DISSERTAÇÃO DE MESTRADO

CONTROLE PREDITIVO BASEADO EM MODELO APLICADO EM SATÉLITES RÍGIDO-FLEXÍVEIS

PEDRO JORGE DE DEUS PEIXOTO

Brasília, Março de 2018

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA MECÂNICA

CONTROLE PREDITIVO BASEADO EM MODELO APLICADO EM SATÉLITES RÍGIDO-FLEXÍVEIS

PEDRO JORGE DE DEUS PEIXOTO

DISSERTAÇÃO SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA MECÂNICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM SISTEMAS MECATRÔNICOS

APROVADA POR:

Prof. Dr. André Murilo de A. Pinto, PPMEC/UnB
Orientador

Prof. Dr. Luiz Carlos Gadelha de Souza, FGA/UnB
Co-orientador

Prof. Dr. Eugênio Feitosa Fortaleza, PPMEC/UnB
Membro Interno

Prof. Dr. Renato Vilela Lopes, FGA/UnB
Membro Externo

BRASÍLIA/DF, 26 MARÇO DE 2018

Peixoto, Pedro Jorge de Deus
Controle preditivo baseado em modelo aplicado em satélites rígido-flexíveis / PEDRO JORGE DE DEUS PEIXOTO. –Brasil, 2018.
75 p.

Orientador: André Murilo de Almeida Pinto

Dissertação (Mestrado) – Universidade de Brasília – UnB
Faculdade de Tecnologia – FT
Programa de Pós-Graduação em Sistemas Mecatrônicos – PPMEC, 136/2018.

1. Satélite rígido-flexível. 2. Controle Preditivo. 3. Controle de atitude. 4. Regulador linear quadrático (LQR). 5. Hardware in the loop (HIL). I. André Murilo de Almeida Pinto, orientador. II. Universidade de Brasília. III. Faculdade de Tecnologia.

Dedicatória

Ao meu avô, Dr. José Ulysses Peixoto Neto.

PEDRO JORGE DE DEUS PEIXOTO

Agradecimentos

Em primeiro lugar, agradeço a Deus pelo dom da vida e pela capacidade de chegar até este ponto, enfrentando e superando os inúmeros obstáculos que apareceram no caminho.

Aos meus pais, Jorge Peixoto e Germana Abath, dos quais sempre recebi apoio sobre qualquer decisão na minha vida acadêmica e profissional. Sem eles, eu nunca teria chegado até aqui. Agradeço também aos meus irmãos e demais familiares, que sempre apostaram muito no meu potencial, em especial a minha irmã Allana, exemplo de pesquisadora e que me incentivou muito a chegar nesse estágio.

Agradeço ao Prof. Dr. André Murilo, meu orientador, que, desde o início deste trabalho, demonstrou motivação, interesse, compromisso e disponibilidade, atributos que foram essenciais para a conclusão do mesmo. Agradeço também por todo o conhecimento que pude adquirir, não só na área de controle de sistemas, mas também nas múltiplas áreas que o professor domina.

Ao Prof. Dr. Luiz Carlos Gadelha, meu coorientador, meus sinceros agradecimentos por todo o auxílio prestado, principalmente na área de satélites. Os conhecimentos que adquiri nesta área foram essenciais para o entendimento do modelo que utilizei neste trabalho.

Agradeço também aos demais mestres da Universidade de Brasília, que, em maior ou menor grau, contribuíram para a conclusão desta dissertação. Em especial, ao Prof. Renato Coral, que contribuiu substancialmente para o desenvolvimento do meu projeto. Agradeço ainda à Profa. Dra. Suzana Ávila, minha prima e minha porta de entrada na UnB, a qual é minha inspiração como professora e engenheira.

Aos colegas e amigos que fiz em Brasília nesse período de UnB: Reurison Silva, Rafael Rodrigues, Vinícius Oliveira, Maíra Barros, Alceu Castanheira, entre tantos outros que, através do compartilhamento de experiências, tornaram a trajetória do mestrado muito agradável.

Agradeço aos Professores que participaram da minha banca, Prof. Renato Lopes e Prof. Eugênio Fortaleza, pelas importantes contribuições neste trabalho.

Agradeço também à CAPES e ao CNPq pelo fomento, incentivo e auxílio financeiro na realização deste trabalho.

Por fim, agradeço a todos que, de uma forma ou de outra, contribuíram para a conclusão do meu trabalho de mestrado.

PEDRO JORGE DE DEUS PEIXOTO

RESUMO

Satélites artificiais têm se tornado muito importantes nas últimas décadas devido à sua vasta aplicabilidade nas diversas áreas do conhecimento, tais como, por exemplo, meteorologia, telecomunicações e astronomia. Um processo importante ao se colocar um satélite em órbita, para que ele cumpra a missão à qual é destinado, é o seu controle de atitude, que é o controle da posição angular do satélite em relação a um referencial fixo. Diversas técnicas de controle de atitude têm sido pesquisadas para as diferentes aplicações e particularidades de cada tipo de satélite. Uma característica que torna o controle de atitude de um satélite ainda mais desafiador é a presença de apêndices flexíveis no corpo do satélite. Na literatura, satélites que possuem esse tipo de apêndice são geralmente denominados rígido-flexíveis. Os apêndices flexíveis são bastante comuns em satélites, podendo consistir de painéis solares, velas, ou mesmo antenas, e podem dificultar, ou até mesmo impossibilitar, as manobras do satélite, pois, durante o movimento do mesmo, oscilações indesejadas podem ser excitadas nas estruturas flexíveis. Este trabalho tem o objetivo de projetar e validar controladores para um modelo de satélite rígido-flexível. O modelo do satélite é desenvolvido utilizando a abordagem de Lagrange, resultando em um sistema não-linear, que posteriormente é linearizado para o projeto de duas técnicas de controle baseadas em modelo linear: o *Linear Quadratic Regulator* (LQR) e o *Model-based Predictive Control* (MPC). Os controladores foram validados utilizando uma plataforma *Hardware in the loop* (HIL), uma arquitetura onde o controlador é implementado em *Hardware*, e o modelo do satélite, em ambiente de *Software*. O ambiente *Simulink Real Time* foi utilizado para implementar o modelo, e os controladores foram implementados em placa micro-processada *Beaglebone*. A plataforma HIL foi validada, pois conseguiu-se controlar a planta com os dois controladores propostos, para diversos cenários. Os resultados obtidos mostraram que o controlador MPC obtém melhor desempenho que o LQR em relação às oscilações da haste flexível, já que essa técnica lida diretamente com restrições nas variáveis do problema, apresentando também menor sobressinal para um horizonte de predição 60, com tempo de assentamento relativamente baixo. Os resultados de simulação também demonstraram que os modelos linear e não-linear do satélite possuem comportamentos bastante semelhantes, indicando que os termos não-lineares deste modelo têm pouca influência na dinâmica do sistema.

Palavras-chave: Satélite Rígido-Flexível. Controle de Atitude. Controle Preditivo. Regulador Linear Quadrático (LQR). *Hardware in the Loop* (HIL).

ABSTRACT

Artificial satellites have become very important in the last decades because of their wide applicability in the many areas of knowledge, such as meteorology, telecommunications and astronomy. An important process when placing a satellite in orbit, in order to fulfill the mission for which it is intended, is its attitude control, which is the control of the angular position of the satellite in relation to a fixed reference frame. Several attitude control techniques have been researched for the different applications and particularities of each type of satellite. One feature that makes the attitude control of a satellite even more challenging is the presence of flexible appendages in the satellite body. In the literature, satellites that have this type of appendix are usually called rigid-flexible. Flexible appendages are quite common in satellites, and may consist of solar panels, sails, or even antennas, and may hinder or even render impossible the maneuvers of the satellite, because, during the movement of the satellite, unwanted oscillations can be excited on the flexible structures. This work has the objective of designing and validating controllers for a rigid-flexible satellite model. The satellite model is developed using the Lagrange approach, resulting in a nonlinear system, which is later linearized for the design of two techniques of linear model-based control: Linear Quadratic Regulator (LQR) and Model-based Predictive Control (MPC). The controllers were validated using a *hardware in the loop* (HIL) platform, an architecture in which the controller is implemented in Hardware, and the satellite model is implemented in software environment. The environment *Simulink Real Time* was used to implement the model, and the controllers were implemented on micro-processed Beaglebone board. The HIL platform was validated inasmuch as it was possible to control the plant with the two controllers proposed for different scenarios. The results showed that the MPC controller performs better than the LQR controller with regard to the oscillations of the flexible appendix, since this technique deals directly with constraints on the problem variables, presenting also a smaller overshoot for a prediction horizon of 60, with a quite low settling time. The simulation results showed also that the linear and the nonlinear models of the satellite have very similar behavior, which indicates that the nonlinear terms of such model have little influence on the dynamics of the system.

Keywords: Rigid-Flexible Satellite. Attitude Control. Predictive Control. Linear Quadratic Regulator (LQR). Hardware in the Loop (HIL).

SUMÁRIO

RESUMO ..... **i**

ABSTRACT ..... **ii**

LISTA DE FIGURAS ..... **iv**

LISTA DE TABELAS ..... **vi**

LISTA DE ABREVIATURAS E ACROGRAMAS ..... **viii**

1 Introdução **1**
1.1 Contextualização 1
1.2 Definição do Problema 3
1.3 Objetivos da Dissertação 3
1.3.1 Objetivo Geral 3
1.3.2 Objetivos Específicos 3
1.4 Resumo da Metodologia 3
1.5 Contribuições do Trabalho 5
1.6 Apresentação do Documento 5

2 Revisão Bibliográfica **7**
2.1 Introdução 7
2.2 Revisão da Literatura 7

3 Fundamentação Teórica **14**
3.1 Introdução 14
3.2 Satélites 14
3.2.1 Desenvolvimento do modelo de satélite rígido-flexível 16
3.3 Controle Preditivo Baseado em Modelo (MPC) 22
3.4 Regulador Quadrático Linear (LQR) 31
3.5 Arquitetura de validação de controladores - Plataforma HIL 33
3.5.1 *Model in the loop* (MIL) 34
3.5.2 *Software in the loop* (SIL) 35
3.5.3 *Process in the loop* (PIL) 35
3.5.4 *Hardware in the loop* (HIL) 35

4 Metodologia 38
4.1 Introdução 38
4.2 Parametrização 38
4.2.1 Parametrização Exponencial 40
4.3 Plataforma HIL utilizada 42
4.4 Solvers 44
4.4.1 Quadprog 44
4.4.2 Expansão do Gradiente 45
4.4.3 KKT 46
4.4.4 qpOASES 47

5 Resultados 49
5.1 Introdução 49
5.2 Parâmetros Utilizados 50
5.3 Resultados com LQR 51
5.3.1 Resultados de Simulação para o LQR em MIL 52
5.3.2 Resultados de Simulação para o LQR em HIL 55
5.4 Resultados com MPC 59
5.4.1 Resultados de Simulação para o MPC em MIL 60
5.4.2 Resultados de Simulação para o MPC em HIL 63
5.5 Análise dos Resultados Obtidos 67

6 Conclusões 71
6.1 Conclusões do Trabalho 71
6.2 Sugestões para Trabalhos Futuros 72

REFERÊNCIAS BIBLIOGRÁFICAS 73

LISTA DE FIGURAS

1.1 Comunicação via satélite 1
1.2 Satélites com estruturas flexíveis 2

2.1 Módulo flexível utilizado por (STINGA et al., 2008) 9
2.2 Simulador utilizado por (SOUZA; ARENA, 2012) 10
2.3 Plataforma HIL utilizada por (VICENTE; RIBEIRO, 2014) 11

3.1 Referenciais em um satélite 15
3.2 Modelo de Satélite rígido-flexível 16
3.3 Comportamento do sistema linear (em vermelho) e do sistema não-linear (em azul) em malha aberta para uma excitação degrau 22
3.4 Etapas para validação de controladores 34
3.5 Esquema mostrando uma malha de controle em *Model in the Loop* (MIL) 34
3.6 Esquema de malha de controle em *Hardware in the Loop* (HIL) 36

4.1 Plataforma HIL utilizada para as simulações deste trabalho 42

5.1 Cenários de simulação 49
5.2 Comportamento do satélite utilizando o controlador LQR para o modelo linear em MIL 52
5.3 Comportamento do satélite utilizando o controlador LQR saturado para o modelo linear em MIL 53
5.4 Comportamento do satélite utilizando o controlador LQR para o modelo não-linear em MIL 54
5.5 Comportamento do satélite utilizando o controlador LQR saturado para o modelo não-linear em MIL 55
5.6 Comportamento do satélite utilizando o controlador LQR para o modelo linear em HIL 56
5.7 Comportamento do satélite utilizando o controlador LQR saturado para o modelo linear em HIL 57
5.8 Comportamento do satélite utilizando o controlador LQR para o modelo não-linear em HIL 58
5.9 Comportamento do satélite utilizando o controlador LQR saturado para o modelo não-linear em HIL 59
5.10 Comportamento do satélite utilizando o controlador MPC para o modelo linear em MIL, com um horizonte de predição de 20 60
5.11 Comportamento do satélite utilizando o controlador MPC para o modelo linear em MIL, com um horizonte de predição de 60 61

5.12 Comportamento do satélite utilizando o controlador MPC para o modelo não-linear em MIL, com um horizonte de predição de 20 62
5.13 Comportamento do satélite utilizando o controlador MPC para o modelo não-linear em MIL, com um horizonte de predição de 60 63
5.14 Comportamento do satélite utilizando o controlador MPC para o modelo linear em HIL, com um horizonte de predição de 20 64
5.15 Comportamento do satélite utilizando o controlador MPC para o modelo linear em HIL, com um horizonte de predição de 60 65
5.16 Comportamento do satélite utilizando o controlador MPC para o modelo não-linear em HIL, com um horizonte de predição de 20 66
5.17 Comportamento do satélite utilizando o controlador MPC para o modelo não-linear em HIL, com um horizonte de predição de 60 67

LISTA DE TABELAS

3.1 Frequências naturais de vibração da estrutura flexível para os cinco primeiros modos 20

5.1 Parâmetros do modelo de satélite rígido-flexível 50
5.2 Parâmetros de sintonia do LQR 50
5.3 Parâmetros de sintonia do MPC 51
5.4 Tempos de cálculo da variável de comando pelo controlador 68
5.5 Indicadores de desempenho para as simulações realizadas em MIL para o controlador LQR. 68
5.6 Indicadores de desempenho para as simulações realizadas em HIL para o controlador LQR . 68
5.7 Indicadores de desempenho para as simulações realizadas em MIL para o controlador MPC 69
5.8 Indicadores de desempenho para as simulações realizadas em HIL para o controlador MPC. 69

LISTA DE ABREVIATURAS E ACROGRAMAS

Símbolos Latinos

A	Matriz de transmissão de estados do sistema	
B	Matriz de entrada do sistema	
b_m	Componente de atrito viscoso	[m^2/s]
E	Módulo de Young	[N/m^2]
E_c	Energia cinética	[J]
E_p	Energia potencial	[J]
I	Momento de inércia seccional da haste	[kg.m^2]
J	Função custo	
K_e	Coeficiente de amortecimento	
L	Comprimento da haste	[m]
m_L	Massa concentrada na ponta da haste	[kg]
M_p	Sobressinal	
R	Raio do rotor	[m]
T_a	Tempo de acomodação 2%	[s]
u	Variável de comando	
x	Vetor de estados	
w	Deslocamento flexível da haste	[m]

Símbolos Gregos

α	Deflexão total da haste	[rad]
θ	Deflexão rígida do satélite	[rad]
μ	Densidade linear da haste	[kg/m]
τ	Torque	[N.m]
η	Coordenada generalizada	
χ	Forma matricial das coordenadas generalizadas	
φ	Função de forma	
δ	Variação da variável de comando	
Π	Matriz de seleção	

Siglas

DPM	Distributed Parameters Model
HIL	Hardware in the Loop
LIT	Linear e Invariante no Tempo
LMI	Linear Matrix Inequality
LQG	Linear Quadratic Gaussian Regulator
LQR	Linear Quadratic Regulator
MIL	Model in the Loop
MPC	Model based Predictive Control
MSM	Mass-Spring Model
NMPC	Nonlinear Model based Predictive Control
PD	Proportional Derivative Control
PID	Proportional Integral Derivative Control
PIL	Process in the Loop
PWM	Pulse-Width Modulation
RPY	Roll Pitch Yaw
SDRE	State Dependent Riccati Equation
SIL	Software in the Loop
UDP	User Datagram Protocol

Capítulo 1

Introdução

1.1 Contextualização

O uso de satélites tem crescido muito nas últimas décadas, devido a seu importante papel nas telecomunicações, na astronomia, na meteorologia, entre outras áreas. Com o auxílio de um satélite, a comunicação entre duas pessoas em lados opostos do planeta é feita em poucos segundos, um avanço tecnológico inimaginável algumas décadas atrás. A figura abaixo mostra um exemplo de comunicação via satélite.

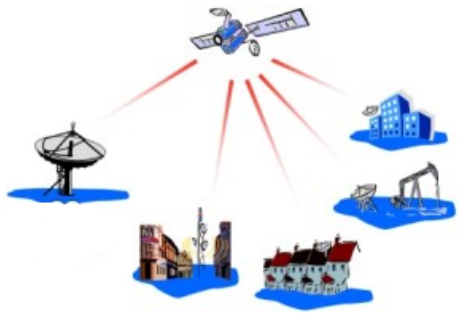


Figura 1.1: Comunicação via satélite

Algumas tecnologias hoje em dia somente são viáveis devido aos satélites artificiais em órbita ao redor da Terra. Uma tecnologia bastante comum que utiliza esses dispositivos é o *Global Positioning System*, conhecido como GPS, que utiliza um conjunto de satélites que conseguem cobrir a totalidade da superfície do planeta, possibilitando a localização de pontos na superfície terrestre com uma precisão de poucos metros (RODRIGUES et al., 2014).

Ao colocar um satélite em órbita, é importante que este adquira uma posição adequada para conseguir receber e transmitir sinais na direção correta. Nesse contexto, uma etapa crucial é o controle de atitude do satélite, ou seja, o controle do posicionamento angular do mesmo em relação a eixos de referência fixos.

