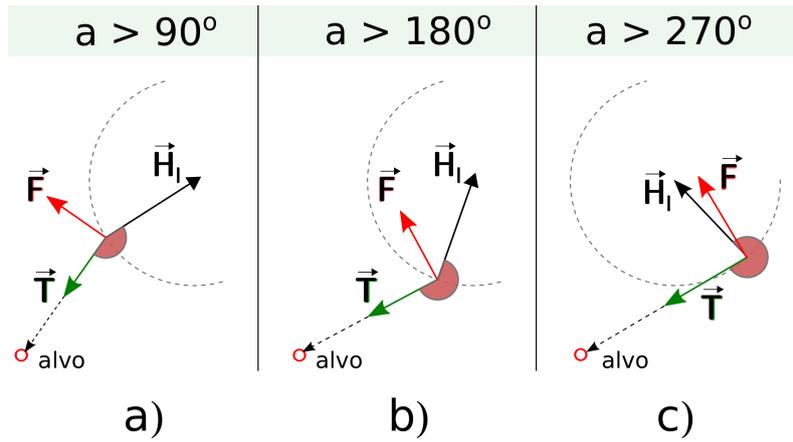


Figura 4.15 - Comportamento comando de correção

os valores angulares maiores que  $180^\circ$  e menores que  $360^\circ$  destacados em vermelho.

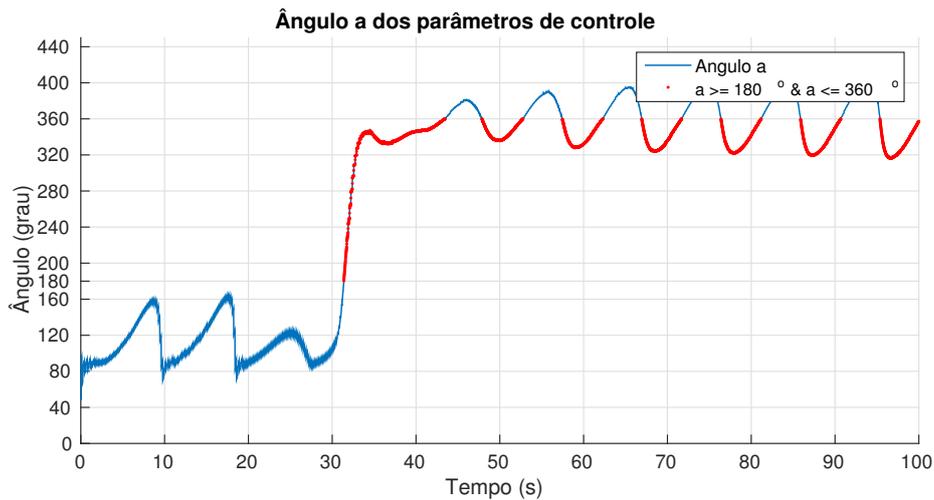


Figura 4.16 - Ângulo  $a$  no tempo para o caso  $\theta: 15^\circ$  e  $\psi: 13^\circ$

Uma possível forma de correção desse problema é mudar os comandos de correção para um modo de comando que provoque as mudanças no vetor  $\vec{H}_l$  para que esse se alinhe com a linha perpendicular do vetor distância e que diminua o módulo de  $\vec{H}_l$ , reduzindo o ângulo  $\gamma$ .

Analisando o efeito do comando de aceleração para a região de instabilidade, figura 4.15, é possível notar que:

\* Caso **a**: com o ângulo  $a$  maior que  $180^\circ$  e menor que  $270^\circ$ , o acionamento do jato

gera um torque que reduz a magnitude de  $\vec{H}_l$  e o direciona para um sentido que faz o ângulo  $a$  diminuir e retornar ao valor de  $90^\circ$  como desejado.

\* Caso **b**: com o ângulo  $a$  maior que  $270^\circ$  e menor que  $360^\circ$ , o acionamento do jato gera um torque que reduz a magnitude de  $\vec{H}_l$  freando a manobra de atitude.

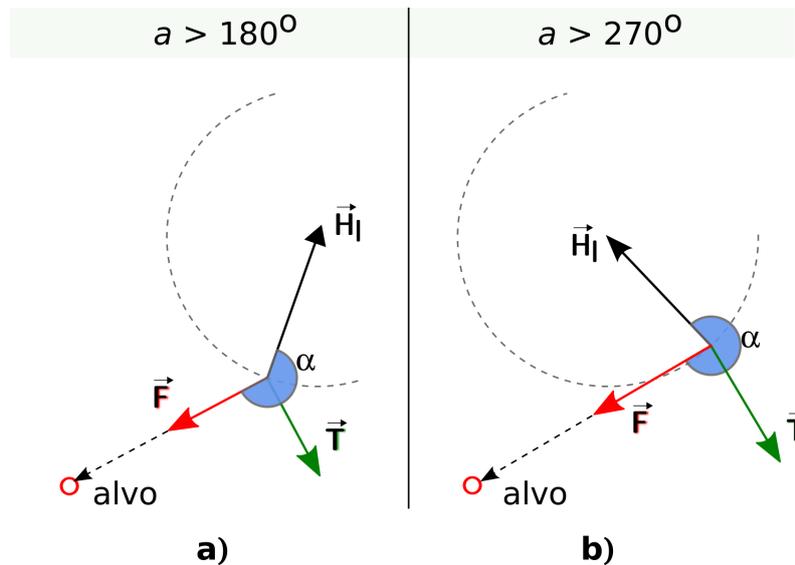


Figura 4.17 - Comportamento comando de aceleração para  $a > 180^\circ$

Então, uma proposta de solução para que o veículo não esteja na condição de instabilidade é limitar os comandos de correção para ângulos entre  $90^\circ$  e  $180^\circ$ , acionamento comandos de aceleração fora desse limite que, como vistos na figura 4.17, evita a condição de instabilidade do algoritmo de controle 1.

O que essa proposta representa do código é inserir a condição lógica para o acionamento do comando de aceleração. Essa alteração é feita no código mostrado no Anexo B, inicialmente como na figura 4.18 e alterado como visto na figura 4.19 e o resultado da alteração é visto na figura 4.20.

```

if SC_atual <= SC_ref % cone de espaco atual < cone desejado
    if a < pi/2
        Ang_pulso = b + pi; % comando de aceleracao
    else
        Ang_pulso = mod(b + pi/2,2*pi); % comando de correcao
    end;
    Frear = 0; % Falso
else
    Ang_pulso = c - pi/2; % reduzir a velocidade
    Frear = 1; % Verdadeiro
end

```

Figura 4.18 - Código para definição do tipo de controle

```

if SC_atual <= SC_ref % cone de espaco atual < cone desejado
    if a < pi/2 || a > pi
        Ang_pulso = b + pi; % comando de aceleracao
    else
        Ang_pulso = mod(b + pi/2,2*pi); % comando de correcao
    end;
    Frear = 0; % Falso
else
    Ang_pulso = c - pi/2; % reduzir a velocidade
    Frear = 1; % Verdadeiro
end

```

Figura 4.19 - Código para definição do tipo de controle com correção

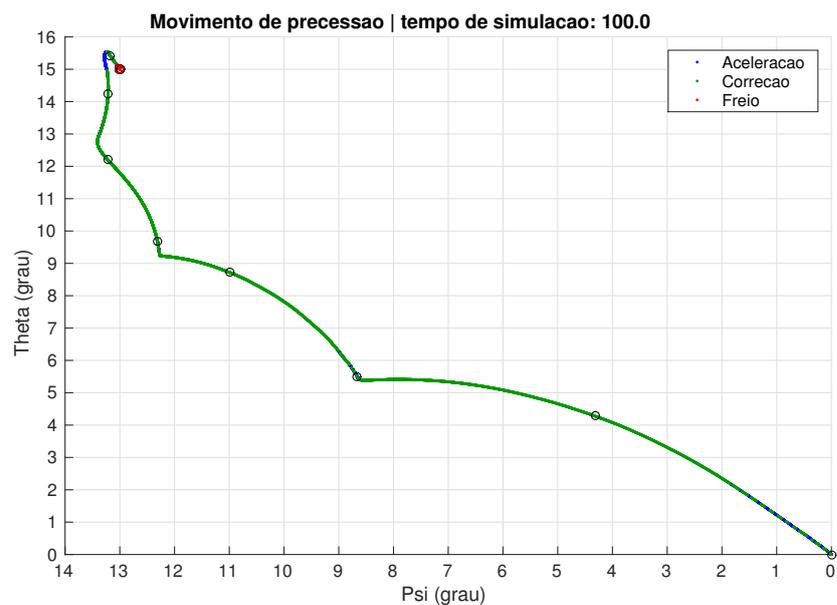


Figura 4.20 - Movimento de precessão para o alvo  $\theta$ :  $15^\circ$  e  $\psi$ :  $10^\circ$  com correção do comando

### 4.2.3 Controle 2

Também partindo a simulação na condição inicial

$$X_0 = \begin{bmatrix} \phi_0 \\ \theta_0 \\ \psi_0 \\ \omega_{x0} \\ \omega_{y0} \\ \omega_{z0} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 4\pi \\ 0 \\ 0 \end{bmatrix},$$

e definindo a posição do alvo como

$$\theta_{alvo} = 13^\circ \quad \text{e} \quad \psi_{alvo} = 15^\circ,$$

são testados os mesmos casos que feitos usando o algoritmo de controle 1.

O movimento de precessão que o corpo faz com o algoritmo de controle 2 é mostrado nas figuras 4.21 e 4.22. Percebe-se que nesse caso, *theta* e *psi* variam de forma a ir mais rapidamente para o alvo, não sofrendo tanto com a tendência de mudar de direção por conta da nutação.

O ângulo de cone do espaço  $\gamma$  dessa manobra, mostrado em função do tempo na figura 4.23, aumenta com o início da manobra e é mantido no seu valor máximo permitido de  $4^\circ$  até haver a aproximação do alvo.