O controle de atitude de um satélite fica especialmente complexo quando ele possui, acopladas em seu

corpo, estruturas flexíveis, tais como antenas, painéis solares, braços robóticos ou velas, pois tais estruturas podem oscilar durante o movimento do satélite, dificultando sua missão. Satélites com estruturas flexíveis são também chamados de satélites rígido-flexíveis, por possuírem um corpo rígido e apêndices flexíveis acoplados nele. Na figura abaixo são mostrados alguns exemplos de estruturas flexíveis em satélites.

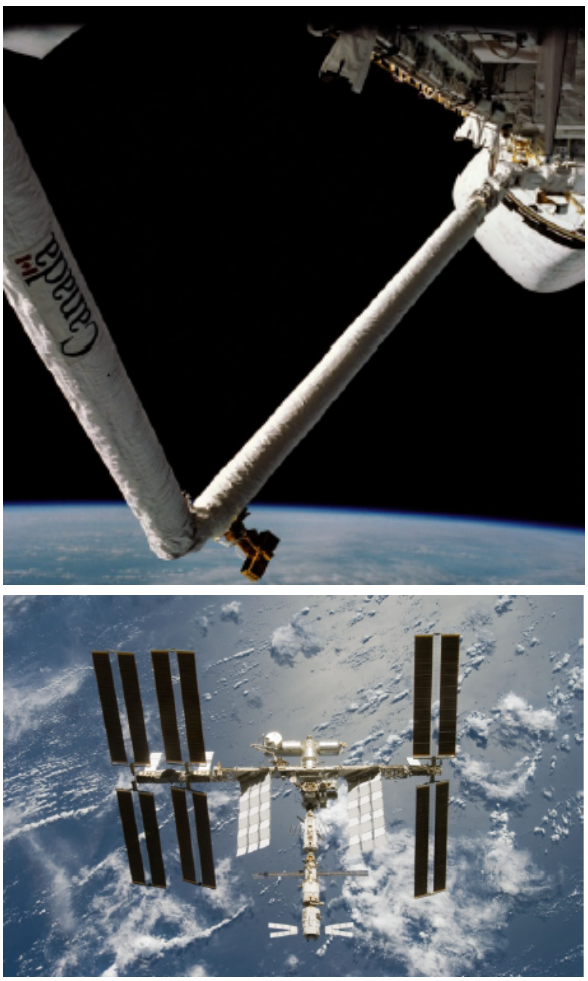


Figura 1.2: Satélites com estruturas flexíveis
Fonte: (GRAVADE, 2009)

As mais diversas técnicas de controle podem ser aplicadas para controlar a atitude do satélite. Alguns métodos que podem ser utilizados são o controle Proporcional Integral Derivativo (PID) (KUKRETI et al., 2015), o Regulador Linear Quadrático (LQR) (SOUZA; GALVÃO, 2015), o controle preditivo baseado em modelo (MPC) (EREN et al., 2017), o filtro H_∞ (PINHEIRO; SOUZA, 2013), o SDRE (*State Dependent Riccati Equation*) (SOUZA; ARENA, 2012), entre outros.

Qualquer que seja o método de controle empregado, faz-se necessário um procedimento de testes do controlador projetado, para que, ao ser embarcado no satélite, o controlador tenha um grau de confiabilidade alto. Nesse sentido, um dos métodos que vem sendo bastante utilizado para testar controladores é o *Hardware in the loop* (HIL), o qual utiliza um modelo para a planta simulado em *software*, em uma malha de controle com o controlador implementado em *hardware* (VICENTE; RIBEIRO, 2014). Essa arquitetura

aproxima bem o cenário real, podendo identificar previamente vários problemas que surgiriam ao embarcar o controle no satélite.

1.2 Definição do Problema

A vibração de estruturas flexíveis durante a movimentação de um satélite rígido-flexível pode prejudicar ou inviabilizar a missão do mesmo. Por isso, é importante que tais vibrações sejam controladas de forma eficaz, de modo a deixar as amplitudes de oscilação da estrutura flexível dentro de uma faixa que não prejudique a movimentação do satélite. Nesse sentido, existem diversos métodos de controle automático que podem ser utilizados para resolver este problema.

1.3 Objetivos da Dissertação

1.3.1 Objetivo Geral

O objetivo geral deste trabalho é projetar um controlador para controlar as vibrações de uma haste flexível em um modelo de satélite rígido-flexível, utilizando uma plataforma de HIL para validar tal controlador.

1.3.2 Objetivos Específicos

Para atingir o objetivo geral no presente trabalho, foram estabelecidos os objetivos específicos a seguir:

- Obter um modelo matemático que represente com fidedignidade a dinâmica de um satélite rígido-flexível;
- Projetar um controlador adequado para controlar as vibrações da estrutura flexível;
- Validar o controlador projetado em ambientes de simulação de *software* e *hardware*;
- Analisar o desempenho das simulações realizadas.

1.4 Resumo da Metodologia

Para atingir o objetivo deste trabalho, de caráter principalmente experimental, foi seguida uma metodologia, a qual será brevemente descrita neste capítulo.

A princípio, foi desenvolvido um modelo que represente um satélite rígido-flexível. O modelo foi desenvolvido a partir da equação de Euler-Bernoulli, que, juntamente com o método dos modos assumidos, representa a dinâmica da haste flexível do satélite. O desenvolvimento segue utilizando o formalismo lagrangiano da mecânica analítica até chegar nas equações finais que definem o movimento de um satélite rígido-flexível com um grau de liberdade de rotação.

O modelo encontrado, composto de equações diferenciais não-lineares, foi linearizado em torno de um ponto de operação, através do método de Taylor, para que fosse possível a aplicação de técnicas de controle baseadas em modelos lineares na forma de espaço de estados, tais como o LQR e o MPC, técnicas fundamentais no desenvolvimento deste trabalho.

Uma vez bem definido o modelo do satélite, foram projetados dois tipos de controladores para controlar o modelo linear do satélite.

O primeiro deles, o LQR, é uma técnica de controle por realimentação de estados, em que os ganhos de realimentação são calculados através da resolução da equação de Riccati. Esta técnica é considerada uma técnica de controle ótimo, porém não lida com restrições nas variáveis do sistema.

O segundo controlador utilizado é o MPC, uma técnica de controle preditivo baseado em um modelo linear, que consegue achar o melhor "caminho"para uma variável regulada, seguindo restrições inerentes ao sistema.

Os mesmos controladores também foram utilizados para controlar o modelo não-linear original, verificando assim a robustez desses controladores.

Os testes em malha fechada foram realizados primeiramente em software, numa arquitetura chamada *Model in the Loop* (MIL). Nesta arquitetura, tanto o modelo quanto o controlador são simulados numa malha de controle em ambiente de *software*. O *software* utilizado para esta etapa foi o *Matlab/Simulink*, utilizando o método de Runge-Kutta de quarta ordem. As simulações foram realizadas em um computador com processador *Intel Core* i5 de 1,70 GHz, 4 GB de memória RAM e sistema operacional *Windows* 8.

Em seguida, foi utilizada uma plataforma *Hardware in the loop* (HIL) para validar os controladores projetados. Nesta arquitetura, o modelo do satélite continua em ambiente de *software*, e o controlador é implementado em *hardware*.

O modelo do satélite foi implementado em uma máquina dedicada somente a ele, para simular com a maior realidade possível a velocidade com que um satélite real responderia aos estímulos do controlador. O ambiente utilizado para implementar o modelo foi o *Simulink Real-Time*, uma plataforma apropriada para aplicações de tempo real.

Os controladores foram implementados em uma placa micro-processada de baixo custo, a *Beaglebone Black*. Essa placa possui uma série de facilidades para a aplicação pretendida neste trabalho. Ela possui interface USB e *Ethernet*, além de entradas e saídas analógicas e digitais. Seu processador é o AM3358 Cortex-A8, de 1,0 GHz, memória de 512 MB DDR3, e possui suporte a diversos sistemas operacionais, tais como Debian, Android e Ubuntu (COLEY, 2013). O sistema operacional utilizado foi o Debian.

O envio de estados do modelo para o controlador foi feito através de conexões analógicas, utilizando uma placa de aquisição de dados PCI-DAC6703, que possui suporte para 16 saídas analógicas e 8 canais de entradas e saídas digitais (CORPORATION, 2009). Já a transmissão de comando do controlador para o modelo foi implementada via cabo *Ethernet* com o protocolo da camada de transporte UDP.

Através da plataforma HIL, é possível verificar algumas limitações de *hardware* que não eram previstas na simulação em *software*. Assim, torna-se possível construir um cenário mais próximo da realidade, no qual se terá o controlador em *hardware* embarcado no satélite, conectado a sensores e atuadores por meio

de conexões analógicas.

Por fim, os dados coletados nas simulações em MIL e em HIL foram usados para gerar gráficos que foram utilizados para validar e comparar o desempenho dos controladores utilizados utilizando-se como parâmetros o tempo de assentamento (2%) do sistema, o sobressinal atingido e o respeito às restrições de oscilação da estrutura flexível do satélite.

1.5 Contribuições do Trabalho

Este trabalho traz algumas contribuições para a comunidade científica com relação ao controle de atitude de satélites rígido-flexíveis.

O problema de controle de atitude em satélites com estruturas flexíveis tem sido bastante abordado no meio científico. Muitos trabalhos abordam a técnica do LQR, por ser uma técnica de controle ótimo, simples de implementar, de baixo custo computacional, e que em geral obtém bons resultados.

Neste trabalho, além da técnica do LQR, é utilizada também uma técnica de controle preditivo para restringir as oscilações da estrutura flexível. A técnica utilizada, o MPC, possui características compatíveis com o tipo de problema atacado, ao lidar com restrições nas variáveis do problema.

As técnicas utilizadas mostraram-se eficientes para controlar não só o sistema linear do satélite, mas também seu modelo não-linear, dispensando métodos de controle mais complexos, baseados em modelos não-lineares.

Além disso, este trabalho não se limita a realizar simulações em *software*. Foram realizados também experimentos em *Hardware in the loop*, em que os controladores projetados são implementados e validados em *Hardware*, enquanto a planta é simulada em software numa máquina dedicada, deixando a simulação mais próxima do que seria o processo físico real.

O presente trabalho pretende suprir a falta de artigos abordando MPC para satélites rígido-flexíveis com validação em HIL. Uma das grandes contribuições desta dissertação consiste em ter implementado o algoritmo de controle MPC, utilizando parametrização para redução de custo computacional, embarcado em um *hardware* de baixo custo para simulação de tempo real.

1.6 Apresentação do Documento

Esta dissertação é composta de seis capítulos. O primeiro e presente capítulo consiste da introdução do trabalho, apresentando uma breve contextualização do problema envolvido, objetivos pretendidos com o trabalho, um breve resumo da metodologia, e as contribuições geradas pelo mesmo.

O segundo capítulo apresenta uma revisão de literatura com alguns trabalhos que realizam controle de atitude de satélites empregando diversas técnicas de controle, em especial as técnicas LQR e MPC. Alguns trabalhos sobre satélites com estruturas flexíveis também são mostrados, e são apresentados também trabalhos que empregam a metodologia do *Hardware in the loop* para testar controladores.

No terceiro capítulo, será mostrada a fundamentação teórica que embasa os métodos de controle utilizados neste trabalho. Além disso, é desenvolvido o modelo de satélite rígido-flexível utilizado, e também mostra a arquitetura de uma plataforma de teste de controladores *Hardware in the loop*, utilizada para as simulações deste trabalho.

No quarto capítulo, é apresentada a metodologia aplicada neste trabalho.

No quinto capítulo, são apresentados os resultados experimentais obtidos para os modelos de satélite utilizados na plataforma HIL, para as técnicas de controle testadas, e, por fim, no sexto capítulo são apresentadas as conclusões do trabalho, com uma sugestão de possíveis trabalhos futuros que deem continuidade a este.

Capítulo 2

Revisão Bibliográfica

2.1 Introdução

No presente capítulo, é apresentada uma breve revisão bibliográfica para motivar todo o estudo realizado nesta dissertação. Para ilustrar o estado-da-arte, serão mostrados neste capítulo trabalhos realizados sobre controle de atitude de satélites, em especial aqueles rígido-flexíveis, e também trabalhos que utilizam, entre outras, as técnicas de controle MPC e LQR. Por fim, serão mostrados também trabalhos que têm utilizado plataformas de *Hardware in the Loop* (HIL) para teste de controladores.

2.2 Revisão da Literatura

Souza (1992), em sua tese de doutorado, mostra um modelo rígido-flexível para dinâmica da configuração MB-1 da estação espacial. O autor projeta um controlador robusto utilizando o método do LQG, na intenção de enfraquecer as vibrações dos painéis flexíveis. O autor chegou à conclusão, após simulações numéricas, de que o controlador projetado atende bem aos requisitos, respondendo rapidamente a mudanças de atitude e amortecendo as vibrações indesejadas (SOUZA, 1992).

O artigo publicado por Gorinevsky e Vukovich (1998) mostra os resultados obtidos ao controlar um satélite com apêndice flexível realizando grandes manobras, cujo modelo é não-linear. O controlador idealizado pelos autores utiliza uma combinação entre retroalimentação (*feedback*) e alimentação direta (*feedforward*). O controlador implementado mostrou resultados bons, apesar de gerar oscilações no apêndice flexível (GORINEVSKY; VUKOVICH, 1998).

Silva (2000) investiga o comportamento de um satélite composto de um corpo rígido preso a um apêndice flexível com um amortecedor de vibrações ativo, localizado na extremidade livre do apêndice. Utilizando a abordagem lagrangiana, o modelo do satélite é desenvolvido. A autora chega à conclusão, através de resultados de simulação, de que é necessário um atuador para o controle de cada um dos modos de vibração da estrutura flexível (SILVA, 2000).

Pittet et al. (2000) projetaram um controlador para a dinâmica de um micro-satélite rígido-flexível baseado em desigualdades matriciais (LMI) e no filtro H_∞. Os autores implementam, junto ao modelo,

perturbações externas, atraso nos sensores, saturação nos atuadores e modos de vibração da estrutura flexível. Além do controlador projetado, um controlador de atraso de fase também é projetado, e os resultados de ambos, comparados. Os resultados mostraram que o controlador H_∞ teve melhor desempenho com relação à estabilidade, saturação dos atuadores, tempo de resposta e fator de amortecimento (PITTET; MIGNOT; FALLET, 2000).

Charbonnel (2002) realiza um trabalho para controle de sistemas espaciais com grandes estruturas flexíveis. A técnica de controle utilizada no trabalho é a do H_∞ com otimização LMI, e, segundo o autor, é o primeiro trabalho que utiliza esta técnica para realizar controle de modos de vibração (CHARBONNEL, 2002).

Gervini (2003) realizou um trabalho em que deseja controlar um sistema com estrutura flexível. O autor projetou um controlador LQG e o validou em simulação e também experimentalmente com um robô flexível. É proposto também no trabalho um controlador neural para o rastreamento de sinal do robô com elo flexível. Os resultados obtidos mostram que o controlador neural projetado possui melhor desempenho que o controlador LQG projetado anteriormente (GERVINI, 2003).

Fenili e Souza (2004) projetam um controlador para posicionamento de uma haste flexível ligada a um corpo rígido, usando como atuador um motor DC. A técnica utilizada, apesar de ser baseada em modelo linear, é também usada para controlar o sistema não-linear. A curvatura da haste é considerada linear, e os termos não-lineares aparecem devido a forças centrípetas. O desempenho do controlador projetado foi satisfatório (FENILI; SOUZA, 2004).

Tafazoli e Khorasani (2006) apresentam um trabalho em que analisam a estabilidade de um sistema de controle para a atitude de um satélite rígido-flexível utilizando o método de Lyapunov. O sistema foi implementado em um simulador, e o problema foi dividido em duas partes: a parte externa linear e a parte interna, não-linear e não-observável. Para qualquer uma das partes, os resultados visualizados pelo simulador atestaram estabilidade assintótica do sistema (TAFAZOLI; KHORASANI, 2006).

Valdivia (2007) realiza um estudo da influência da flexibilidade de um satélite rígido-flexível no controle de sua atitude. O satélite em estudo possui duas hastes flexíveis e um corpo rígido. Os controladores projetados para a análise foram o LQR e o LQG, e as abordagens utilizadas para desenvolver o modelo do satélite foram a abordagem lagrangiana e o método dos modos assumidos. Os resultados de simulação mostraram que o LQR apresenta ótimo resultado, porém ele depende da medida de todos os estados do sistema, o que não é factível na prática. O LQG aparece como solução para este problema, já que utiliza um filtro de Kalman para observar os estados não mensuráveis (VALDIVIA, 2007).

Em sua dissertação de mestrado, Lopes (2008) projetou controladores de atitude para dois modelos diferentes de satélite: um rígido e outro rígido-flexível. Foram utilizados os métodos GEO (Otimização Extrema Generalizada) com uma abordagem multi-objetivo e LQR para otimizar os ganhos das leis de controle. O autor ressalta a contribuição de seu trabalho ao afirmar que o problema de controle ótimo de atitude com abordagem multi-objetivo seria uma área de estudo ainda pouco explorada até então (LOPES, 2008).

Stinga et al. (2008) realizam um estudo de comparação entre as técnicas de controle LQR e MPC para o controle de atitude de um módulo com apêndice flexível, a nível de software. O módulo possui um grau de

liberdade de rotação. Tenta-se controlar seu posicionamento sem que haja muita vibração da haste flexível. O resultado a que chegam os autores é que a técnica do MPC, lidando com restrições sobre o sistema, consegue obter um melhor resultado para o sistema do que a técnica do LQR. As restrições na abordagem do MPC foram impostas à variável de comando (tensão de entrada do motor) e ao ângulo rígido, deixando o ângulo flexível livre de restrições (STINGA et al., 2008).

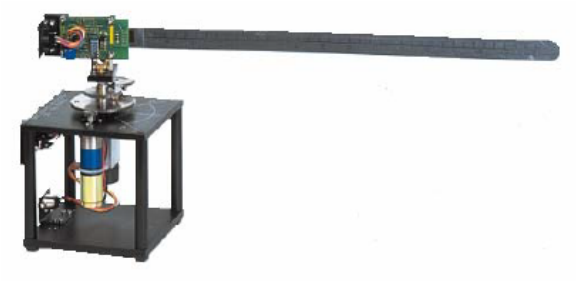


Figura 2.1: Módulo flexível utilizado por (STINGA et al., 2008)

No presente trabalho, também foi projetado um controlador MPC para controlar uma estrutura flexível, porém foram impostas também restrições ao ângulo flexível, diferentemente do trabalho de Stinga et al. (2008). Além disso, este trabalho diferencia-se do trabalho de Stinga et al. (2008) por ter realizado simulações em HIL além de simulações em *software*.