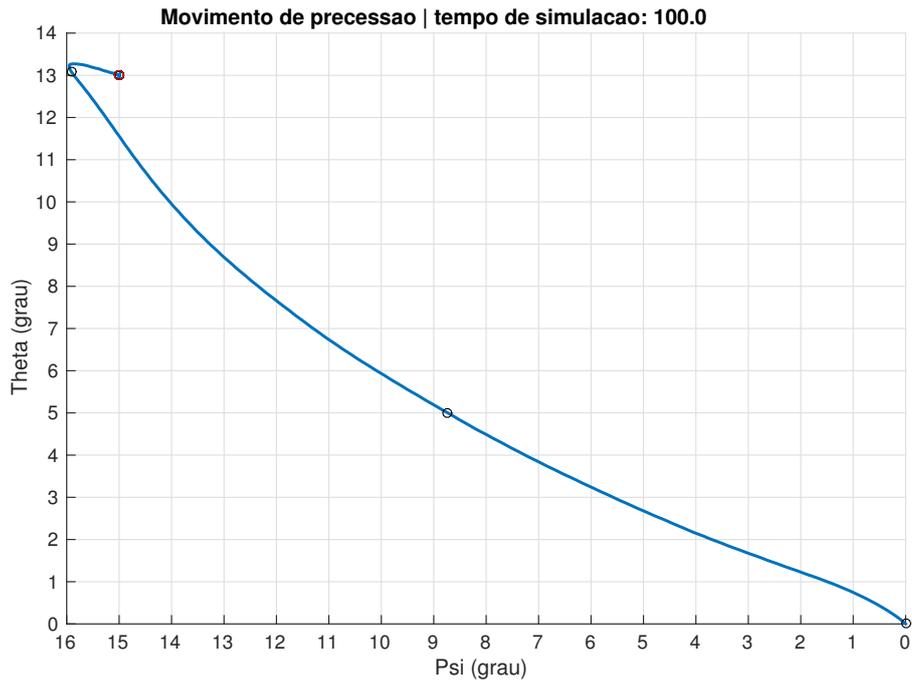


Figura 4.21 - Movimento de precessão para o alvo  $\theta: 13^\circ$  e  $\psi: 15^\circ$  usando o algoritmo de precessão 2

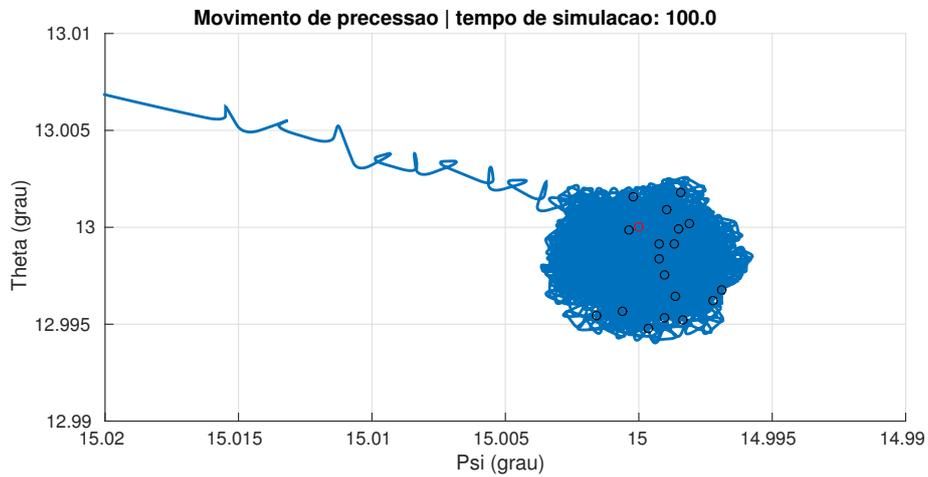


Figura 4.22 - Movimento de precessão para o alvo  $\theta: 13^\circ$  e  $\psi: 15^\circ$  na região do alvo usando o algoritmo de precessão 2

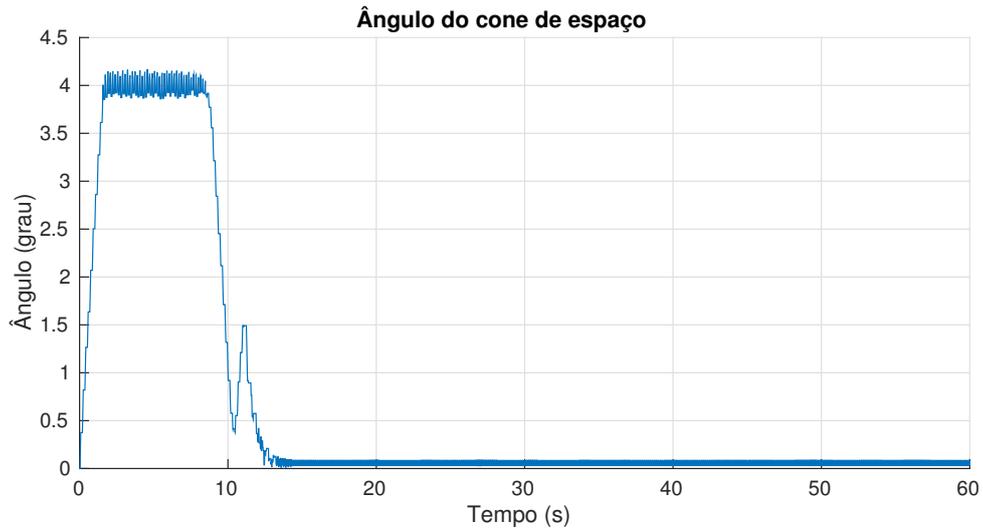


Figura 4.23 - Ângulo de cone  $\gamma$  para  $\theta_{alvo} = 13^\circ$  e  $\psi_{alvo} = 15^\circ$

Testando o algoritmo também com o caso

$$\theta_{alvo} = 15^\circ \quad \text{e} \quad \psi_{alvo} = 13^\circ,$$

obtém-se resultados muito semelhantes a esses, tanto no movimento de precessão, visto nas figuras 4.24 e 4.25, como no ângulo de cone de espaço, mostrado na figura 4.26.

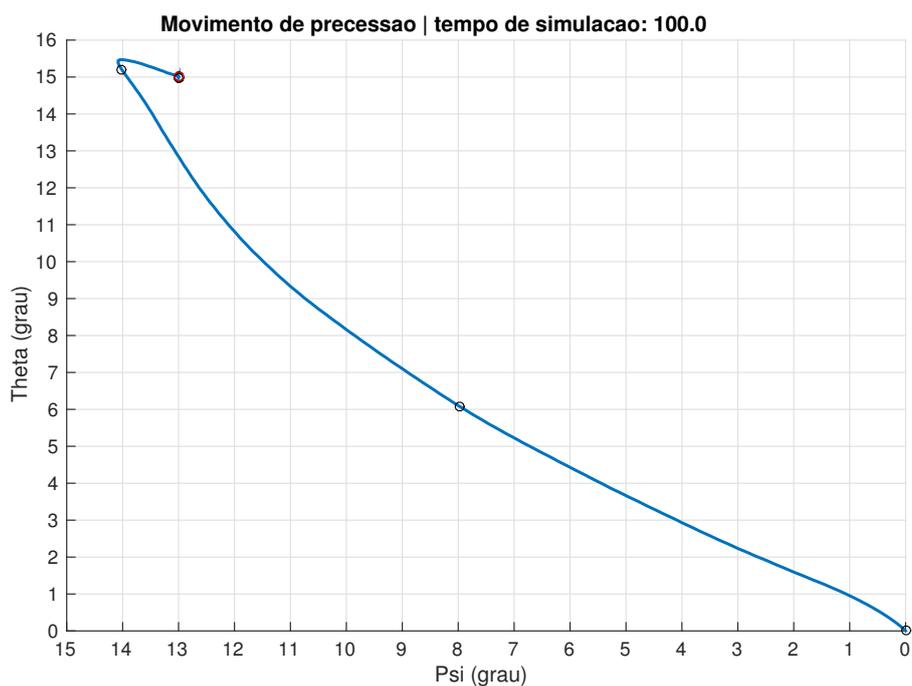


Figura 4.24 - Movimento de precessão para o alvo  $\theta: 15^\circ$  e  $\psi: 13^\circ$  usando o algoritmo de precessão 2

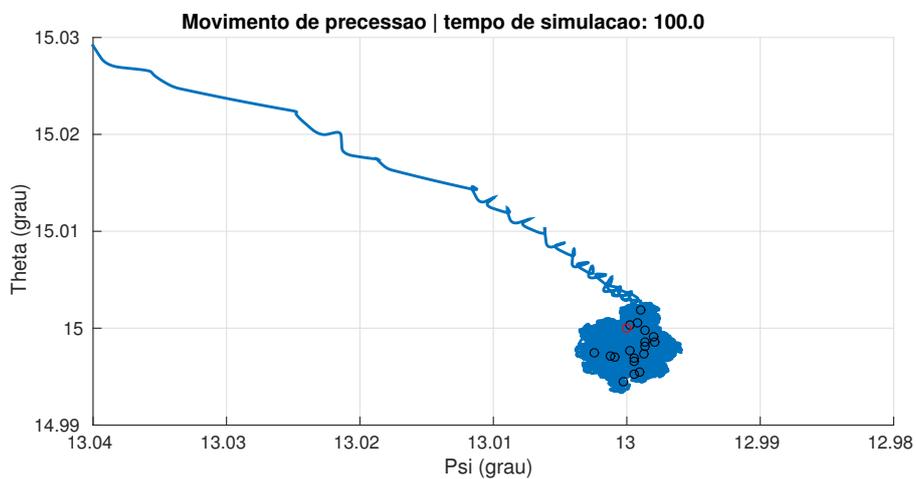


Figura 4.25 - Movimento de precessão para o alvo  $\theta: 15^\circ$  e  $\psi: 13^\circ$  na região do alvo usando o algoritmo de precessão 2