Zhang et al. (2008) projetaram controladores não-lineares para um satélite com duas hastes flexíveis. O algoritmo de controle utilizado pelos autores cai em um problema de otimização convexa restrito a LMI. Os resultados são apresentados através de simulações em software, mostrando que o controle projetado obteve bom desempenho, conseguindo amortecer as vibrações das hastes flexíveis (ZHANG et al., 2008).

Castro et al. (2009) realizam um trabalho de controle de atitude para satélites com estruturas flexíveis utilizando duas técnicas: LQR e filtro H_∞. As simulações realizadas atestam que o filtro H_∞ consegue resultados melhores que aqueles mostrados com a técnica LQR (CASTRO, 2009).

Gravade (2009) utiliza uma combinação de rodas de reação e atuadores piezoelétricos como atuadores para amortecer vibrações mecânicas em um painel flexível acoplado em um satélite. Para modelar a dinâmica do satélite, utilizou três modelos. O primeiro modelo utiliza parâmetros de massa e rigidez concentrados, resultando em um sistema linear. O segundo modelo utiliza o método dos modos assumidos, ou modelo de Rayleigh-Ritz, resultando em um modelo não linear, o qual é linearizado em etapa posterior. O último modelo utilizado no trabalho também usa o método dos modos assumidos, só que agora com a introdução de atuadores piezoelétricos. Este último modelo também é não-linear, e o autor o lineariza para prosseguir com as simulações. O autor faz simulações com os modelos nos softwares *Matlab* e ANSYS e conclui que vibrações em painéis flexíveis, induzidas por manobras de correção de atitude, podem ser amortecidas utilizando atuadores piezoelétricos (GRAVADE, 2009).

Cromwell (2011) fez um trabalho no *Massachussetts Institute of Technology* (MIT) com resultados práticos, em que constroem uma plataforma de testes em satélites. A plataforma, que possui três graus de liberdade, foi construída com rolamentos de ar, simulando, quase que perfeitamente, o ambiente ao redor de um satélite, com fricção reduzida e gravidade próxima de zero. O sistema utiliza rodas de reação como

atuadores de torque e pode ser utilizado para testar diversos tipos de leis de controle com a planta física real (CROMWELL, 2011).

Souza e Arena (2012) projetaram um simulador 3D para controlar atitude de satélites com modelo não-linear. Os atuadores do sistema são formados por jatos de gás e rodas de reação e possibilitam manobras nos três eixos. O controlador projetado, do tipo SDRE (*State-dependent Riccati Equation*), validou o simulador projetado (SOUZA; ARENA, 2012).

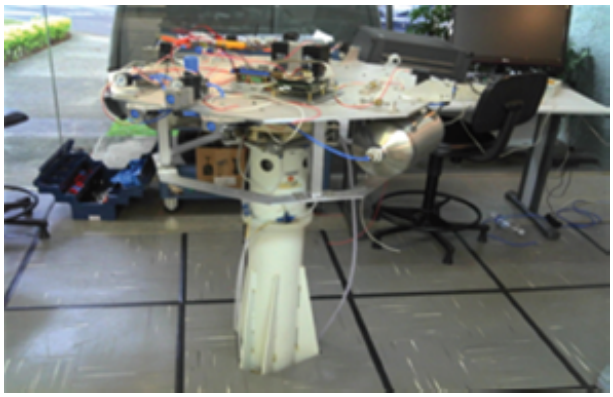


Figura 2.2: Simulador utilizado por (SOUZA; ARENA, 2012)

Saad et al. (2012) desenvolvem um modelo para uma haste flexível e realizam simulações para verificar seu comportamento. A haste é modelada como um modelo de Euler-Bernoulli, e a abordagem analítica de Lagrange é utilizada para obter as equações dinâmicas do sistema. Os testes foram realizados imprimindo um torque senoidal ao sistema, e o que se pôde perceber é que, para torques de baixa magnitude, os sistemas linear e não-linear possuem comportamentos semelhantes, o que não acontece para torques de grande amplitude (SAAD; AKHRIF; SAYDY, 2012).

O trabalho de Sales et al. (2013) mostra o desenvolvimento de um controlador que utiliza transdutores piezoelétricos para controlar a atitude de um satélite com dois painéis solares flexíveis. A análise dos resultados permitiu concluir que, controlando as vibrações dos painéis, também se controla os níveis de acoplamento entre os movimentos do corpo rígido e flexível, diminuindo bastante o consumo de energia (SALES; RADE; SOUZA, 2013).

Pinheiro e Souza (2013) projetam um sistema de controle de atitude para um modelo linear de micro-satélite, considerado de peso e tamanho reduzidos. O desafio é que, devido ao tamanho peculiar, o sistema é mais sensível a perturbações do meio e a incertezas em seu momento de inércia, por exemplo. O sistema de controle idealizado pelos autores é uma composição das técnicas H_2 e H_∞ utilizando otimização LMI (*Linear Matrix Inequality*). O controlador misto projetado mostrou boa robustez, característica do H_∞, e também baixo sinal de controle, contribuição do H_2 (PINHEIRO; SOUZA, 2013).

Cao e Chen (2014) projetam uma lei de controle aplicada a satélites. Sabendo que o uso de atuadores do tipo *magnetorquer* no controle de atitude em satélites acrescenta não-linearidades e variabilidade na dinâmica do sistema, os autores desenvolvem um controlador NMPC (*Nonlinear Model Predictive Control*), adequado a modelos não-lineares. Uma abordagem interessante utilizada nesse trabalho foi a utilização de frequências de amostragem variáveis na lei de controle. Isso permitiu uma melhor performance do controle

utilizado e menos esforço computacional, fazendo a frequência de amostragem ser alta quando o satélite está longe da posição desejada, e baixa quando o mesmo está próximo dela. Os resultados das simulações comprovaram a eficácia do controlador proposto (CAO; CHEN, 2014).

Chegeni et al. (2014) realizaram um trabalho em que utilizam a técnica *Generalized Incremental Predictive Control* para controlar a atitude de um satélite. A técnica implementada foi testada e comparada com a técnica de *Generalized Predictive Control*, constatando que aquela apresenta melhor desempenho que esta. No estudo, os autores utilizaram um modulador *Pulse-Width Pulse-Frequency* para modular o sinal de comando de torque, evitando, assim, não-linearidades na lei de controle (CHEGENI; ZANDIEH; EBRAHIMI, 2014).

Vicente e Ribeiro (2014), em seu trabalho, validam algumas leis de controle para atitude de satélites utilizando uma plataforma HIL. O objetivo dos autores é verificar se é possível testar um controlador de atitude de satélite com um micro-controlador de baixo custo em *Hardware in the loop*. As varáveis do sistema são os ângulos RPY (*Roll, Pitch e Yaw*), sendo desprezados os termos de acoplamento entre os ângulos. Uma lei de controle foi projetada no domínio discreto utilizando a técnica do lugar geométrico das raízes. A plataforma HIL utilizada foi construída com um micro-controlador PIC18F4550, uma placa de aquisição de dados DS1104 e um computador pessoal. O micro-controlador envia sinais de controle para a planta via saída PWM (*Pulse-Width Modulation*). O satélite foi modelado utilizando uma extensão do *Simulink*, o *Simulink Real-Time Workshop*, com troca de informação entre o micro-controlador e a planta em tempo real. Os resultados de simulação para uma referência degrau para os três ângulos RPY mostrou ruídos na resposta, além de divergência no regime transiente em relação ao esperado (VICENTE; RIBEIRO, 2014).

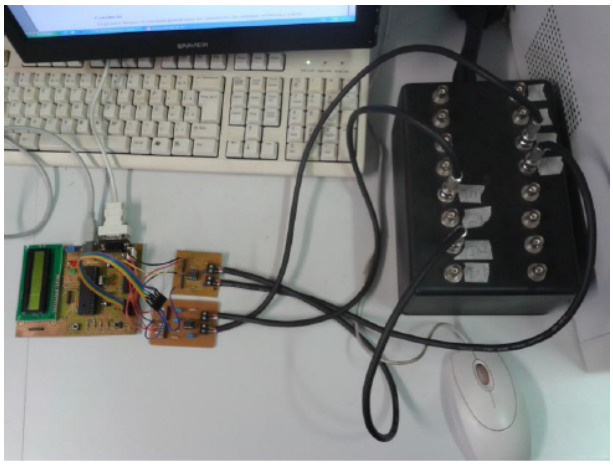


Figura 2.3: Plataforma HIL utilizada por (VICENTE; RIBEIRO, 2014)

O presente trabalho é similar ao de Vicente e Ribeiro, exceto pelo fato de que o controlador utilizado foi o MPC, e o controle foi implementado em uma placa *Beaglebone Black*.

Kukreti et al. (2015) estudam o controle de atitude de um *Cubesat*, satélite pequeno em formato cúbico, com atuadores magnéticos (*magnetorquers*). Os torques envolvidos nas equações dinâmicas do *Cubesat* consistem de torques de gradiente gravitacional e torques de controle magnético. Este ultimo

é resultado da interação do campo magnético local com os *magnetorquers*, bobinas de corrente elétrica que produzem momento magnético. O artigo compara o desempenho de 3 tipos de controladores: um controlador Proporcional-Derivativo (PD) simples, um controlador *Linear Quadratic Regulator* (LQR) comum e um controlador LQR genético, o qual consiste em utilizar um algoritmo heurístico para resolver o problema de otimização dos parâmetros do LQR. Os resultados mostram que os controladores PD e LQR obtêm resultados muito similares em termos de tempo de assentamento e comportamento em regime permanente. Já o LQR genético mostra resultados melhores que os outros dois métodos, com função de custo mais simples e maior robustez (KUKRETI et al., 2015).

Bayat (2015) montou uma plataforma de HIL de baixo custo para teste de controladores de atitude de satélites em tempo real. O autor utilizou a plataforma *Simulink Real-Time* do *Matlab* para implementar o modelo, e a conexão do modelo com o controlador foi feita através de uma placa de aquisição de dados. Por fim, o autor projetou ainda um simulador gráfico 3D para visualizar as manobras do satélite em tempo real. Os resultados obtidos em simulação validaram a plataforma montada (BAYAT, 2015).

Souza e Galvão (2015), em seu trabalho, realizam simulações para um satélite com estrutura flexível para validar controladores de atitude. O controlador escolhido foi o LQR. Os autores fazem, então, uma comparação do controlador LQR para satélites rígido-flexíveis quando modelados por dois diferentes modelos: modelo massa-mola (MSM, do inglês *Mass-Spring Model*) e modelo dos parâmetros distribuídos (DPM, do inglês *Distributed Parameters Model*) baseado no método dos modos assumidos. Os estados do sistema são o ângulo rígido, o ângulo flexível da haste flexível e suas respectivas derivadas em relação ao tempo, resultando em sistema linear de ordem quatro. Todos os estados são considerados conhecidos para a implementação da lei de controle do LQR, embora se reconheça a dificuldade em se medir a deflexão da haste flexível. Deseja-se controlar o deslocamento rígido e o deslocamento flexível, de modo que o deslocamento flexível amorteça o mais rápido possível. Foram então realizadas simulações no *Matlab/Simulink* com o controle LQR. Os resultados para controlar tanto o deslocamento rígido quanto o deslocamento flexível mostraram que o modelo DPM tem melhor desempenho. O mesmo acontece para o controle dos demais estados (SOUZA; GALVÃO, 2015).

Peixoto et al. (2016) fizeram simulações em *Matlab/Simulink* para controlar a atitude de um satélite rígido-flexível utilizando a técnica do MPC para restringir as vibrações da haste flexível do satélite. Foram utilizados dois modelos diferentes para o satélite: o modelo massa-mola e o modelo dos modos assumidos. Os autores chegam à conclusão de que a técnica de controle utilizada é uma boa alternativa para se controlar a atitude de satélites rígido-flexíveis, pois consegue manter as oscilações da haste flexível numa faixa aceitável para as manobras pretendidas (PEIXOTO; MURILO; SOUZA, 2016).

Eren et al. (2017) realizam um artigo de revisão sobre a técnica do MPC aplicado a sistemas aeroespaciais, reunindo mais de trezentos trabalhos que dão uma boa ideia do estado-da-arte no tema (EREN et al., 2017).

Souza (2017), em sua tese de doutorado, desenvolve algoritmos de controle robusto para controlar um satélite com estrutura flexível e também satélites com líquido em seu interior, o que é chamado de *sloshing*. A técnica utilizada pelo autor é a do H_∞, a qual trabalha com incertezas em alguns parâmetros do sistema (SOUZA, 2017).

Os trabalhos mostrados nesta seção sintetizam o estado-da-arte em controle de atitude de satélites

rígido-flexíveis, mostrando também alguns trabalhos que utilizam arquitetura de *Hardware in the loop* para validação dos controladores. Com essa revisão da literatura, pôde-se perceber que existem muitos trabalhos que utilizam o método do LQR para controlar a atitude de satélites rígido-flexíveis. Observou-se também que existem poucos trabalhos abordando a técnica do MPC para tal aplicação, ainda mais quando se trata da validação deste tipo de controlador em *Hardware in the loop*.

Dado esse contexto, o presente trabalho pretende suprir a falta de trabalhos abordando MPC para satélites rígido-flexíveis com validação em HIL.

Capítulo 3

Fundamentação Teórica

3.1 Introdução

Este capítulo apresenta os fundamentos teóricos necessários para a compreensão deste trabalho. Primeiramente, será explicado um pouco sobre as características de satélites em geral e, em específico, aqueles com estruturas flexíveis. Em seguida, é desenvolvido o modelo de satélite utilizado neste trabalho. Então, serão descritos detalhadamente os algoritmos de controle do LQR e do MPC, e, por fim, será explicada a arquitetura de uma plataforma *Hardware in the loop*, com suas características e vantagens na validação de controladores.

3.2 Satélites

São muitos os campos de atuação dos satélites artificiais em órbita ao redor da Terra. Algumas das aplicações são: missões militares, astronomia, telecomunicações, meteorologia, navegação e observação da terra.

Segundo Vicente e Ribeiro (2014), são quatro os problemas que definem a missão de um satélite, listados a seguir:

1. Lançamento do veículo, quando o satélite se encontra dentro de um foguete em direção ao local da missão;
2. Aquisição de órbita, quando o satélite atinge a órbita desejada;
3. Aquisição de atitude, quando o satélite aponta para a direção desejada;
4. Controle de atitude, que são correções da orientação do satélite quando o mesmo é desviado de seu apontamento ideal por torques ambientais.

Um satélite em órbita necessita ser orientado na posição correta, para que consiga enviar e receber sinais com o máximo de eficiência possível. Por isso, é interessante conhecer sua posição e orientação

angular em relação a um referencial bem conhecido. Este processo se refere ao controle de atitude do satélite, etapa de extrema importância e criticidade para que o satélite realize sua missão como desejado. Alguns satélites precisam de alta precisão, sendo necessário aplicar controle de atitude de 3 eixos. Outros têm controle de atitude em apenas um eixo, dependendo de sua aplicação.

A órbita na qual o satélite se encontra também influencia no controle de sua atitude. Por exemplo, satélites em órbitas mais baixas ficam sujeitos a maiores influências de torque de gradiente gravitacional, torque magnético e torque aerodinâmico do que satélites em órbitas mais altas, sendo necessário mais desempenho do controle de atitude (PINHEIRO; SOUZA, 2013).

Para entender melhor o significado de atitude de um satélite, consideremos os referenciais $X_0Y_0Z_0$ e XYZ que situam um satélite genérico, como mostrado na figura abaixo.

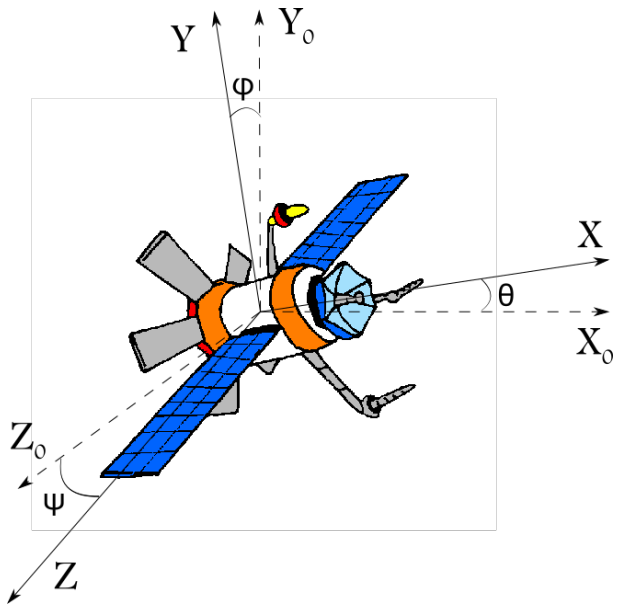


Figura 3.1: Referenciais em um satélite

O referencial $X_0Y_0Z_0$ é fixo na Terra e, portanto, pode ser considerado inercial. O referencial XYZ, por sua vez, é fixo ao satélite, e rotaciona juntamente com ele quando ele realiza alguma manobra.

Chamemos os ângulos que os eixos X, Y e Z fazem com os eixos X_0, Y_0 e Z_0 de θ, φ e ψ, respectivamente. Esses ângulos são normalmente chamados ângulos de Euler e representam a orientação angular do satélite a cada instante de tempo.

Na literatura, tais ângulos são chamados também de ângulos RPY (do inglês *Roll, Pitch, Yaw*), pois representam movimentos bem característicos no ramo aeroespacial. *Roll* é o movimento de rolagem (rotação em torno de X_0), *Pitch* é o movimento de arfagem (rotação em torno de Z_0), e *Yaw* é o movimento de guinada (rotação em torno de Y_0). Juntos, os ângulos de Euler formam o que se define como atitude do satélite.

O problema que o controle de atitude de um satélite busca resolver é, então, a estabilização do satélite em um determinado conjunto de ângulos (θ, φ, ψ), definidos pela missão a que o satélite deve atender.

Satélites possuem frequentemente partes flexíveis anexadas ao seu corpo, como antenas, velas, painéis solares e câmeras. Esses satélites são chamados comumente de satélites rígido-flexíveis. Um problema adicional aparece no controle de atitude de tais satélites, pois, ao realizar manobras, principalmente aquelas de grande amplitude, as estruturas flexíveis podem vibrar de modo indesejado, fazendo com que a missão do satélite seja prejudicada.