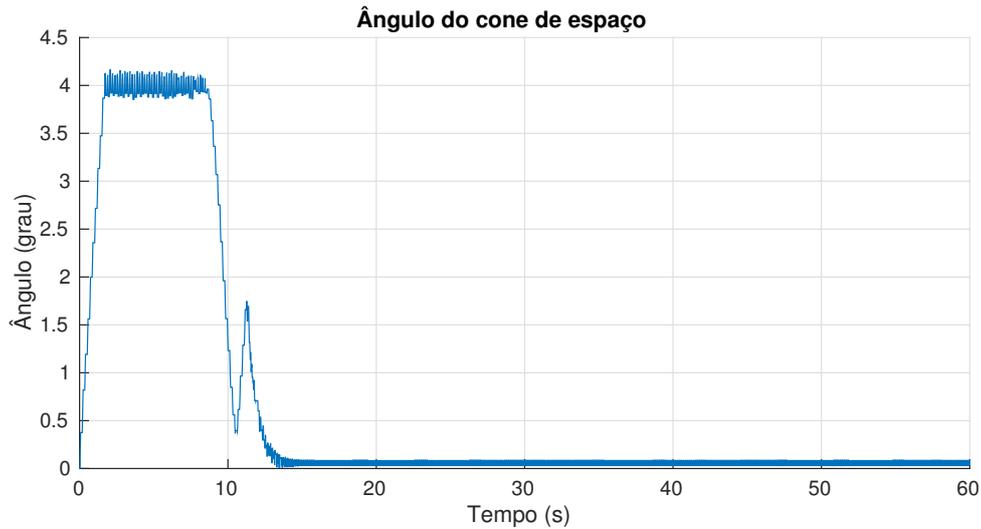


Figura 4.26 - Ângulo de cone  $\gamma$  para  $\theta_{alvo} = 15^\circ$  e  $\psi_{alvo} = 13^\circ$

#### 4.2.4 Comparação algoritmos de controle

Como métrica de comparação foram escolhidos dois parâmetros, o tempo de simulação até o veículo chegar ao alvo e o custo das manobras durante esse tempo.

Para medir o tempo de manobra, foi criada uma função que comanda a parada da simulação. O bloco dessa função pode ser visto na figura 4.4 com o nome de "condicaoParada", nesse bloco são lidos os módulos dos elementos  $a_{12e}$  e  $a_{13e}$ , que são proporcionais ao erro angular, e então somados, se esse valor obtido é menor que  $0.5e-3$  e mantido assim por 100 iterações seguidas é enviado um comando que encerra a simulação.

Para calcular o parâmetro de custo da manobra é executada uma função após o fim da simulação lê as curvas de comandos dos 3 jatos e então calcula a área de cada curva e soma a resposta. Essa resposta representa a área total dos comandos executados e isso tem relação direta com a quantidade de energia gasta na manobra.

Para se ter uma ideia mais ilustrativa do que a função faz, é mostrada na figura 4.27 um exemplo de uma curva de acionamentos. Nessa figura há um gráfico para cada jato e percebe-se o formato de onda quadrada característico de um comando digital.

Agora, para gerar as métricas de tempo de simulação e custo de manobra, são executados alguns casos aleatórios.

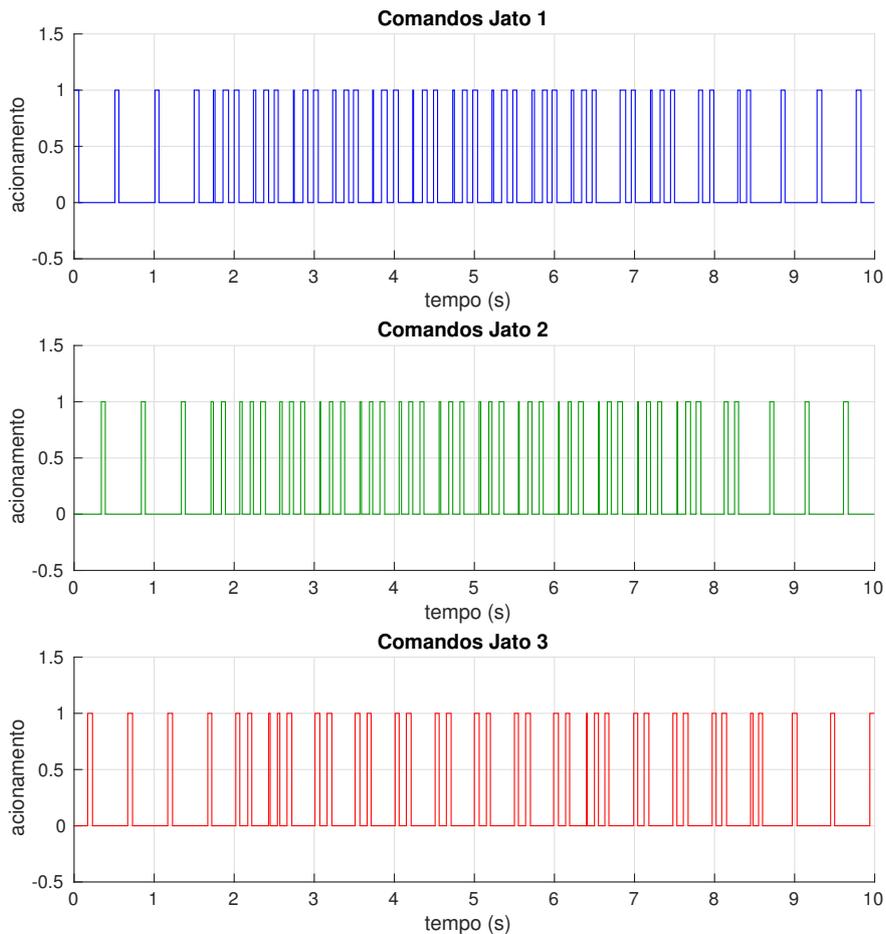


Figura 4.27 - Comandos de acionamento para o caso  $\theta = 15$  e  $\psi = 13$  usando o controle 2

A tabela 4.2 mostra o resultado das simulações para diferentes casos. De forma geral, o algoritmo 1 se mostra consistentemente melhor quanto a custo da manobra, porém o algoritmo 2 chega ao alvo mais rapidamente.

Apesar da falha no algoritmo de Controle 1 que acontece em apenas alguns casos, a comparação dos dois algoritmos de controle apresenta o Controle 1 como mais eficiente quanto a custo de manobra, gastando menos comandos para chegar ao alvo, porém o algoritmo 2 é mais rápido e apresenta uma trajetória de precessão mais regular.

Porém, a combinação Tempo x Custo, tabela 4.3, mostra que a multiplicação do tempo pelo custo dos algoritmos sugere que o algoritmo de controle 2 é mais eficiente que o algoritmo de controle 1.

Tabela 4.2 - Comparação entre algoritmo de Controle 1 e algoritmo de Controle 2

Nro	$\theta/\psi$	Controle 1		Controle 2	
		Tempo (s)	Custo	Tempo (s)	Custo
01	01/01	06.8	00.8	04.2	01.9
02	02/02	13.9	01.1	05.3	02.2
03	02/05	15.2	01.2	08.1	03.2
04	05/02	19.8	01.7	08.6	03.3
05	06/07	23.2	01.9	10.3	04.5
06	07/01	29.3	02.1	09.0	03.6
07	10/03	29.1	02.4	09.8	04.7
08	10/15	33.5	03.4	13.8	07.3
09	15/13	39.9	04.2	14.4	07.8
10	13/15	35.1	03.8	14.7	08.0
11	20/18	39.8	04.9	17.7	10.2
12	18/20	40.9	04.9	17.8	10.1

Tabela 4.3 - Comparação combinada Tempo X Custo

Nro	$\theta/\psi$	Controle 1	Controle 2
		Tempo (s) x Custo	Tempo (s) x Custo
01	01/01	005.5	008.2
02	02/02	016.4	011.6
03	02/05	019.0	026.4
04	05/02	035.0	028.5
05	06/07	044.7	047.2
06	07/01	063.8	033.0
07	10/03	071.5	046.5
08	10/15	116.5	101.1
09	15/13	168.3	113.4
10	13/15	133.3	117.8
11	20/18	198.2	181.0
12	18/20	202.4	180.6

## 5 CONCLUSÃO

Esse estudo foi originado de um projeto conduzido pelo DLR no qual foi necessário o uso de uma manobra de apontamento para cumprir a missão.

O primeiro desafio foi desenvolver um ambiente de simulação para poder avaliar a resposta dinâmica do veículo em uma manobra dessas. Nesse ponto foram estudadas e trabalhadas várias ferramentas até que, com o uso do MATLAB/SIMULINK e das bibliotecas de ferramentas matemáticas PROPAT, foi desenvolvido o ambiente apresentado na seção 4. Esse trabalho, por si só, já gerou grande ganho de conhecimento e possibilitou uma análise mais produtiva dos algoritmos de controle.

Com a ferramenta de simulação foi possível testar e comparar os dois algoritmos para a manobra de apontamento propostos na seção 3. O primeiro resultado interessante gerado foi o da identificação de um erro no algoritmo de apontamento usado pelo DLR no seu experimento, o estudo mostra que existe uma situação que pode levar o veículo a uma condição de instabilidade. Foi possível, analisando desse resultado, encontrar uma solução viável para corrigir o problema de instabilidade e garantir o funcionamento do algoritmo para os seus casos de operação.

Um outro resultado interessante foi visto na comparação dos algoritmos de apontamento, nela há uma implicação de que o algoritmo de controle 2 tem vantagens em relação ao algoritmo 1 já que ele tem uma relação de custo e tempo de manobra melhor, tornando-o uma opção aparentemente mais adequada para a missão.

Desse estudo apresentado podem ser apontados alguns trabalhos futuros. Um deles é melhorar o ambiente de simulação inserindo nos cálculos matemáticos as modelagens dos atuadores e sensores, tornando-o mais realista e então analisar o efeito dessa modelagem no desempenho dos algoritmos de apontamento. Outro trabalho interessante é fazer um estudo mais aprofundado no algoritmo 2, que por ter sido usado apenas como parâmetro de comparação com o primeiro algoritmo acabou não sendo foco desse trabalho e ficou pouco explorado.



## REFERÊNCIAS BIBLIOGRÁFICAS

BRYSON, A. E. **Applied optimal control: optimization, estimation and control**. Waltham: CRC Press, 1975.

CAVALCANTE, A. P. de M. **Simulação e controle da manobra de precessão de um veículo espacial**. 2013. Trabalho de Graduação. 37

CURTIS, H. **Orbital mechanics for engineering students**. Oxford: Elsevier, 2005.

ETTL, J. Precession manoeuvre. In: DEUTSCHES ZENTRUM FÜR LUFT- UND RAUMFAHRT, 2009, Bad Reichenhall, Germany. Oberpfaffenhofen: ESA, 2009. xv, 2, 11, 40

GREENSITE, A. L. **Analysis and design of space vehicle flight control systems**. New York: Spartan Books, 1970.