Nesse contexto, é importante que um tratamento especial seja dado a satélites rígido-flexíveis ao se controlar sua atitude.

3.2.1 Desenvolvimento do modelo de satélite rígido-flexível

O sistema abordado neste trabalho consiste de um satélite rígido-flexível. Tal satélite possui um corpo rígido cilíndrico acoplado a uma haste flexível, a qual pode representar uma antena, um painel solar, uma vela ou qualquer outro tipo de apêndice que possa sofrer vibrações quando o satélite se movimenta.

O desenvolvimento matemático utilizado nesta seção para chegar a um modelo de satélite rígido-flexível foi baseado nos trabalhos de Souza (2017) e Gu et al. (2005).

Para encontrar um modelo matemático que retrate a dinâmica do satélite, utiliza-se o esquema da Fig. (3.2).

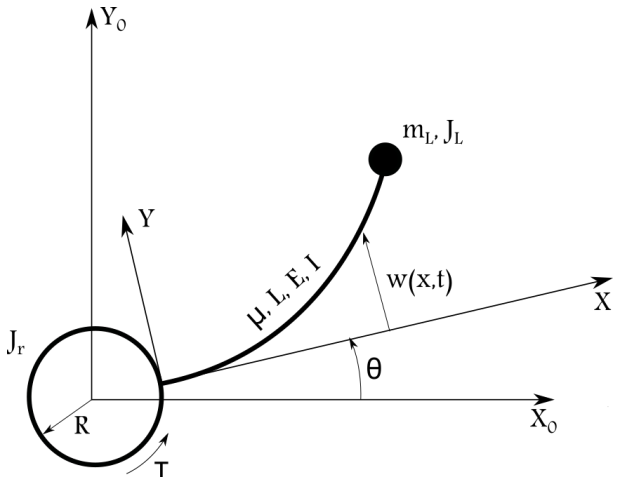


Figura 3.2: Modelo de Satélite rígido-flexível
Fonte: Adaptado de (GU; PETKOV; KONSTANTINOV, 2005)

No modelo representado na figura acima, o corpo rígido do satélite é um rotor que possui geometria cilíndrica, de raio R e momento de inércia J_r em relação ao eixo de simetria do rotor. Presa ao rotor, encontra-se a haste flexível, de comprimento L e densidade linear μ, e, em sua extremidade livre, encontra-se uma massa concentrada m_L de momento de inércia J_L em relação ao eixo do rotor. A haste possui também um coeficiente de amortecimento K_e e rigidez EI, onde E é o módulo de Young e I o momento de inércia seccional da haste.

Considera-se o referencial inercial X_0Y_0 e o referencial não-inercial XY preso ao corpo do satélite, rotacionando juntamente com ele. O ângulo que o corpo rígido gira em relação ao referencial X_0Y_0 é

$\theta(t)$, variável com o tempo t. A haste flexível, no entanto, deflete um ângulo maior que θ. Para modelar o deslocamento da haste flexível, consideramos, então, a função $w(x,t)$, que é a deformação elástica da haste medida em relação ao referencial XY, ou seja, na verdade, ela é a ordenada dos pontos da haste em relação a esse referencial. Essa deformação depende da posição x na haste e do tempo t, de forma que a extremidade livre da haste terá uma deformação $w(L,t)$ a cada instante de tempo t. O ângulo total que a extremidade da haste deflete então será chamado de $\alpha(t)$.

É importante lembrar que, embora a Fig. (3.2) mostre uma deformação considerável da haste para efeito didático, tal deformação será considerada de pequena magnitude, se comparada com as dimensões da haste.

O rotor será modelado possuindo uma componente de atrito viscoso b_m e gerando um torque τ.

Para desenvolver as equações dinâmicas do satélite rígido-flexível, considera-se a haste flexível como uma viga do tipo Euler-Bernoulli, cujas deformações elásticas obedecem à equação diferencial parcial mostrada na Eq. (3.1).

$$EI\frac{\partial^4 w(x,t)}{\partial x^4} + \mu\frac{\partial^2 w(x,t)}{\partial t^2} = 0 \tag{3.1}$$

Esta equação possui algumas condições de contorno. A extremidade da haste fixa ao corpo rígido, por exemplo, não possui deslocamento flexível em relação a este. Além disso, a extremidade livre da haste é livre de esforços. Desta forma, devemos ter as seguintes condições de contorno para resolver a Eq. (3.1):

$$\begin{aligned} w(0,t) &= 0 \\ \frac{\partial w(0,t)}{\partial x} &= 0 \\ \frac{\partial^2 w(L,t)}{\partial x^2} &= 0 \\ \frac{\partial^3 w(L,t)}{\partial x^3} - \frac{m_L}{EI}\frac{\partial^2 w(L,t)}{\partial t^2} &= 0 \end{aligned} \tag{3.2}$$

As equações acima formam o que se chama problema de valor de contorno (PVC). Um técnica bastante utilizada na literatura para resolver um PVC é a técnica da separação de variáveis. Nesta técnica, supomos que a solução é um produto entre uma função do tempo $T(t)$ e outra função do espaço $X(x)$, como na equação abaixo.

$$w(x,t) = X(x)T(t) \tag{3.3}$$

Substituindo essa solução na equação e aplicando as condições de contorno, encontramos a seguinte solução para a função dependente do espaço (GU; PETKOV; KONSTANTINOV, 2005):

$$X(x) = K\left[cosh(\beta_i x) - cos(\beta_i x) - \left(\frac{cosh(\beta_i L) + cos(\beta_i L)}{senh(\beta_i L) + sen(\beta_i L)}\right)(senh(\beta_i x) - sen(\beta_i x))\right] \quad (3.4)$$

Nesta função, K é uma constante de normalização e β_i são as raízes da equação transcendental abaixo:

$$1 + cosh(\beta L)cos(\beta L) + \frac{m_L}{\mu L}\beta L(senh(\beta L)cos(\beta L) - cosh(\beta L)sen(\beta L)) = 0 \quad (3.5)$$

Utiliza-se o método dos modos assumidos para representar o deslocamento flexível da haste. Este método propõe que o deslocamento flexível seja um somatório do produto entre uma função de forma $\varphi(x)$ e uma função dependente do tempo $\eta(t)$, para n modos de vibração, como na equação abaixo.

$$w(x,t) = \sum_{i=1}^{n} \varphi_i(x)\eta_i(t) \quad (3.6)$$

Nesta equação $\varphi_i(x)$ são as autofunções, e $\eta_i(t)$ são coordenadas generalizadas que representam como as funções de forma variam no tempo. As funções de forma $\varphi_i(x)$ devem satisfazer às seguintes condições de normalização (GU; PETKOV; KONSTANTINOV, 2005):

$$\mu \int_0^L \varphi_i(x)\varphi_j(x)dx + m_L\varphi_i(L)\varphi_j(L) = 0, i \neq j \quad (3.7)$$

$$\mu \int_0^L \varphi_i^2(x)dx + m_L\varphi_i^2(L) = 1 \quad (3.8)$$

Utilizando a solução $X(x)$ como função de forma, pode-se obter o valor da constante C através das equações de normalização. Desta forma, pode-se calcular as frequências naturais de cada modo de vibração, ω_i, através da Eq. (3.9).

$$\omega_i = \beta_i^2\sqrt{\frac{EI}{\mu}} \quad (3.9)$$

Utiliza-se a abordagem analítica de Lagrange para derivar as equações dinâmicas do satélite. A abordagem lagrangiana utiliza a energia do sistema no seu desenvolvimento, portanto calcula-se a energia cinética e a energia potencial do sistema.

A energia cinética do sistema pode ser calculada como a soma das energias cinéticas do rotor, da haste e da massa concentrada.

$$E_c = E_c(rotor) + E_c(haste) + E_c(massa) \quad (3.10)$$

Desenvolvendo a expressão de cada uma das energias das partes do sistema, obtém-se:

$$E_c = \frac{1}{2}\dot{\theta}^2\left(J_r + \mu\left(\int_0^L w^2 dx + \frac{1}{3}(R+L)^3 - \frac{1}{3}R^3\right) + m_L(w_L^2 + (R+L)^2) + J_L\right) + \frac{1}{2}\dot{\theta}\left(2\mu\int_0^L \dot{w}(R+s(x))dx + 2m_L\dot{w}_L + J_L\left(\frac{\partial \dot{w}}{\partial x}\right) + \frac{1}{2}\left(\mu\int_0^L \dot{w}^2 dx + m_L\left(\frac{\partial \dot{w}}{\partial x}\right)\right)\right) \tag{3.11}$$

Na Eq. (3.11), a função s representa a distância da origem do sistema de coordenadas XY até um elemento de massa da haste.

Expressando as funções de forma $\varphi_i(x)$ e coordenadas generalizadas $\eta_i(t)$ na forma matricial, obtém-se:

$$\phi = \begin{bmatrix}\varphi_1(x) & \varphi_2(x) & \dots & \varphi_n(x)\end{bmatrix}^T \quad e \quad \chi = \begin{bmatrix}\eta_1(t) & \eta_2(t) & \dots & \eta_n(t)\end{bmatrix}^T \tag{3.12}$$

A energia potencial do sistema pode ser dada pela Equação:

$$E_p = \frac{1}{2}EI\int_0^L \left(\frac{\partial^2 w}{\partial x^2}\right)^2 dx \tag{3.13}$$

Esse sistema é ainda sujeito a uma função de dissipação de Rayleigh R, como mostrada na equação abaixo (SOUZA, 2017):

$$R = \frac{1}{2}b_m\dot{\theta}^2 + \frac{1}{2}K_eEI\int_0^L \left(\frac{\partial^2 \dot{w}}{\partial x^2}\right)^2 dx \tag{3.14}$$

É possível então escrever a função de Lagrange L, também conhecida como lagrangiana:

$$L = E_c - E_p \tag{3.15}$$

De acordo com o método lagrangiano, a função de Lagrange deve satisfazer à seguinte equação:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial q_i}\right) - \frac{\partial L}{\partial q_i} + \frac{\partial R}{\partial q_i} = \Gamma_i \tag{3.16}$$

Nesta equação, q_i são as coordenadas generalizadas, neste caso representadas por $q = \begin{bmatrix}\theta & \eta_1(t) & \eta_2(t)\dots\eta_n(t)\end{bmatrix}$, e Γ representa o torque generalizado $\Gamma = \begin{bmatrix}\tau & 0 & 0\dots 0\end{bmatrix}$.

Aplicando a lagrangiana na Eq. (3.16), obtém-se as equações dinâmicas do satélite:

$$\begin{aligned}\ddot{\theta}[I_t + \chi^T C_{rr}\chi] + \ddot{\chi}^T M_{rf} + \dot{\theta}[2\chi^T C_{rr}\dot{\chi} + b_m] &= \tau \\ \ddot{\theta}M_{rf} + M_{ff}\ddot{\chi} - \dot{\theta}^2 C_{rr}\chi + K_{ff}\chi + B_{ff}\dot{\chi} &= 0\end{aligned} \tag{3.17}$$

Na equação acima, as constantes possuem as seguintes expressões:

$$
\begin{aligned}
I_t &= J_r + \frac{1}{3}\mu((R+L)^3 - R^3) + m_L(R+L)^2 + J_L \\
C_{rr} &= \mu \int_0^L \phi\phi^T dx + m_L\phi_L\phi_L^T \\
M_{ff} &= \mu \int_0^L \phi\phi^T dx + m_L\phi_L\phi_L^T + \frac{1}{2}J_L \frac{d\phi_L}{dx}\frac{d\phi_L}{dx}^T \\
M_{rf} &= \mu \int_0^L \phi(R+x)dx + m_L(R+L)\phi_L + \frac{1}{2}J_L \frac{d\phi_L}{dx}\frac{d\phi_L}{dx}^T \\
B_{ff} &= K_e EI \int_0^L \frac{d^2\phi}{dx^2}\frac{d^2\phi}{dx^2}^T dx \\
K_{ff} &= EI \int_0^L \frac{d^2\phi}{dx^2}\frac{d^2\phi}{dx^2}^T dx
\end{aligned}
\tag{3.18}
$$

Com esses valores conhecidos, consegue-se expressar os estados x do sistema na forma $\dot{x} = f(x, u)$ onde f é função não-linear dos estados x e da variável de comando u.

Com o auxílio da Eq. (3.9), podemos calcular as frequências naturais do sistema para cada modo de vibração. A tabela abaixo mostra as frequências naturais para os cinco primeiros modos de vibração do modelo em questão.

Tabela 3.1: Frequências naturais de vibração da estrutura flexível para os cinco primeiros modos

Modo de Vibração	**Frequência (ω)**
1	6,0 rad/s
2	45,5 rad/s
3	136,4 rad/s
4	277,3 rad/s
5	469,9 rad/s

Visto que as frequências, a partir do terceiro modo de vibração, têm valor relativamente alto em relação às duas primeiras, optou-se por utilizar um modelo com apenas dois modos de vibração (GU; PETKOV; KONSTANTINOV, 2005).

As equações de estado obtidas para dois modos de vibração são as mostradas abaixo.

$$
\begin{aligned}
\ddot{\theta} &= (\tau + M_{rf}(1,1)K_{ff}(1,1)\eta_1 + M_{rf}(2,1)K_{ff}(2,2)\eta_2 + M_{rf}(1,1)B_{ff}(1,1)\dot{\eta_1} + M_{rf}(2,1)B_{ff}(2,2)\dot{\eta_2} - 2\dot{\theta}\dot{\eta_1}\eta_1 \\
&\quad -2\dot{\theta}\dot{\eta_2}\eta_2 - bm\dot{\theta} - M_{rf}(1,1)\dot{\theta}^2\eta_1 - M_{rf}(2,1)\dot{\theta}^2\eta_2)/(I_t + \eta_1^2 + \eta_2^2 - M_{rf}(1,1)^2 - M_{rf}(2,1)^2) \\
\ddot{\eta_1} &= \dot{\theta}^2\eta_1 - K_{ff}(1,1)\eta_1 - B_{ff}(1,1)\dot{\eta_1} - M_{rf}(1,1)((\tau + M_{rf}(1,1)K_{ff}(1,1)\eta_1 + M_{rf}(2,1)K_{ff}(2,2)\eta_2 \\
&\quad +M_{rf}(1,1)B_{ff}(1,1)\dot{\eta_1} + M_{rf}(2,1)B_{ff}(2,2)\dot{\eta_2} - 2\dot{\theta}\dot{\eta_1}\eta_1 - 2\dot{\theta}\dot{\eta_2}\eta_2 - bm\dot{\theta} - M_{rf}(1,1)\dot{\theta}^2\eta_1
\end{aligned}
$$

$$
\begin{aligned}
& -M_{rf}(2,1)\dot{\theta}^2\eta_2)/(I_t+\eta_1^2+\eta_2^2-M_{rf}(1,1)^2-M_{rf}(2,1)^2)) \\
\ddot{\eta_2} &= \dot{\theta}^2\eta_2 - K_{ff}(2,2)\eta_2 - B_{ff}(2,2)\dot{\eta_2} - M_{rf}(2,1)((\tau + M_{rf}(1,1)K_{ff}(1,1)\eta_1 + M_{rf}(2,1)K_{ff}(2,2)\eta_2 \\
& +M_{rf}(1,1)B_{ff}(1,1)\dot{\eta_1} + M_{rf}(2,1)B_{ff}(2,2)\dot{\eta_2} - 2\dot{\theta}\dot{\eta_1}\eta_1 - 2\dot{\theta}\dot{\eta_2}\eta_2 - bm\dot{\theta} - M_{rf}(1,1)\dot{\theta}^2\eta_1 \\
& -M_{rf}(2,1)\dot{\theta}^2\eta_2)/(I_t+\eta_1^2+\eta_2^2-M_{rf}(1,1)^2-M_{rf}(2,1)^2))
\end{aligned}
\tag{3.19}
$$

Nas equações dos estados acima, utilizou-se a notação $A(i,j)$ para representar o elemento da linha i e coluna j da matriz A.

Percebe-se que as equações mostradas na Eq. (3.19) são não-lineares, pois contêm produtos entre variáveis dos sistema, como os termos $\chi^T C_{rr}\chi$, $2\chi^T C_{rr}\dot{\chi}$ e $\dot{\theta}^2 C_{rr}\chi$.

É interessante então linearizar o sistema em torno de um ponto de operação. Tal linearização é importante quando se deseja, por exemplo, projetar leis de controle baseadas em modelo linear, como é o caso das técnicas de controle LQR e MPC, objetos de estudo deste trabalho. Linearizando o sistema através do método de Taylor, encontramos o seguinte sistema linear no espaço de estados:

$$
\dot{x} = Ax + Bu \tag{3.20}
$$

Nessa equação, x representa o vetor de estados, u representa a variável de comando (neste caso, torque gerado no rotor) e A e B são a matriz de transmissão de estados e a matriz de entrada do sistema, respectivamente, calculadas segundo as seguintes expressões:

$$
A = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \quad e \quad B = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial u_2} & \cdots & \frac{\partial f_1}{\partial u_n} \\ \frac{\partial f_2}{\partial u_1} & \frac{\partial f_2}{\partial u_2} & \cdots & \frac{\partial f_2}{\partial u_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial u_1} & \frac{\partial f_n}{\partial u_2} & \cdots & \frac{\partial f_n}{\partial u_n} \end{bmatrix} \tag{3.21}
$$

A linearização é feita em torno do ponto de operação $\begin{bmatrix} \theta & \chi \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \end{bmatrix}$. O sistema linearizado em torno do ponto de operação fica então dado pela Eq. (3.22).

$$
\begin{aligned}
\ddot{\theta} &= (\tau + M_{rf}(1,1)K_{ff}(1,1)\eta_1 + M_{rf}(2,1)K_{ff}(2,2)\eta_2 + M_{rf}(1,1)B_{ff}(1,1)\dot{\eta_1} + M_{rf}(2,1)B_{ff}(2,2)\dot{\eta_2} \\
& -bm\dot{\theta})/(I_t - M_{rf}(1,1)^2 - M_{rf}(2,1)^2) \\
\ddot{\eta_1} &= -K_{ff}(1,1)\eta_1 - B_{ff}(1,1)\dot{\eta_1} - M_{rf}(1,1)((\tau + M_{rf}(1,1)K_{ff}(1,1)\eta_1 + M_{rf}(2,1)K_{ff}(2,2)\eta_2 \\
& +M_{rf}(1,1)B_{ff}(1,1)\dot{\eta_1} + M_{rf}(2,1)B_{ff}(2,2)\dot{\eta_2} - bm\dot{\theta})/(I_t - M_{rf}(1,1)^2 - M_{rf}(2,1)^2)) \\
\ddot{\eta_2} &= \dot{\theta}^2\eta_2 - K_{ff}(2,2)\eta_2 - B_{ff}(2,2)\dot{\eta_2} - M_{rf}(2,1)((\tau + M_{rf}(1,1)K_{ff}(1,1)\eta_1 + M_{rf}(2,1)K_{ff}(2,2)\eta_2 \\
& +M_{rf}(1,1)B_{ff}(1,1)\dot{\eta_1} + M_{rf}(2,1)B_{ff}(2,2)\dot{\eta_2} - bm\dot{\theta})/(I_t - M_{rf}(1,1)^2 - M_{rf}(2,1)^2))
\end{aligned}
\tag{3.22}
$$

Os dois modelos (linear e não-linear) foram simulados em malha aberta, obtendo comportamentos

similares, como se pode perceber no gráfico da Figura 3.3, que representa o deslocamento do satélite para uma excitação em degrau em malha aberta.

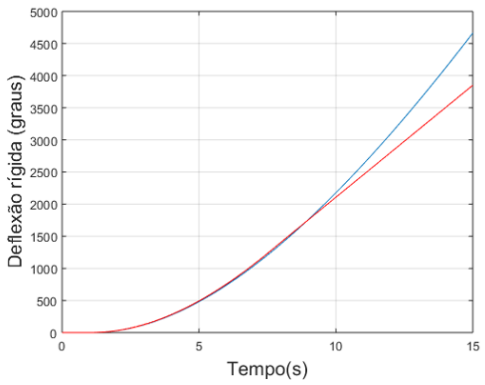


Figura 3.3: Comportamento do sistema linear (em vermelho) e do sistema não-linear (em azul) em malha aberta para uma excitação degrau

Percebe-se então que os dois modelos são bem próximos, indicando fracas não-linearidades.

3.3 Controle Preditivo Baseado em Modelo (MPC)

A técnica de controle preditivo baseado em modelo, também conhecida como MPC (do inglês *Model Predictive Control*) foi desenvolvida e tem sido aplicada na indústria principalmente a partir da década de 80 (MAYNE et al., 2000). A grande vantagem da utilização de controle preditivo é que ele oferece um método sistemático para o tratamento de restrições do sistema que se deseja controlar. Tais restrições são, muitas vezes, inerentes ao sistema e não podem ser negligenciadas ao se projetar uma lei de controle. Alguns tipos de restrições bastante encontradas em problemas práticos são: restrições nos atuadores do sistema, restrições de segurança, restrições de qualidade de produto ou referentes a legislação específica, entre outras (ALAMIR, 2013). Para exemplificar algumas restrições, pode-se pensar nas seguintes situações hipotéticas:

- Um gerador que só consegue fornecer no máximo 12 V em sua saída. Essa é uma restrição que deve ser levada em conta para um sistema que utilize a tensão de saída desse gerador;
- Uma máquina que só consiga operar bem numa faixa de temperaturas delimitada (por exemplo, de $0°C$ a $150°C$). Temperaturas fora dessa faixa podem danificar o processo que a máquina realiza ou ameaçar a segurança dos operadores da mesma. Desta forma, essas limitações não devem ser ignoradas no projeto de controle das variáveis da máquina;
- Um motor rotativo que só consiga fornecer um torque de até 1 N.m, usado como atuador em um sistema, não irá conseguir controlar muito bem a planta se for solicitado um torque maior que aquele. Desta forma, é importante incluir no projeto da lei de controle a sua restrição no torque gerado.

O algoritmo de controle preditivo baseia-se na minimização de uma função custo escolhida pelo projetista, levando-se em conta as restrições do sistema. A variável de controle gerada é, então, a variável ótima para controlar o sistema, considerando a função custo utilizada, dentro das possibilidades que o sistema tem de ficar dentro dos limites impostos pelas restrições.

Tendo-se escolhido uma função custo adequada, o que o algoritmo do MPC faz é, a cada instante de amostragem, medir os estados atuais do sistema, calcular a melhor sequência de comandos que minimizam a função custo para aqueles estados e aplicar apenas a primeira ação de controle no próximo instante de amostragem.

É interessante notar que, apesar de, a cada instante de tempo, o algoritmo fornecer uma sequência de N comandos para serem transmitidos à planta entre os instantes k e $k+N-1$, apenas um (o primeiro) é de fato aplicado. No instante de amostragem seguinte, serão calculados novamente N outros comandos, e será feita a aplicação apenas do primeiro novamente. Isso se dá porque o algoritmo do controle preditivo tem o poder de prever, como o próprio nome sugere, o impacto de cada sequência de comando no valor da função custo (ALAMIR, 2013). Então, pode ser que a sequência de controle calculada no instante k, já não seja a sequência ótima para o instante seguinte $(k+1)$. Por isso, aplica-se apenas o primeiro comando e, a cada instante de amostragem, calcula-se uma nova sequência de controle.

Será utilizada, no decorrer desta dissertação, a notação $x[k]$ para representar o vetor de estados do sistema no instante discreto k, e a notação $u[k]$ para denotar o sinal de comando/controle no instante discreto k. Como a formalização do MPC é toda no domínio discreto, considera-se um período de amostragem T_s que relaciona o tempo no domínio contínuo t com o mesmo no domínio discreto, de forma que $t = k.T_s$.

A sequência de controle que foi mencionada acima será então representada pela seguinte notação:

$$\tilde{u}[k] = \begin{bmatrix} u[k] \\ u[k+1] \\ u[k+2] \\ \vdots \\ u[k+N-1] \end{bmatrix} \tag{3.23}$$

A sequencia $\tilde{u}[k]$ gera uma sequencia de vetores de estado, que será representada por:

$$\tilde{x}[k] = \begin{bmatrix} x[k+1] \\ x[k+2] \\ x[k+3] \\ \vdots \\ x[k+N] \end{bmatrix} \tag{3.24}$$

Essa notação está coerente, pois o comando $u[k]$ gera o próximo vetor de estados $x[k+1]$, de forma que o último elemento da sequência de comandos, $u[k+N-1]$, gera o vetor de estados $x[k+N]$. Como já mencionado, apenas o primeiro comando, $u[k]$, é aplicado à planta a cada instante k.

O número N que vem sendo repetidamente mencionado no texto é chamado de horizonte de predi-

ção. Ele é um número natural escolhido pelo projetista e significa o número de iterações à frente que o controlador estará "olhando"para "prever"a melhor sequência de controle para aquele instante.

Há um *trade-off* na escolha da magnitude do horizonte de predição. Quando houver um modelo eficaz da planta, o aumento do horizonte de predição implica em um sinal de controle mais eficiente, pois o algoritmo estará prevendo mais iterações à frente. Entretanto, como será mostrado mais adiante neste texto, o aumento do horizonte de predição implica numa complexidade maior para o cálculo da sequência de controle, o que pode deixar o algoritmo inviável de ser realizado na prática. Por isso, deve-se escolher um horizonte de predição adequado para cada tipo de aplicação.

Neste ponto, define-se uma forma de selecionar algum vetor de estado específico, $x[k+i]$, ou comando, $u[k+i]$, dentro dos vetores N-dimensionais $\tilde{u}[k]$ e $\tilde{x}[k]$. Para isso, definimos a matriz de seleção Π, a qual seleciona o vetor de estados $x[k+i]$ ou o vetor de comandos $u[k+i-1]$ dentro das sequências de estados e comandos $\tilde{x}$ e $\tilde{u}$, como sugerem as equações (3.25) e (3.26) abaixo.

$$x[k+i] = \Pi_i^{(n,N)} \tilde{x}[k] \tag{3.25}$$

$$u[k+i-1] = \Pi_i^{(n_u,N)} \tilde{u}[k] \tag{3.26}$$

Nas equações acima, n é o número de estados do vetor x, e n_u é o número de variáveis de comando, ou, em outras palavras, o número de atuadores do sistema.

A matriz Π pode ser definida da seguinte forma:

$$\Pi_i^{(n,N)} = \begin{bmatrix} O_{n\times n} & \dots & O_{n\times n} & I_{n\times n} & O_{n\times n} & \dots & O_{n\times n} \end{bmatrix} \in R^{n\times(Nn)} \tag{3.27}$$

A matriz de seleção, como mostrada na equação acima, é composta por $i-1$ matrizes nulas de dimensão $n \times n$, uma matriz identidade de ordem n e mais $n - i$ matrizes nulas de dimensão $n \times n$.

A definição da função custo é um passo importante no projeto de um algoritmo de controle preditivo. É interessante que a função custo, denotada por J, possa indicar o quão próximo da referência desejada está o estado que se deseja regular. Analogamente, pode-se incluir na função custo uma parcela que indique quão próxima está a variável de comando de um valor desejado.

Por exemplo, em um sistema simples de um motor rotativo acoplado a uma barra, pode-se desejar que a barra siga uma certa referência angular. Neste caso, o valor desejado para o estado (posição da barra) está definido pela referência, e o valor desejado para o comando (torque do motor) é zero, pois é ideal que, ao se atingir a referência, o atuador pare de agir, imprimindo torque nulo na barra.

Com esta motivação, pode-se definir uma função custo que implemente um erro quadrático para o estado e para a variável de comando da seguinte forma:

$$J = \sum_{i=1}^{N} \|y[k+i] - y_d[k+i]\|_{Q_y}^2 + \sum_{i=1}^{N} \left\| \Pi_i^{(n_u,N)} \tilde{u} - u_d \right\|_{Q_u}^2 \tag{3.28}$$

Nesta equação, é utilizada a notação $\|a\|_P^2$, que significa a norma ponderada do vetor a, utilizando a matriz P como matriz de ponderação. Essa norma ponderada pode ser calculada de acordo com a seguinte definição:

$$\|a\|_P^2 = a^T P a \tag{3.29}$$

A matriz de ponderação P é uma matriz quadrada simétrica positivo definida, e seus elementos indicam o quão priorizados devem ser os respectivos estados regulados ao acompanharem as referências desejadas. Em geral, opta-se por utilizar uma matriz diagonal como matriz de ponderação para evitar acoplamento entre termos.

Na Eq. (3.28), a matriz de ponderação para o vetor de estados regulados é chamada de Q_y, e a matriz de ponderação utilizada para a variável de comando é chamada de Q_u. O vetor y é o vetor de estados regulados, y_d é o valor desejado de y, ou seja, a referência a ser seguida, e u_d representa o valor desejado para a variável de comando.

O vetor de estados regulados é dado pela expressão:

$$y_r = C_r x \tag{3.30}$$

Sendo x uma matriz coluna e C_r uma matriz constante, y_r vem a ser um vetor formado por combinações lineares dos estados que desejam ser regulados. Por exemplo, caso se deseje regular o estado x_1 e a soma dos estados x_2 e x_3 de um vetor hipotético x formado pelos estados x_1, x_2 e x_3, a matriz C_r adequada deve ser $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}$, de forma que $y_r = \begin{bmatrix} x_1 \\ x_2 + x_3 \end{bmatrix}$.

A planta a ser controlada, considerada linear, será representada no espaço de estados no domínio discreto. A equação abaixo representa o espaço de estados:

$$x[k+1] = Ax[k] + Bu[k] \tag{3.31}$$

Nesta equação, as matrizes A e B são constantes, o que acaba definindo o sistema como linear e invariante no tempo (LIT). O conhecimento dessas matrizes do modelo é fundamental para que o algoritmo do MPC seja desenvolvido, daí seu nome "controle preditivo baseado em modelo".

É fácil verificar com algumas manipulações algébricas matriciais que, utilizando a Eq. (3.31), a expressão para o vetor de estados no $(k+i)$-ésimo instante é dado por:

$$x[k+i] = A^i x[k] + \begin{bmatrix} A^{i-1}B & A^{i-2}B & \dots & AB & B \end{bmatrix} \begin{bmatrix} u[k] \\ \vdots \\ u[k+i-2] \\ u[k+i-1] \end{bmatrix} \tag{3.32}$$

Para simplificar a expressão da equação acima, pode-se utilizar a matriz de seleção Π definida na Eq.

(3.27). A equação ficará então:

$$x[k+i] = A^i x[k] + \begin{bmatrix} A^{i-1}B & A^{i-2}B & \dots & AB & B \end{bmatrix} \begin{bmatrix} \Pi_1^{(n_u,N)} \\ \vdots \\ \Pi_{i-1}^{(n_u,N)} \\ \Pi_i^{(n_u,N)} \end{bmatrix} \tilde{u}[k] \tag{3.33}$$

Essa equação, de uma forma mais compacta fica igual a

$$x[k+i] = \Phi_i x[k] + \Psi_i \tilde{u}[k] \tag{3.34}$$

Na Eq. (3.34), Φ e Ψ são matrizes que dependem de A e B. Desta forma, tem-se uma forma de expressar o vetor de estados no $(k+i)$-ésimo instante de tempo, em função do estado atual $x[k]$ e da variável de comando atual $u[k]$.

Para concatenar todos os vetores de estado, quando i varia de 0 a N, a equação acima fica reduzida a :

$$\tilde{x}[k] = \Phi x[k] + \Psi \tilde{u} \tag{3.35}$$

Onde Φ e Ψ equivalem a

$$\Phi = \begin{bmatrix} \Phi_1 \\ \vdots \\ \Phi_N \end{bmatrix} \quad e \quad \Psi = \begin{bmatrix} \Psi_1 \\ \vdots \\ \Psi_N \end{bmatrix} \tag{3.36}$$

Utilizando então a Eq. (3.34), pode-se expressar o vetor de variáveis reguladas y_r como

$$y_r[k+i] = C_r x[k+i] = C_r \Phi_i x[k] + C_r \Psi_i \tilde{u} \tag{3.37}$$

Voltando à expressão da função custo mostrada na Eq. (3.28), pode-se escrevê-la agora da seguinte forma:

$$J = \sum_{i=1}^{N} \|C_r \Phi_i x[k] + C_r \Psi_i \tilde{u} - y_d[k+i]\|_{Q_y}^2 + \sum_{i=1}^{N} \left\| \Pi_i^{(n_u,N)} \tilde{u} - u_d \right\|_{Q_u}^2 \tag{3.38}$$

Desenvolvendo o somatório de normas dessa função custo, obtém-se:

$$\begin{aligned} J \;=\; & \tilde{u}^T \left[\sum_{i=1}^{N} \Psi_i^T C_r^T Q_y C_r \Psi_i \right] \tilde{u} + \left[\left(2 \sum_{i=1}^{N} \Psi_i^T C_r^T Q_y C_r \Phi_i \right) x[k] - \left(2 \sum_{i=1}^{N} \Psi_i^T C_r^T Q_y \Pi_i^{(n_u,N)} \right) \tilde{y}_d[k] \right]^T \tilde{u} + \\ & \left\| \sum_{i=1}^{N} (C_r \Phi_i x[k] - y_d[k+i]) \right\|_{Q_y}^2 + \tilde{u}^T \left[\sum_{i=1}^{N} \left(\Pi_i^{(n_u,N)} \right)^T Q_u \left(\Pi_i^{(n_u,N)} \right) \right] \tilde{u} + \left[2 \sum_{i=1}^{N} \left(\Pi_i^{(n_u,N)} \right)^T Q_u u_d \right]^T \tilde{u} + \|u_d\|_{Q_u}^2 \end{aligned}$$

(3.39)

A função custo pode então ser simplificada para a seguinte notação:

$$J = \frac{1}{2}\tilde{u}^T H \tilde{u} + F^T \tilde{u} + Constante \tag{3.40}$$

Na equação acima, F é uma matriz que assume o seguinte valor:

$$F = F_1 x[k] + F_2 \tilde{y}_d + F_3 u_d \tag{3.41}$$

E as matrizes H, F_1, F_2 e F_3 assumem os seguintes valores:

$$\begin{aligned}
H &= 2\sum_{i=1}^{N}\left[\Psi_i^T C_r^T Q_y C_r \Psi_i + \left(\Pi_i^{(n_u,N)}\right)^T Q_u \left(\Pi_i^{(n_u,N)}\right)\right] \\
F_1 &= 2\sum_{i=1}^{N}\Psi_i^T C_r^T Q_y C_r \Phi_i \\
F_2 &= -2\sum_{i=1}^{N}\Psi_i^T C_r^T Q_y \Pi_i^{(n_u,N)} \\
F_3 &= 2\sum_{i=1}^{N}\left(\Pi_i^{(n_u,N)}\right)^T Q_u
\end{aligned} \tag{3.42}$$

A matriz H na Eq. (3.40) é chamada de matriz Hessiana da função. É interessante notar que a função custo agora tomou a forma de um problema quadrático na variável $\tilde{u}$. Isso significa que existe uma sequência de comando $\tilde{u}$ ótima que minimiza aquela função custo.

Uma vez formalizada a função custo, deve-se agora formalizar sistematicamente a utilização de restrições para o problema de controle.

De forma geral, pode-se ter três tipos de restrição em sistemas: restrição nos estados, restrição na variável de comando e restrição na variação da variável de comando. Separando esses três tipos de restrição, estamos considerando que não haverá restrição em uma variável que seja uma combinação linear entre estados e variáveis de comando do sistema.