HUGHES, P. **Spacecraft attitude dynamics**. New York: Courier Dover Publications, 2012.

SIDI, M. J. **Spacecraft dynamics and control: a practical engineering approach**. Cambridge: Cambridge university press, 1997. 2, 21

STEVENS, B. L.; LEWIS, F. L. **Aircraft control and simulation**. New York: Wiley New York, 2003.

TEWARI, A. **Atmospheric and space flight dynamics: modeling and simulation with matlab® and simulink®**. Boston: Birkhäuser, 2007.

TURNER, J.; HÖRSCHGEN, M.; ETTL, J.; JUNG, W.; TURNER, P. Shefex-2 - development status of the vehicle and sub-systems for a hypersonic re-entry flight experiment. In: SYMPOSIUM ON EUROPEAN ROCKET AND BALLOON PROGRAMMES AND RELATED RESEARCH, 2009, Bad Reichenhall, Germany. Bad Reichenhall: ESA, 2009. 1

WIE, B. **Space vehicle dynamics and control**. Blacksburg: AIAA, 1998.



## ANEXO A - Código da função para montar dados de entrada da simulação

```
% funcao para montar estrutura com dados de entrada da simulacao
function modelo = montaModelo(ANG, PQR, inercia, jatosForca,...
    jatosBraco, jatosRaio, K_cone, coneMax, K_pulso, pulsoMax, pulsoMin)

% Elementos dos angulos de Euler rotacao 3-2-1 (z-y-x)
eulzyx = [ANG(3) ANG(2) ANG(1)]*pi/180;    % em radianos

% Attitude in quaternions
quat    = ezyxquat(eulzyx); % angulos de euler -> quaternios

% Matriz inertia
iner    = diag(inercia);
inerin  = inv(iner);

% Parametros do modelo
modelo.iner          = iner;           % Matriz tensora de inercia
modelo.inerin        = inerin;         % Inversa da matriz iner
modelo.forcaJ        = jatosForca;     % Forca dos jatos
modelo.bracoJ        = jatosBraco;     % Braco de torque (0.5m)
modelo.raio          = jatosRaio;     % Distancia Radial do atuador [m]
modelo.condicaoInicial = [quat; PQR];  % Condicao inicial

% Parametros controle do Ettl
param.SC_k          = K_cone;          % Ganho cone
param.SC_max        = coneMax*pi/180; % Angulo de cone espacial maximo
param.Pulso_k       = K_pulso;         % Ganho pulso
param.Pulso_min     = pulsoMin;        % Duracao minima do pulso (s)
param.Pulso_max     = pulsoMax;        % Duracao maxima do pulso (s)
param.Freq_rot      = PQR(1)/2/pi;    % Frequencia de rotacao do eixo X [Hz]

% Configuracoes da simulacao
modelo.param        = param;
end
```



## ANEXO B - Código do algoritmo de controle 1

```
%% Funcao de controle dos jatos
function U = controle(u,p_min,p_max,sfr,sc_const,p_const,sc_max)

%% Renomeando entradas
a      = u(1); % [rad]
b      = u(2); % [rad]
c      = u(3); % [rad]
d      = u(4); % [rad]
r1     = u(5); % [rad] angulo r1
r2     = u(6); % [rad] angulo r2
r3     = u(7); % [rad] angulo r3
sc_atual = u(8); % [rad] angulo do cone do espaco

%% Definindo angulo de pulso (rad)

% Referencia sc_ref para angulo do cone do espaco
sc_calc = sc_const*d;
if sc_calc > sc_max; sc_ref = sc_max; else
    sc_ref = sc_calc; end;

% Calculo do angulo de pulso
if sc_atual < sc_ref
    if a < pi/2; ang_pulso = pi + b; else
        ang_pulso = mod(pi/2 + b,2*pi); end;
    frear = 0;
else
    ang_pulso = c - pi/2;
    frear     = 1;
end

% correcao da referencia
ang_pulso = mod(ang_pulso+pi,2*pi);

%% Definindo do comprimento do pulso (s)

% Tempo de duracao do pulso (s)
comp_pulso      = p_const*d/1000;
comp_pulso_calc = comp_pulso;

if comp_pulso < 0.01; comp_pulso = p_min; end;

if (frear || comp_pulso > p_max); comp_pulso = p_max; end;
```

```

% Comprimento do arco formado pelo acionamento do pulso
ang_comp_pulso = comp_pulso*2*pi*sfr;

% Posicao angularo inicial e final do acionamento das valvulas
ang_inicial = mod((ang_pulso - ang_comp_pulso/2),2*pi);
ang_final   = mod((ang_pulso + ang_comp_pulso/2),2*pi);

%% Comandos de ABRIR/FECHAR valvulas
if ang_final > ang_inicial
    if (r1 > ang_inicial && r1 < ang_final); abre_1 = 1; else
                                                abre_1 = 0; end;

    if (r2 > ang_inicial && r2 < ang_final); abre_2 = 1; else
                                                abre_2 = 0; end;

    if (r3 > ang_inicial && r3 < ang_final); abre_3 = 1; else
                                                abre_3 = 0; end;
else
    if (r1 > ang_inicial || r1 < ang_final); abre_1 = 1; else
                                                abre_1 = 0; end;

    if (r2 > ang_inicial || r2 < ang_final); abre_2 = 1; else
                                                abre_2 = 0; end;

    if (r3 > ang_inicial || r3 < ang_final); abre_3 = 1; else
                                                abre_3 = 0; end;
end

%% Saidas da funcao

U = [abre_1
     abre_2
     abre_3
     sc_calc
     sc_ref
     ang_pulso
     frear
     comp_pulso_calc
     comp_pulso
     ang_comp_pulso
     ang_inicial
     ang_final];
end

```

## ANEXO C - Código do algoritmo de controle 2 - precessão

```
function [Tc, acionaJatos, vetorAbertura, alvo, errB] = fcn(ANG, PQR, B, angCom)
    %% controle de precessao
    % ganhos = [kyp kyd kzp kzd];
    % precisa melhorar a logica, se ha um acionamento na mesma posicao a
    % forca nao eh dobrada como aconteceria nessa situacao

    % POSICAO ANGULAR
    ax      = ANG(1);

    % VELOCIDADE ANGULAR
    % wx     = vAng(1);
    wy      = PQR(2);
    wz      = PQR(3);

    % CALCULANDO MATRIZ DE ERRO 'errB'
    tgB     = ezyxrmx([angCom(2) angCom(1) ax]);
    errB    = B*tgB';

    % lendo elementos de erro na matriz
    % ex     = errB(2,3);
    ey      = errB(3,1);
    ez      = errB(1,2);

    % i_ = rotx(ax)*[0 ey ez]';

    % GANHOS DO CONTROLE
    Ky      = 3.14; Kyd = 2.23;
    Kz      = 3.14; Kzd = 2.23;

    % LEI DE CONTROLE PD
    Tcy     = Ky*ey + Kyd*wy;
    Tcz     = Kz*ez + Kzd*wz;
    torque  = [0 Tcy Tcz]';

    % VETOR ALVO E POSICAO DOS BOCAIS
    vetorAlvo      = rotx(ax)*torque;
    vetorAcionamento = rotx(-pi/2)*vetorAlvo;

    ang1  = ax;
    ang2  = ang1+2*pi/3;
    ang3  = ang1+4*pi/3;
```

```

arco      = 40;
d2r      = pi/180;

arco      = arco*d2r; % convertendo para radianos

% vetor na posicao de abertura do bocal
vetorAbertura = rotx(0)*vetorAcionamento;

% vetores nas posicoes dos bocais [jato1 alinhado com eixo Y]
jato      = [0 0 1]';
jato1     = rotx(ang1)*jato;
jato2     = rotx(ang2)*jato;
jato3     = rotx(ang3)*jato;

% encontrando angulo entre jato e vetorAbertura
angAcionamento1 = anguloVetores(jato1,vetorAbertura);
angAcionamento2 = anguloVetores(jato2,vetorAbertura);
angAcionamento3 = anguloVetores(jato3,vetorAbertura);

% acionando jatos
acionaJatos = [false false false]';
if angAcionamento1 >= 0 && angAcionamento1 < abs(arco)
    acionaJatos(1) = true;
end
if angAcionamento2 >= 0 && angAcionamento2 < abs(arco)
    acionaJatos(2) = true;
end
if angAcionamento3 >= 0 && angAcionamento3 < abs(arco)
    acionaJatos(3) = true;
end

alvo = torque;
if all(acionaJatos==0)
    torque = torque*0;
else
    torque = torque/norm(torque)*5;
end

Tc = torque;

```

## ANEXO D - Código do algoritmo de controle 2 - nutação

```
%% funcao para controle de nutacao do veiculo
function acionamento = fcn(inercia,PQR)
    % calculando angulo entre vetor velocidade angular plano YZ e o vetor jato Ja
    velYZ = [0 PQR(2) PQR(3)]';

    % Vetores direcao das valvulas
    v1 = [0 1 0]';
    v2 = rotx(120*pi/180)*v1;
    v3 = rotx(120*pi/180)*v2;

    % projecao do vetor velocidade angular por valvula
    w1 = 0; w2 = 0; w3 = 0;
    if abs(norm(velYZ)) > 0
        w1 = dot(v1,velYZ)/norm(v1);
        w2 = dot(v2,velYZ)/norm(v2);
        w3 = dot(v3,velYZ)/norm(v3);
    end

    % acionamento por valvula
    velNut = -0.1/2/pi;

    % angulo de cone
    W_l = norm([PQR(2) PQR(3)]);
    W_r = PQR(1);
    Iz = inercia(2,2);
    Ix = inercia(1,1);

    Beta = atan2(Iz*W_l,Ix*W_r); % Angulo entre H_x e H
    Alpha = atan2(W_l,W_r); % Angulo do cone do corpo
    Gamma = (Beta-Alpha); % Angulo do cone do espaco

    p1 = false; p2 = false; p3 = false;
    if Gamma > 4*pi/180
        if w1 < velNut
            p1 = true;
        elseif w2 < velNut
            p2 = true;
        elseif w3 < velNut
            p3 = true;
        end
    end
    acionamento = [p1 p2 p3]';
```