Consideremos então o vetor y_c como aquele que expressa quais estados do sistema possuirão restrições a serem seguidas. Desta forma, podemos definir a matriz C_c que escolhe, a partir do vetor de estados, aqueles que serão restritos.

$$y_c = C_c x \tag{3.43}$$

Aqui então, pode-se definir os valores de restrição de y_c:

$$y_c^{min} \leq y_c[k+i] \leq y_c^{max} \tag{3.44}$$

Desenvolvendo a expressão acima, temos:

$$y_c^{min} \leq C_c x[k+i] \leq y_c^{max} \tag{3.45}$$

E então, substituindo $x[k+i]$ através da matriz de seleção Π, temos:

$$y_c^{min} \leq C_c \Pi_i^{(n,N)} \tilde{x}[k] \leq y_c^{max} \tag{3.46}$$

$$y_c^{min} \leq C_c \Pi_i^{(n,N)} \left(\Phi x[k] + \Psi \tilde{u}\right) \leq y_c^{max} \tag{3.47}$$

Expressando os elementos das matrizes desta última equação,

$$\begin{bmatrix} y_c^{min} \\ \vdots \\ y_c^{min} \end{bmatrix} \leq \begin{bmatrix} C_c \Pi_1^{(n,N)} \Phi \\ \vdots \\ C_c \Pi_N^{(n,N)} \Phi \end{bmatrix} x[k] + \begin{bmatrix} C_c \Pi_1^{(n,N)} \Psi \\ \vdots \\ C_c \Pi_N^{(n,N)} \Psi \end{bmatrix} \tilde{u} \leq \begin{bmatrix} y_c^{max} \\ \vdots \\ y_c^{max} \end{bmatrix} \tag{3.48}$$

Pode-se reduzir a inequação dupla acima a apenas uma inequação da seguinte forma:

$$\begin{bmatrix} +C_c\Psi_1 \\ \vdots \\ +C_c\Psi_N \\ -C_c\Psi_1 \\ \vdots \\ -C_c\Psi_N \end{bmatrix} \tilde{u} \leq \begin{bmatrix} -C_c\Phi_1 \\ \vdots \\ -C_c\Phi_N \\ +C_c\Phi_1 \\ \vdots \\ +C_c\Phi_N \end{bmatrix} x[k] + \begin{bmatrix} +y_c^{max} \\ \vdots \\ +y_c^{max} \\ -y_c^{min} \\ \vdots \\ -y_c^{min} \end{bmatrix} \tag{3.49}$$

A equação acima representa as restrições nas variáveis de estado. As restrições na variação das variáveis de comando podem ser expressas da seguinte forma:

$$\delta_{min} \leq u[k+i] - u[k+i-1] \leq \delta_{max} \tag{3.50}$$

Esse tipo de restrição é, muitas vezes, necessária, pois atuadores podem não conseguir variar de forma muito brusca a sua variável de saída. Expressando os elementos das matrizes da inequação acima, temos:

$$\begin{bmatrix}\delta_{min}\\ \vdots \\ \delta_{min}\end{bmatrix} \leq \begin{bmatrix}+I & O & O & \dots & O & O\\ -I & +I & O & \dots & O & O\\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots\\ O & O & O & \dots & -I & +I\end{bmatrix} \tilde{u} + \begin{bmatrix}-I\\ O\\ \vdots\\ O\end{bmatrix} u[k-1] \leq \begin{bmatrix}\delta_{max}\\ \vdots \\ \delta_{max}\end{bmatrix} \tag{3.51}$$

Aqui, I representa a matriz identidade, e O representa a matriz nula. Também esta inequação dupla pode ser simplificada para apenas uma inequação, da seguinte forma:

$$\begin{bmatrix}+I & O & O & \dots & O & O\\ -I & +I & O & \dots & O & O\\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots\\ O & O & O & \dots & -I & +I\\ -I & O & O & \dots & O & O\\ +I & -I & O & \dots & O & O\\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots\\ O & O & O & \dots & +I & -I\end{bmatrix} \tilde{u} \leq \begin{bmatrix}+I\\ O\\ \vdots\\ O\\ -I\\ O\\ \vdots\\ O\end{bmatrix} u[k-1] + \begin{bmatrix}+\delta_{max}\\ \vdots\\ +\delta_{max}\\ -\delta_{min}\\ \vdots\\ -\delta_{min}\end{bmatrix} \tag{3.52}$$

Por fim, pode-se representar as restrições na variável de comando da seguinte forma:

$$u_{min} \leq u[k+i-1] \leq u_{max} \tag{3.53}$$

Na equação acima, u_{min} e u_{max} são os valores de saturação da variável de comando. Esta inequação é obedecida para cada instante de tempo. Colocando-a na forma matricial para todos os instantes de tempo dentro do horizonte de predição, teremos:

$$\begin{bmatrix}u_{min}\\ \vdots \\ u_{min}\end{bmatrix} \leq \tilde{u} \leq \begin{bmatrix}u_{max}\\ \vdots \\ u_{max}\end{bmatrix} \tag{3.54}$$

Resumindo então todas as restrições, obtém-se o seguinte sistema de inequações:

$$A_{ineq}\tilde{u} \leq B_{ineq} \tag{3.55}$$

$$\tilde{u}_{min} \leq \tilde{u} \leq \tilde{u}_{max} \tag{3.56}$$

Aqui, definimos B_{ineq} da seguinte forma:

$$B_{ineq} = G_1 x[k] + G_2 u[k-1] + G_3 \tag{3.57}$$

As matrizes A_{ineq}, G_1, G_2 e G_3 apresentadas nas equações acima assumem os seguintes valores:

$$A_{ineq} = \begin{bmatrix} & & & +C_c\Psi_1 & & \\ & & & \vdots & & \\ & & & +C_c\Psi_N & & \\ & & & -C_c\Psi_1 & & \\ & & & \vdots & & \\ & & & -C_c\Psi_N & & \\ +I & O & O & \dots & O & O \\ -I & +I & O & \dots & O & O \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ O & O & O & \dots & -I & +I \\ -I & O & O & \dots & O & O \\ +I & -I & O & \dots & O & O \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ O & O & O & \dots & +I & -I \end{bmatrix} \tag{3.58}$$

$$G_1 = \begin{bmatrix} -C_c\Phi_1 \\ \vdots \\ -C_c\Phi_N \\ +C_c\Phi_1 \\ \vdots \\ +C_c\Phi_N \\ O \end{bmatrix} \quad G_2 = \begin{bmatrix} O \\ +I \\ O \\ \vdots \\ O \\ -I \\ O \\ \vdots \\ O \end{bmatrix} \quad G_3 = \begin{bmatrix} +y_c^{max} \\ \vdots \\ +y_c^{max} \\ -y_c^{min} \\ \vdots \\ -y_c^{min} \\ +\delta_{max} \\ \vdots \\ +\delta_{max} \\ -\delta_{min} \\ \vdots \\ -\delta_{min} \end{bmatrix} \tag{3.59}$$

O conjunto das equações (3.40), (3.55) e (3.56) formam um problema quadrático sujeito a restrições. Existem várias alternativas, em termos de software, para solucionar este problema de otimização e encontrar a melhor sequência de controle $\tilde{u}$ dentro do horizonte de predição que minimiza a função custo, obedecendo as restrições do problema.

Muitas vezes, a solução desse problema de otimização pode ser bastante custosa em termos computacionais, pois envolve operações com matrizes de ordem grande, como a matriz hessiana H, de ordem $N \times N$. Para diminuir esse custo, Alamir (2013) também desenvolve um método para diminuir o número de variáveis de decisão na função custo, o que reduziria bastante o tempo de otimização da função custo. Este método é chamado de parametrização. Para alguns sistemas, a redução do tempo de otimização é crucial, pois, caso a função custo não consiga calcular a melhor sequência de controle antes de um período de amostragem, a sequência de controle enviada para a planta não será a ótima, podendo modificar o

comportamento do sistema, e inclusive até comprometer sua estabilidade.

3.4 Regulador Quadrático Linear (LQR)

O Regulador Quadrático Linear (LQR) é um controlador considerado como ótimo. Ele calcula a melhor variável de comando possível para que os estados regulados atinjam a referência no menor tempo possível, sem considerar restrições para as variáveis do problema.

O LQR é considerado um caso particular de controle preditivo, no qual não se consideram as restrições do problema, grande vantagem do controle preditivo, e adota-se um horizonte de predição infinito. Além disso, considera-se, no desenvolvimento de sua lei de controle, que todos os estados da planta são conhecidos em todos os instantes de tempo. Sabe-se, entretanto, que nem sempre os estados são conhecidos em um sistema real. Muitas vezes, são necessários sensores muito caros ou muito complexos que não consigam ser implementados em um sistema. Assim, é necessário algum tipo de adaptação ao LQR. Por exemplo, há como aplicar um estimador de estados no sistema, tal como o Filtro de Kalman, para estimar o valor dos estados que não se consegue medir. Neste caso, o LQR passa a ser chamado Controle Linear Quadrático Gausssiano (LQG).

Além disso, para controlar o sistema, o LQR não restringe a magnitude da variável de comando enviada para a planta. No entanto, muitos atuadores, na realidade, não conseguem fornecer valores muito altos, o que os acaba saturando, e fazendo o LQR perder em desempenho.

Apesar de o LQR não levar em consideração restrições sobre a variável de comando, ele não deixa de ser uma boa técnica de controle para diversos sistemas, e frequentemente é utilizado por pesquisadores para comparar seu desempenho com o de outros controladores.

Para desenvolver a lei de controle do LQR, considera-se o seguinte sistema, no espaço de estados, a ser controlado:

$$\dot{x} = Ax + Bu \tag{3.60}$$

O algoritmo do LQR permite calcular a solução ótima para a variável de comando $u(t)$ por realimentação de estado, tal como mostra a equação abaixo.

$$u = -Kx \tag{3.61}$$

Nessa equação, K é o vetor de ganhos do LQR, e é esse vetor que gera a variável de comando ótima. Para calculá-lo, deve-se, similarmente ao que se faz em controle preditivo, achar o valor de K que minimiza uma função custo J. No caso do LQR, tal função custo é definida como a seguinte integral no tempo:

$$J = \int_0^\infty (x^T Q x + u^T R u) dt \tag{3.62}$$

Nessa equação, Q e R, matrizes de ponderação dos estados e da variável de comando, devem ser

matrizes positivo definidas.

Substituindo a Eq. (3.61) no espaço de estados do sistema, teremos:

$$\dot{x} = (A - BK)x \tag{3.63}$$

Agora, substituindo a Eq. (3.61) na expressão da função custo, teremos:

$$\begin{aligned} J &= \int_0^\infty (x^T Q x + x^T K^T R K x) dt \\ &= \int_0^\infty x^T (Q + K^T R K) x dt \end{aligned} \tag{3.64}$$

Para a resolução desta integral, define-se uma matriz positivo definida P, tal que obedeça à relação abaixo.

$$\begin{aligned} x^T (Q + K^T R K) x = -\frac{d}{dt}(x^T P x) &= -\dot{x}^T P x - x^T P \dot{x} \\ &= -x^T [(A - BK)^T P + P(A - BK)] x \end{aligned} \tag{3.65}$$

Para que a última equação seja verdadeira para qualquer vetor x, deve-se ter

$$(A - BK)^T P + P(A - BK) = -(Q + K^T R K) \tag{3.66}$$

Se a matriz $A - BK$ for estável, possuindo todos os seus auto-valores com parte real negativa, sempre existirá uma matriz P que satisfaça a equação acima (OGATA, 2010). Uma vez definida a matriz P, a função custo pode ser reescrita:

$$J = \int_0^\infty -\frac{d}{dt}(x^T P x) dt = -x^T(\infty) P x(\infty) + x^T(0) P x(0) \tag{3.67}$$

Como se supôs estabilidade do sistema, pode-se inferir que $x(\infty) = x^T(\infty) = 0$. Desta forma a função custo se reduz a

$$J = x^T(0) P x(0) \tag{3.68}$$

Deve-se então minimizar a função custo acima para achar a melhor solução para o sistema. Como R é uma matriz positivo definida, pode-se escrevê-la como $R = T^T T$. Desta forma, a Eq. (3.66) pode ser escrita da seguinte forma:

$$(A^T - K^T B^T) P + P(A - BK) + Q + K^T T^T T K = 0 \tag{3.69}$$

Aplicando algumas propriedades de álgebra matricial e reorganizando os elementos desta equação, tem-se:

$$A^T P + PA + [TK - (T^T)^{-1} B^T P]^T [TK - (T^T)^{-1} B^T P] - PBR^{-1} + Q = 0 \tag{3.70}$$

A minimização da função custo requer então a minimização da parcela dependente de K na equação acima. Sendo assim, deve-se ter:

$$TK - (T^T)^{-1} B^T P = 0 \tag{3.71}$$

Isolando a matriz de ganhos K em um dos lados da equação, tem-se:

$$K = T^{-1}(T^T)^{-1} B^T P = R^{-1} B^T P \tag{3.72}$$

Tem-se então uma expressão para a matriz de ganhos K que minimiza a função custo em função da matriz P, a qual pode ser determinada resolvendo-se a equação matricial reduzida de Riccati:

$$A^T P + PA - PBR^{-1} B^T P + Q = 0 \tag{3.73}$$

Percebe-se então que a solução para o LQR pode ser obtida analiticamente. Primeiro, deve-se resolver a equação matricial de Riccati, determinar a matriz P, e, através dela, na Eq. (3.72), determinar a matriz ótima de ganhos K que minimiza a função custo.

3.5 Arquitetura de validação de controladores - Plataforma HIL

No projeto de controladores, uma série de etapas deve ser cumprida antes de se implementar a lei de controle na planta física real. As etapas preliminares existem para que a probabilidade de falha do controlador, quando implementado em malha fechada com o processo físico real, seja reduzida, diminuindo assim também os custos no projeto, variável essencial quando se trata de problemas na engenharia.

Além disso, alguns tipos de sistema podem ameaçar a segurança dos operadores ao redor, se não forem bem controlados. Daí a importância de se fazer testes e simulações em ambientes seguros antes de implementar o controle no processo físico real.

As principais etapas para validação de controladores são 4: *Model in the Loop* (MIL), *Software in the Loop* (SIL), *Process in the Loop* (PIL) e *Hardware in the Loop* (HIL) (SILVA, 2017), sendo cada uma associada a uma função no processo de validação, como mostra a figura abaixo.

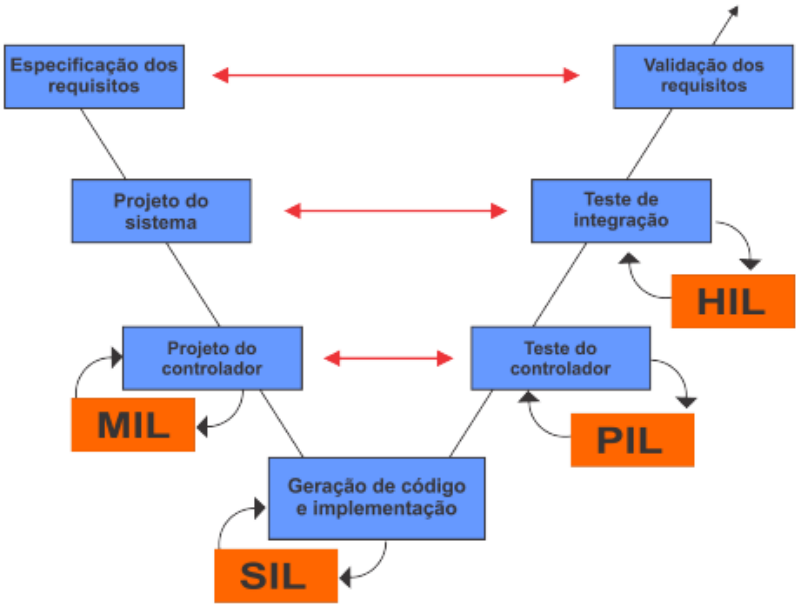


Figura 3.4: Etapas para validação de controladores
Fonte: (SILVA, 2017)

A seguir, serão detalhadas cada uma dessas etapas.

3.5.1 *Model in the loop* (MIL)

A primeira etapa no projeto de controladores é a chamada *Model in the Loop*, em que é feita uma simulação do comportamento do controlador em malha fechada com a planta, tudo em um ambiente computacional, ou seja, tanto o modelo da planta, quanto o controlador são implementados em *software*, como mostrado na Fig. (3.5). Uma ferramenta muito utilizada para simulações de MIL é o *Matlab/Simulink*, que possui recursos próprios para simulação de sistemas de controle.

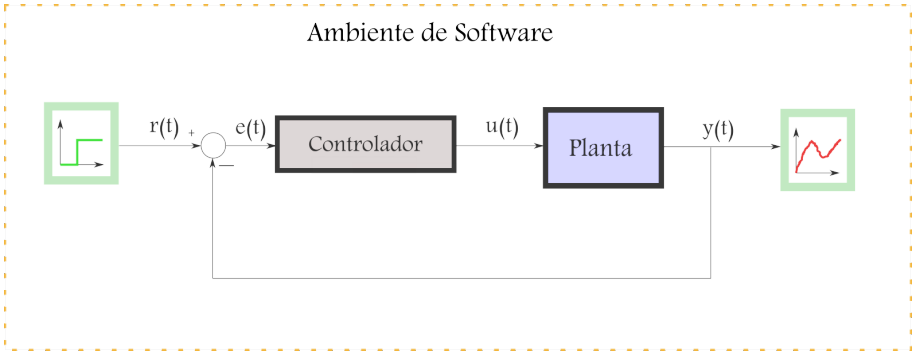


Figura 3.5: Esquema mostrando uma malha de controle em *Model in the Loop* (MIL)

A vantagem do teste em MIL é que se consegue ter uma boa ideia do comportamento do sistema, sem ter que utilizar componentes de *hardware* específicos. Algumas das informações do sistema que já podem

ser obtidas nesta etapa são a resposta do sistema a vários cenários, o tempo de execução do algoritmo, o respeito às restrições, além de poder ser realizada a sintonia adequada de parâmetros do controlador.

A realização desta etapa pode ainda identificar erros com o modelo do sistema utilizado, possibilitando ao projetista uma correção relativamente rápida e de custo pouco significante.

3.5.2 *Software in the loop* (SIL)

Uma vez testado em MIL, a etapa subsequente é a de simulação em *Software in the loop*. Nesta etapa, a planta do sistema continua sendo implementada por um modelo funcional em *Simulink*. A diferença da simulação em MIL para a simulação em SIL é que, nesta última, o algoritmo de controle é implementado em software, porém em código-fonte, utilizando linguagens de programação de nível mais baixo como *C* ou *C++*, por exemplo, podendo ser codificado manualmente ou gerado automaticamente com ferramentas do *Simulink*.

Na etapa de SIL, verifica-se a correspondência do código de controle com os resultados obtidos na etapa de MIL. Para verificar sucesso nesta etapa, espera-se obter resultados de simulação próximos aos que se obtiveram na etapa de MIL. Nesta etapa, o projetista pode perceber e corrigir erros de programação.

3.5.3 *Process in the loop* (PIL)

A etapa seguinte é a chamada *Process in the loop* (PIL). Nesta etapa, o algoritmo de controle já é implementado em um *hardware* externo, como a placa microcontroladora *Beaglebone*, por exemplo, e troca informações com a planta, a qual continua sendo simulada por blocos funcionais do *Simulink*.

A importância desta etapa é que se pode identificar problemas relacionados ao sistema embarcado, como por exemplo, memória disponível no *hardware*, sincronia de troca de dados, velocidade de processamento, ruído e perda de precisão na conversão analógico/digital, etc.

3.5.4 *Hardware in the loop* (HIL)

A próxima etapa é a simulação do sistema de controle em *Hardware in the loop* (HIL) (BAYAT, 2015), (VICENTE; RIBEIRO, 2014). Nesta etapa, implementa-se o algoritmo de controle em *hardware*, e o modelo da planta ainda em *software*, porém em um computador dedicado a rodar apenas o modelo, aproximando o sistema mais de um sistema real. A Fig. (3.6) mostra um esquema de sistema funcionando em HIL.

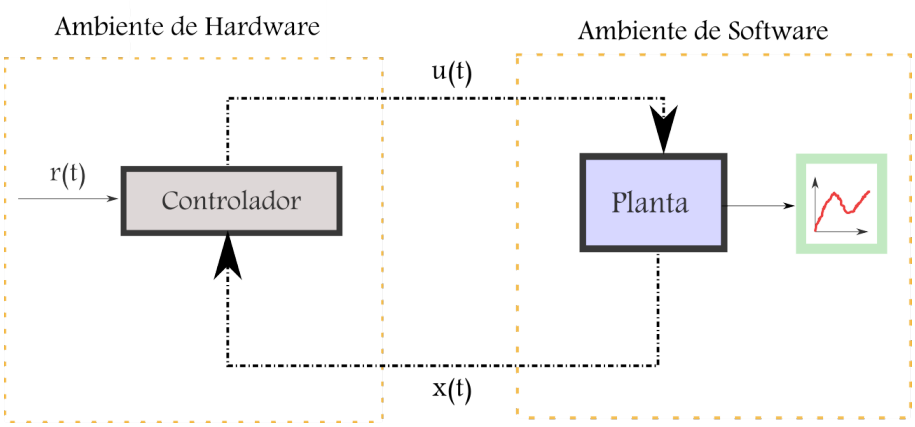


Figura 3.6: Esquema de malha de controle em *Hardware in the Loop* (HIL)

Esta etapa é a que mais se aproxima da realidade, e por isso sua grande importância. Nela pode-se verificar a interação entre os sistemas de *software* e *hardware* que interagem na malha de controle implementada. Implementada esta etapa, espera-se que o controlador esteja pronto para ser embarcado no sistema físico real, pois a arquitetura utilizada no HIL reproduz com alta confiabilidade o sistema modelado.

A divisão estruturada da validação de controladores nestas etapas (MIL, SIL, PIL e HIL) ajuda bastante a perceber possíveis limitação na malha de controle no decorrer do processo de validação, facilitando o trabalho do projetista na detecção e mitigação de erros.

O sistema simulado em HIL fica bem mais próximo do sistema real, pois o modelo da planta é implementado em um processador dedicado (chamado aqui de *Target*), em que não são executados processos paralelos. Desta forma, consegue-se ter uma melhor visualização do comportamento do sistema ao ser controlado.

Em um computador à parte, não mostrado na Fig. (3.6), chamado *Host*, o modelo do sistema a ser controlado é criado e carregado no *Target*. Uma vez carregado o modelo no *Target*, o *Host* é desconectado dele, deixando o sistema composto por *Target* e controlador em uma malha fechada de controle.

O *hardware* utilizado para implementar o controle pode ser escolhido de forma a atender requisitos específicos do projeto. Algumas placas controladoras encontradas no mercado são bastante comuns para aplicações a nível de laboratório, tais como Arduino, Raspberry, BeagleBone, Odroid, entre outros microcontroladores para sistemas embarcados.

As conexões entre o controlador e o *Target* devem ser, de preferência, conexões analógicas, para simular com maior fidelidade as conexões que aconteceriam no processo físico real. A utilização de conexões analógicas, porém, é limitada pelas restrições de cada controlador. Se o controlador só possuir 6 entradas analógicas, e o modelo possuir 8 estados, por exemplo, não será possível enviar todos os estados do *Target* para o controlador via conexão analógica. Para contornar esse problema, pode-se utilizar estimadores de estado ou conexões alternativas para envio de estados (do *Target* para o controlador) ou de comando (do controlador para o *Target*).

Neste trabalho, embora tenham sido realizadas todas as etapas propostas (MIL, SIL, PIL e HIL), será dada maior ênfase às simulações de MIL e HIL, pois estas representam os o início e o fim do processo de validação do controlador.

Capítulo 4

Metodologia

4.1 Introdução

Neste capítulo, é apresentada a técnica da parametrização utilizada para melhorar o tempo de execução do algoritmo do MPC, em particular a parametrização exponencial, a qual foi utilizada neste trabalho. Em seguida, são apresentadas as características da plataforma HIL utilizada para realizar as simulações, e por fim são apresentados os *solvers* que podem ser utilizados para resolver o problema quadrático existente no desenvolvimento do MPC.

4.2 Parametrização

Parametrização é uma técnica que reduz a quantidade de variáveis de decisão em um problema de otimização sem reduzir substancialmente o desempenho do sistema de controle (ALAMIR, 2013).

Para ilustrar o mecanismo da parametrização em um problema de controle preditivo, consideremos o sistema linear a seguir:

$$x[k+1] = Ax[k] + Bu[k] \tag{4.1}$$

Para este sistema, desejamos que a variável x atinja um valor desejado x_d e que a variável de comando $u[k]$ esteja restrita ao intervalo $[u_{min}, u_{max}]$. Para isso, será utilizado um horizonte de predição N.

Utilizando a formulação geral do MPC, esse problema cai na otimização de uma função custo com N variáveis, pois, dentro do horizonte de predição, a sequência ótima que minimiza a função custo é dada por:

$$\tilde{u}[k] = \begin{bmatrix} u[k] \\ u[k+1] \\ \vdots \\ u[k+N-1] \end{bmatrix} \in R^N \tag{4.2}$$

Consideremos então, a título de exemplo, a seguinte parametrização da sequência de comando, em que apenas dois graus de liberdade são utilizados, p_1 e p_2:

$$\begin{aligned} u[k+i-1] &= p_1, i = 1. \\ u[k+i-1] &= p_2, i \neq 1. \end{aligned} \tag{4.3}$$

Percebe-se que, desta forma, a sequência de comando $\tilde{u}[k]$ pode ser escrita em função dos parâmetros p_1 e p_2 da seguinte forma:

$$\tilde{u}[k] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ \vdots & \vdots \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} = \Pi_r \cdot p \tag{4.4}$$

Na equação acima, a matriz Π_r tem dimensão $N \times 2$.

Assim, pode-se reescrever as matrizes H e F em função da nova variável de decisão p.

$$J(p) = \frac{1}{2}\tilde{u}^T H \tilde{u} + F^T \tilde{u} = \frac{1}{2} p^T [{\Pi_r}^T H \Pi_r] p + [{\Pi_r}^T F]^T p \tag{4.5}$$

Fazendo $H_r = {\Pi_r}^T H \Pi_r$ e $F_r = {\Pi_r}^T F$, simplifica-se a função custo:

$$J(p) = \frac{1}{2} p^T H_r p + {F_r}^T p \tag{4.6}$$

Da mesma forma, pode-se simplificar as matrizes A_{ineq} e B_{ineq}, definidas no capítulo 3, para A_r e B_r, respectivamente:

$$A_r = \begin{bmatrix} A_{ineq}\Pi_r \\ -\Pi_r \\ +\Pi_r \end{bmatrix} \tag{4.7}$$

$$B_r = \begin{bmatrix} B_{ineq} \\ -\tilde{u}_{min} \\ +\tilde{u}_{max} \end{bmatrix} \tag{4.8}$$

O índice r nas novas matrizes referem-se ao fato de estas matrizes serem "reduzidas"devido à parametrização utilizada.

Neste exemplo, reduziu-se a quantidade de variáveis de decisão na função custo de N parâmetros para

apenas dois parâmetros, p_1 e p_2. Entretanto, o número de parâmetros, bem como a escolha da matriz Π_r não são fixos, podendo ser escolhidos de forma a atender melhor cada tipo de problema de controle em específico.

Na seção a seguir, será explicado um caso particular de parametrização que foi utilizado neste trabalho e também amplamente discutido por Alamir (2013): a parametrização exponencial.

4.2.1 Parametrização Exponencial

Existem diversas formas de parametrizar um problema de otimização. Neste trabalho, será focada a parametrização do tipo exponencial.

Considera-se aqui um sistema com n_u atuadores, cada um dos quais com seus respectivos tempos de assentamentos, representados pela variável τ_r. Alamir (2013) define a constante λ conforme a Eq. (4.9):

$$\lambda = \frac{3}{\tau_r} \tag{4.9}$$

Definindo o período de amostragem como τ_s, a variável de controle $u[k+i]$ pode ser parametrizada exponencialmente em função de novas variáveis de decisão p da seguinte forma:

$$u_j[k+i] = e^{-\lambda i \tau_s} p_1 + e^{-\lambda i \tau_s/(\alpha+1)} p_2 + e^{-\lambda i \tau_s/(2\alpha+1)} p_3 + e^{-\lambda i \tau_s/(3\alpha+1)} p_4 + \ldots, \alpha > 1 \tag{4.10}$$

Essa parametrização é feita para cada variável de comando u existente no sistema a ser controlado, de forma que o índice j na equação acima refere-se à ordem do atuador em questão. De maneira geral, considera-se que a quantidade de variáveis de comando (ou de atuadores) no sistema é igual a n_u. O número de exponenciais no somatório acima, n_e, define a quantidade de variáveis de decisão, n_p, na função custo parametrizada, de forma que:

$$n_p = \sum_{j=1}^{n_u} n_e^{(j)} \tag{4.11}$$

A matriz de seleção reduzida Π_r, como definida na Eq. (4.5), no contexto da parametrização exponencial, será chamada de Π_e. Esta matriz permite escrever a sequência de comandos $\tilde{u}$ em função das novas variáveis de decisão p de forma que:

$$\tilde{u}[k] = \Pi_e p[k] \tag{4.12}$$

Para definir os elementos da matriz Π_e, reescreve-se a Eq. (4.10) acima da seguinte forma:

$$u_j[k+i] = \sum_{l=1}^{n_e^{(j)}} [m_{j,l}(i)] p_l^{(j)} \tag{4.13}$$

Na equação acima, os termos $m_{j,l}(i)$ são as exponenciais que aparecem na Eq. (4.10). Colocando o somatório desta equação na forma matricial, obtém-se:

$$u_j[k+i] = [M_j(i)]p^{(j)} \tag{4.14}$$

Nesta equação, $p^{(j)}$ e $M_j(i)$ possuem as seguintes expressões:

$$p^{(j)} = \begin{bmatrix} p_1 & p_2 & \dots & p_{n_e} \end{bmatrix}^T \tag{4.15}$$

$$M_j(i) = \begin{bmatrix} m_{j,1}(i) & m_{j,2}(i) & \dots & m_{j,n_e}(i) \end{bmatrix} \tag{4.16}$$

Concatenando as variáveis de comando de todos os n_u atuadores, tem-se:

$$u[k+i] = \begin{bmatrix} M_1(i) & O & O & \dots & O \\ O & M_2(i) & O & \dots & O \\ O & O & M_3(i) & \dots & O \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ O & O & O & \dots & M_{n_u}(i) \end{bmatrix} \begin{bmatrix} p^{(1)} \\ \dots \\ p^{(n_u)} \end{bmatrix} = [M(i)]p \tag{4.17}$$

Finalmente, fazendo i variar de 0 até $N-1$, a matriz Π_e pode ser expressa como

$$\Pi_e = \begin{bmatrix} M(0) \\ M(1) \\ \vdots \\ M(N-1) \end{bmatrix} \tag{4.18}$$

Assim, agora a sequência de comando pode ser expressa em função das variáveis de decisão p na forma parametrizada.

$$\tilde{u} = \Pi_e p \tag{4.19}$$

A parametrização do MPC possui um importante papel, especialmente nas aplicações que possuem limitação de processamento. O ganho em custo computacional é bastante alto quando se utiliza a parametrização, pois o número de variáveis de decisão diminui bastante. A diferença entre tempos de cálculo para o MPC parametrizado e para o não-parametrizado é alta, podendo inclusive definir se o sistema estabilizará ou não.

Visto que este trabalho pretende embarcar o algoritmo do MPC em uma placa microcontroladora de baixo custo como a *Beaglebone*, é importante que seja implementada uma parametrização para que o custo de processamento não inviabilize o controle da planta.

4.3 Plataforma HIL utilizada

Nesta seção, será descrita a plataforma *Hardware in the Loop* (HIL) utilizada para realizar os experimentos deste trabalho.

A figura abaixo mostra um esquema geral da plataforma HIL utilizada, de um ponto de vista sistêmico.

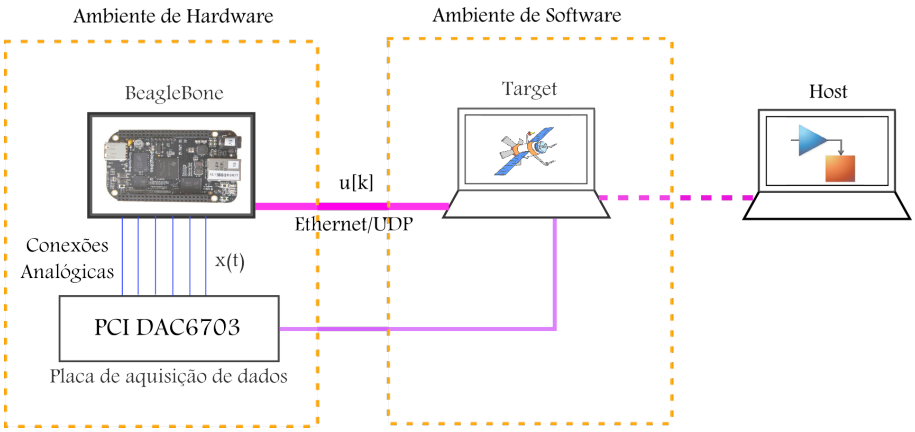


Figura 4.1: Plataforma HIL utilizada para as simulações deste trabalho

Nota-se, nessa arquitetura, a presença de dois elementos: o *Host* e o *Target*.

O *Target* é uma máquina dedicada para rodar em simulação o modelo da planta do sistema. É importante que esta seja uma máquina dedicada quase exclusivamente ao modelo, para que haja o máximo de similaridade com um processo real. Processos rodando em paralelo com o modelo poderiam interferir na dinâmica e no tempo de resposta do modelo. O *Target* está configurado para receber uma variável de comando e enviar seis variáveis de estado do sistema para o controlador.

O *Host*, por sua vez, é responsável por carregar o modelo no *Target*, possuindo, desta forma, função apenas temporária. Uma vez carregado o modelo do *Host* para o *Target*, aquele pode ser desconectado deste, deixando a malha de controle operar por si só. A transmissão do modelo do *Host* para o *Target* é feita via Ethernet e está representada pela linha rosa tracejada na figura.

A placa controladora utilizada em todas as simulações foi a *BeagleBone*. Essa é uma placa de baixo custo que possui uma série de canais de entrada e saída para comunicação com outros sistemas. Entre eles, pode-se citar canal USB, comunicação Ethernet, saída PWM (Modulação por Largura de Pulso), e vários canais de entrada analógica. Seu processador é o AM3358 Cortex-A8, de 1,0 GHz, memória de 512 MB DDR3, e possui suporte a diversos sistemas operacionais, tais como Debian, Android e Ubuntu (COLEY, 2013).

Optou-se então por utilizar canais de entrada analógica da *BeagleBone* para receber os estados, $x(t)$ provenientes do modelo, de forma paralela. Os valores dos estados são recebidos por uma placa de aquisição de dados, a qual faz a interface entre o modelo e o controlador. A transmissão dos estados está representada na Fig. (4.1) pelos fios azuis.

É importante ressaltar que os canais de entrada da *BeagleBone* apenas suportam tensões de 0 V a 1,8

V. Desta forma, faz-se necessário um escalonamento prévio dos valores dos estados. Posteriormente, os valores são reconvertidos aos valores originais, dentro da *BeagleBone*. Tal escalonamento é feito de forma linear como mostra a equação abaixo.

$$x_v = 1,8 \left(\frac{x_t - x_{min}}{x_{max} - x_{min}} \right) \quad V \tag{4.20}$$

Nesta equação, x_t é o valor do estado, x_{max} e x_{min} são os valores máximo e mínimo, respectivamente, que o estado atinge na simulação em *software*, e x_v é o valor do estado convertido para a escala de 0 V a 1,8 V.

Uma vez que os valores dos estados estão disponíveis e convertidos na *BeagleBone*, o código embarcado é responsável por calcular o valor da variável de comando, $u[k]$, que será enviada para controlar a planta.

O código embarcado na *Beaglebone* foi desenvolvido na linguagem *C++* de programação, devido à facilidade em se trabalhar com operações entre matrizes, utilizando as bibliotecas adequadas.

Há que se fazer um comentário aqui sobre a técnica de otimização utilizada para achar a variável de comando ótima. Existem diversos algoritmos que resolvem o problema quadrático mostrado nas equações (3.40), (3.55) e (3.56). Para exemplificar, alguns dos métodos existentes mais conhecidos são: qpOASES, Expansão do Gradiente, KKT e quadprog/*Matlab* (ALAMIR, 2013). Neste trabalho, utilizou-se a função quadprog, do *Matlab*, nas simulações MIL, e o algoritmo qpOASES para as simulações em HIL.

O envio da variável de comando para a planta é feito pelo protocolo UDP via cabo Ethernet. Esta conexão está representada pelo cabo rosa contínuo na Fig. (4.1). A transmissão via UDP é uma boa alternativa para este tipo de aplicação devido à velocidade de transmissão de dados demandada. Apesar de o protocolo UDP não garantir a recepção de todos os pacotes de dados enviados, ele é bastante simples de ser implementado e satisfaz a rapidez de transmissão necessária. Este tipo de conexão não é o ideal, já que, num sistema com satélite real, não se utilizaria conexão via UDP/Ethernet, mas sim envio de dados analógicos. O envio da variável de comando por conexão analógica não foi implementada devido a limitações da placa.

O *software* utilizado para criar o modelo do satélite no *Host* foi o *Matlab/Simulink*, visto que o *Simulink* possui um ambiente de gerenciamento de sistemas de controle, através de blocos funcionais, bastante intuitivo.

O *Simulink* possui ainda uma ferramenta específica para aplicações em tempo real, como é o caso do presente trabalho. Essa ferramenta é o *Simulink Real-Time*. É então com o suporte do *Simulink Real-Time* que se carrega o modelo para o *Target* e se monitora o comportamento do sistema quando se põe a funcionar a malha de controle.

Há ainda que se comentar sobre a sincronia de transmissão de dados entre a *BeagleBone* e o *Target*. A plataforma implementada, apesar de simular com bastante proximidade a malha de controle em um processo real, não funciona em tempo real. Isso acontece porque a placa controladora utilizada, *BeagleBone*, não possui temporizador sincronizado com o *Target*. A consequência de o sistema não trabalhar em tempo real é que não há a garantia de que, no instante de tempo em que a variável de comando for enviada para o

Target, este ainda terá o mesmo estado que foi utilizado para calcular aquela variável de comando.

Essa falta de sincronia, a qual pode se manifestar tanto por atraso quanto por antecipação no envio do comando para o *Target*, pode resultar num comportamento não esperado, ou até indesejado, da dinâmica do sistema, como, por exemplo, a desestabilização do mesmo.

Por este motivo, faz-se necessário um ajuste manual dessa sincronia temporal. Esse ajuste foi feito a nível de software no código implementado na *BeagleBone*, da seguinte forma: mede-se o tempo utilizado para a *BeagleBone* calcular a variável de comando $u[k]$, e este tempo é comparado com o período de amostragem utilizado para discretizar a planta. Se o tempo medido for menor que o período de amostragem, é introduzido um atraso antes de enviar a variável de comando para o modelo. Por outro lado, caso o tempo de cálculo seja superior ao período de amostragem, ao ser atingido o período de amostragem, para-se o algoritmo, enviando a variável de comando calculada até aquele instante pelo processo de otimização, ainda que ela não seja a ótima, para que o sincronismo seja obedecido. O algoritmo qpOASES utilizado na plataforma possui recursos para que isso seja implementado facilmente, através da inserção de um parâmetro na chamada do algoritmo.

4.4 Solvers

Um dos problemas ao implementar o algoritmo do MPC é a resolução do problema quadrático sujeito a restrições, representado pelas equações (3.40), (3.55) e (3.56), onde se precisa encontrar a sequência de controle que minimiza a função custo.

Este não é um problema simples de resolver analiticamente, porém existem algoritmos de otimização que conseguem resolvê-lo computacionalmente de forma satisfatória. A esses algoritmos damos aqui o nome de *solvers*. Alguns exemplos de *solvers* são: quadprog/*Matlab*, Expansão do gradiente, KKT e qpOASES. Cada *solver* possui suas próprias particularidades, que devem ser consideradas ao escolher um adequado para o problema de controle em questão. Abaixo serão detalhados cada um desses *solvers*.

4.4.1 Quadprog

O quadprog é um *solver* desenvolvido na plataforma *Matlab* de fácil utilização e que pode ser utilizado de forma satisfatória em simulações de MIL. É uma função que retorna um vetor u que minimiza uma função $J(u)$ (Eq. (4.21)) sujeita às condições mostradas nas Eq. (4.22).

$$J(u) = \frac{1}{2}u^T H u + F^T u \tag{4.21}$$

$$\begin{aligned} A_{ineq}u &\leq B_{ineq} \\ A_{eq}u &= B_{eq} \\ u_{min} \leq u &\leq u_{max} \end{aligned}$$

(4.22)

As matrizes H, F, A_{ineq} e B_{ineq} são as matrizes já definidas no capítulo 3. É importante que a matriz H seja positivo-definida para que o problema tenha solução. Já as matrizes A_{eq} e B_{eq} fazem parte de uma condição a mais que pode ser resolvida pelo quadprog. Desta forma, esta função pode ser utilizada, de modo simples, com a seguinte linha de comando no *Matlab*:

$$[u, J] = \textbf{quadprog}(H, F, A_{ineq}, B_{ineq}, A_{eq}, B_{eq}, u_{min}, u_{max}) \tag{4.23}$$

A função quadprog pode então retornar tanto o valor de u que minimiza a função custo, quanto o próprio valor mínimo dela, J.

Outros argumentos podem ser inseridos opcionalmente na função quadprog, tais como o tipo de algoritmo utilizado na otimização, inserção de tela de diagnósticos, número máximo de iterações permitidas, entre outros. Entretanto, é suficiente, para o problema proposto neste trabalho, inserir como argumentos as matrizes H, F, A_{ineq} e B_{ineq}.

O quadprog mostrou-se uma ferramenta simples e eficiente para as simulações realizadas em MIL, pois já possui as rotinas desenvolvidas, devendo o projetista apenas utilizar a função na linha de comando.

4.4.2 Expansão do Gradiente

Este *solver* é proposto por Alamir (2013) e é apresentado como uma alternativa a *solvers* que possuem dependência de bibliotecas numéricas externas. Ele é então um algoritmo *self-contained*, pois independe de bibliotecas externas e pode ser desenvolvido pelo próprio projetista.

O método consiste em expandir uma função custo $J(u)$ convexa, pela fórmula de Taylor, em torno de um valor inicial u_0, como na equação abaixo.

$$J(u) = J(u_0) + (u - u_0)\frac{\partial J}{\partial u}(u_0) + \frac{1}{2}(u - u_0)^2\frac{\partial^2 J}{\partial u^2}(u_0) + ... \tag{4.24}$$

A partir daí, através de um processo iterativo, são obtidos valores de u que geram valores cada vez menores na função $J(u)$. Desta forma, em algum momento, atinge-se o ponto de mínimo da função.

Propõe-se que o próximo valor, u_1, seja calculado através da inclinação da função em u_0, como mostrado na equação abaixo.

$$u_1 = u_0 - \alpha\frac{\partial J}{\partial u}(u_0) \tag{4.25}$$

A escolha da constante α deve ser feita de tal forma que nem se escolha uma constante alta demais, que faça a próxima iteração atingir um valor mais alto na função custo, e nem se escolha uma constante baixa demais, que demore muito para atingir o mínimo da função. Alamir (2013) demonstra que um valor que satisfaz essas condições é dado pelo inverso da norma da matriz hessiana H, como mostra a equação

abaixo.

$$\alpha = \frac{1}{||H||} \tag{4.26}$$

Nesta equação, $||H||$ é calculada como a raiz quadrada da soma dos quadrados dos elementos da matriz H.

Um aperfeiçoamento deste método pode ser obtido ponderando a constante α, de forma que sejam tomados passos maiores no sentido de atingir o mínimo da função mais rapidamente.

Apesar de este método ter a vantagem de ser auto-suficiente, independendo de bibliotecas externas e dando ao projetista mais autonomia sobre seu controlador, este *solver* realiza um número grande de iterações, podendo deixar o desempenho da rotina de controle inviável.

4.4.3 KKT

Este *solver* baseia-se nas condições necessárias para a optimalidade de Karush-Kuhn-Tucker, mais conhecidas como condições KKT (ALAMIR, 2013).

Alamir (2013) desenvolve este algoritmo também de forma auto-suficiente, assim como o método expansão do gradiente, dando uma alternativa ao projetista para que ele tenha liberdade de desenvolver seu *solver* sem se limitar a bibliotecas externas.

As condições de optimalidade KKT ditam que, se existe uma solução local u_0 que minimiza uma função custo $J(u)$ sob a restrição $g(u) \leq 0$, onde $g \in R^{n_g}$, então existe um vetor $\mu \in R^{n_g}$ que satisfaz as condições abaixo.

$$\begin{aligned} \frac{\partial J}{\partial u}(u_0) + \sum_{i=1}^{n_g} \mu_i \frac{\partial g_i}{\partial u}(u_0) &= 0 \\ \mu_i \cdot g_i(u_0) &= 0, i \in \{1, \ldots, n_g\}. \\ g_i(u_0) &\leq 0, i \in \{1, \ldots, n_g\}. \\ \mu_i &\geq 0, i \in \{1, \ldots, n_g\}. \end{aligned} \tag{4.27}$$

Definindo então a variável $z = [p \quad \mu]^T$, este *solver* realiza a minimização da função custo $J(z)$ abaixo.

$$J(z) = \frac{1}{2} F^T(z) F(z) \tag{4.28}$$

Nesta equação, a função $F(z)$ é dada pela matriz abaixo.

$$F(z) = \begin{bmatrix} \frac{\partial J}{\partial u}(u) + \sum_{i=1}^{n_g} \mu_i \frac{\partial g_i}{\partial u}(u) \\ 2(s_1 \cdot g_1 + (s_1 - 1)\mu_1) \\ \vdots \\ 2(s_{n_g} \cdot g_{n_g} + (s_{n_g} - 1)\mu_{n_g}) \end{bmatrix} \tag{4.29}$$

Na expressão de $F(z)$, a função s_i assume a expressão binária abaixo, para todo i no conjunto $\{1, \ldots, n_g\}$.

$$\begin{aligned} s_i = 1, g_i(u) + \mu_i \geq 0 \\ s_i = 0, g_i(u) + \mu_i < 0 \end{aligned} \tag{4.30}$$

Este método possui vantagem semelhante ao método de expansão do gradiente, ao se mostrar um algoritmo *self-contained*. Segundo Alamir (2013), ambos os métodos são alternativas para não se utilizarem métodos mais robustos que utilizam bibliotecas numéricas externas. Apesar de ambos os métodos terem sido testados, o *solver* utilizado para as simulações de HIL neste trabalho foi o qpOASES, detalhado a seguir.

4.4.4 qpOASES

O qpOASES é um código livre que é escrito na linguagem *C++*, e que oferece uma solução para o problema quadrático, sendo assim uma ferramenta muito útil em problemas de controle preditivo. Para seu funcionamento, algumas bibliotecas padrão em linguagem *C* são necessárias (FERREAU, 2014).

Por ser escrito em uma linguagem orientada a objeto, para utilizar este *solver*, cria-se instâncias a partir de uma classe chamada QProblem. Uma vez criada a instância, pode-se chamar suas funções para resolver o problema quadrático. Os argumentos inseridos são basicamente os mesmos envolvidos na função quadprog do *Matlab*. Abaixo é mostrado um exemplo de utilização da função *init* que inicializa as estruturas de dados do problema.

$$returnValue \quad \textbf{init}(H, F, A_{ineq}, u_{min}, u_{max}, 0, 0, nWSR, cputime); \tag{4.31}$$

Neste exemplo, são passados como argumentos as matrizes H, F, A_{ineq} e os valores mínimo e máximo da variável de otimização. Adicionalmente, passa-se a variável $nWSR$, que representa o número de novos cálculos realizados, e a variável $cputime$, opcional, que representa o máximo tempo permitido para realizar a rotina. Se ambas as constantes, $nWSR$ e $cputime$, forem não-nulas, a rotina pára quando alguma das duas constantes for atingida primeiro. Por fim, esta função retorna ainda o status que indica se o problema foi inicializado com sucesso.

Ferreau (2014) mostra que o qpOASES possui também diversos recursos para resolver problemas quadráticos em vários contextos, como exemplo:

- Pode resolver problemas quadráticos com matrizes constantes ou variáveis;
- Resolve também problemas quadráticos com funções não-convexas;
- Consegue ler dados e também escrevê-los em arquivos externos;
- Possui interface para ser utilizado com outros *softwares*, tais como Simulink/*Matlab*, Scilab, xPC Target, etc.

O qpOASES foi utilizado neste trabalho, nas simulações em HIL, devido a seu bom desempenho para o problema proposto, além de poder ser implementado em placas controladoras de baixo custo, como a *BeagleBone*, a qual fez parte da plataforma HIL utilizada neste projeto.

Capítulo 5

Resultados

5.1 Introdução

Neste capítulo, serão mostrados primeiramente as constantes numéricas utilizadas nas equações da dinâmica do satélite, bem como os parâmetros utilizados nos algoritmos de controle implementados neste trabalho.

Em seguida, serão mostrados os resultados obtidos no desenvolvimento deste trabalho. As simulações realizadas neste trabalho podem ser divididas em dois principais grupos:

- Simulações com controlador LQR
- Simulaçoes com controlador MPC

Tanto para o controlador LQR, quanto para o controlador MPC, foram realizadas simulações em *Model in the Loop* (MIL) e em *Hardware in the Loop* (HIL), utilizando, como modelo para a planta, os modelos linear e não-linear para o satélite rígido-flexível, desenvolvidos no capítulo 3 deste trabalho.

Para as simulações com o MPC, foram realizadas simulações com dois horizontes de predição (N = 20 e N = 60). Nas simulações com o LQR, foram realizadas simulações para o LQR com e sem saturação na variável de comando. A figura abaixo ilustra todos os cenários das simulações realizadas.

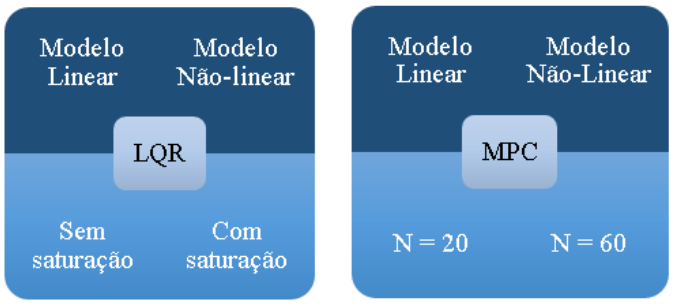


Figura 5.1: Cenários de simulação

As simulações serão mostradas na forma de gráficos do comportamento do satélite, em termos da deflexão rígida, da deflexão da ponta da haste e do torque imposto ao satélite como variável de comando.

5.2 Parâmetros Utilizados

Nesta seção, são mostrados os parâmetros utilizados para desenvolver o modelo do satélite rígido-flexível e também os parâmetros de sintonia utilizados nos algoritmos de controle do LQR e do MPC.

Para utilizar as equações do modelo desenvolvido para o satélite rígido-flexível, foram definidos os valores de algumas constantes referentes ao modelo, tais como comprimento da haste flexível, momento de inércia do rotor, entre outros. Os valores utilizados para o modelo de satélite, retirados de Gu *et al.* (2005), estão mostrados na tabela abaixo.

Tabela 5.1: Parâmetros do modelo de satélite rígido-flexível

Variável	**Símbolo**	**Valor**
Comprimento da haste	L	1,5 m
Raio do rotor	R	0,05 m
Componente de atrito viscoso	b_m	0,15 m^2/s
Momento de inércia do rotor	J_r	0,3 kg.m^2
Densidade linear da haste	μ	0,54 kg/m
Coeficiente de amortecimento	K_e	0,03
Rigidez da haste	EI	18,4 N.m^2
Massa de prova	m_L	0,25 kg
Momento de inércia da massa de prova	J_L	0,04 kg.m^2

Fonte: (GU; PETKOV; KONSTANTINOV, 2005)

Os algoritmos dos controladores LQR e MPC também possuem parâmetros de sintonia que serão mostrados nas tabelas a seguir. Entre tais parâmetros estão as matrizes de ponderação, valores das restrições impostas no caso do MPC, frequências de amostragem, entre outros.

Os parâmetros relacionados ao algoritmo do LQR estão mostrados na tabela abaixo.

Tabela 5.2: Parâmetros de sintonia do LQR

Variável	**Símbolo**	**Valor**
Matriz de ponderação de estados	Q	diag$\begin{bmatrix}100 & 1 & 1 & 1 & 1 & 1\end{bmatrix}$
Matriz de ponderação de comando	R	$\begin{bmatrix}0,1\end{bmatrix}$
Saturação na variável de comando	$[u_{min}, u_{max}]$	[-2,+2] N.m
Frequência de amostragem	f_s	50 Hz
Período de amostragem	T_s	20 ms

Na tabela acima, utilizou-se a notação diag$\begin{bmatrix}100 & 1 & 1 & 1 & 1 & 1\end{bmatrix}$ para representar a matriz quadrada diagonal de ordem 6, em que os elementos da diagonal principal são aqueles mostrados. Ainda na tabela,

é mostrado o valor escolhido para saturação na variável de comando para os cenários em que é utilizado o LQR com saturação.

Os parâmetros utilizados no algoritmo do MPC estão mostrados na tabela abaixo.

Tabela 5.3: Parâmetros de sintonia do MPC

Variável	**Símbolo**	**Valor**
Restrições na deflexão da haste	$[y_c^{min}, y_c^{max}]$	[-5,+5] cm
Restrições na variável de comando	$[u_{min}, u_{max}]$	[-2,+2] N.m
Horizonte de Predição	N	20/60
Período de amostragem	T_s	20 ms
Parâmetro da parametrização	α	10
Número de exponenciais	n_e	2
Tempo de acomodação	τ_r	0,1 s
Matriz de ponderação das variáveis reguladas	Q_y	$\boxed{100000}$
Matriz de ponderação das variáveis de comando	Q_u	$\boxed{0,1}$

O período de amostragem utilizado nas simulações, tanto para o LQR quanto para o MPC, foi escolhido no valor de 20 ms. Para escolher esse valor, realizou-se simulações no *Matlab* para diversos cenários com tempos de amostragem diferentes, tendo em vista que, quanto menor o tempo de amostragem, maior a frequência de amostragem, tornando o sistema o mais próximo possível de um processo físico real. Percebeu-se, no entanto, que, a partir de 20 ms, ao se diminuir ainda mais o período de amostragem, o sistema começava a divergir, sendo este valor o mais baixo que ainda conseguiu fazer o sistema estabilizar.

Para todas as simulações realizadas, foi utilizada uma referência do tipo degrau em $45°$ para a variável deflexão rígida θ, representada por uma linha tracejada verde nos gráficos.

5.3 Resultados com LQR

Nesta seção, serão mostrados os resultados de simulação obtidos utilizando-se o LQR como controlador para o sistema. Os resultados compreendem simulações com os modelos linear e não-linear, simulados em MIL e em HIL.

Além dos cenários citados acima, foram simulados dois cenários para o LQR: um com o LQR comum, e outro com LQR apresentando saturação na variável de comando. Este trabalho referir-se-á a este último cenário como "LQR saturado".

Como nas simulações com o MPC são envolvidas restrições para a variável de comando (τ) e para a deflexão da ponta da haste (w), como mostradas na tabela 5.3, também nos gráficos das simulações com LQR foram inseridas linhas tracejadas na cor vermelha representando os valores dessas restrições, para se conseguir comparar o desempenho do LQR com o MPC em termos de violação de restrições.

5.3.1 Resultados de Simulação para o LQR em MIL

5.3.1.1 Simulações com o modelo linear

As simulações em MIL realizadas para o sistema linear do satélite, utilizando o LQR como controlador, apresentaram os resultados mostrados nos gráficos abaixo.

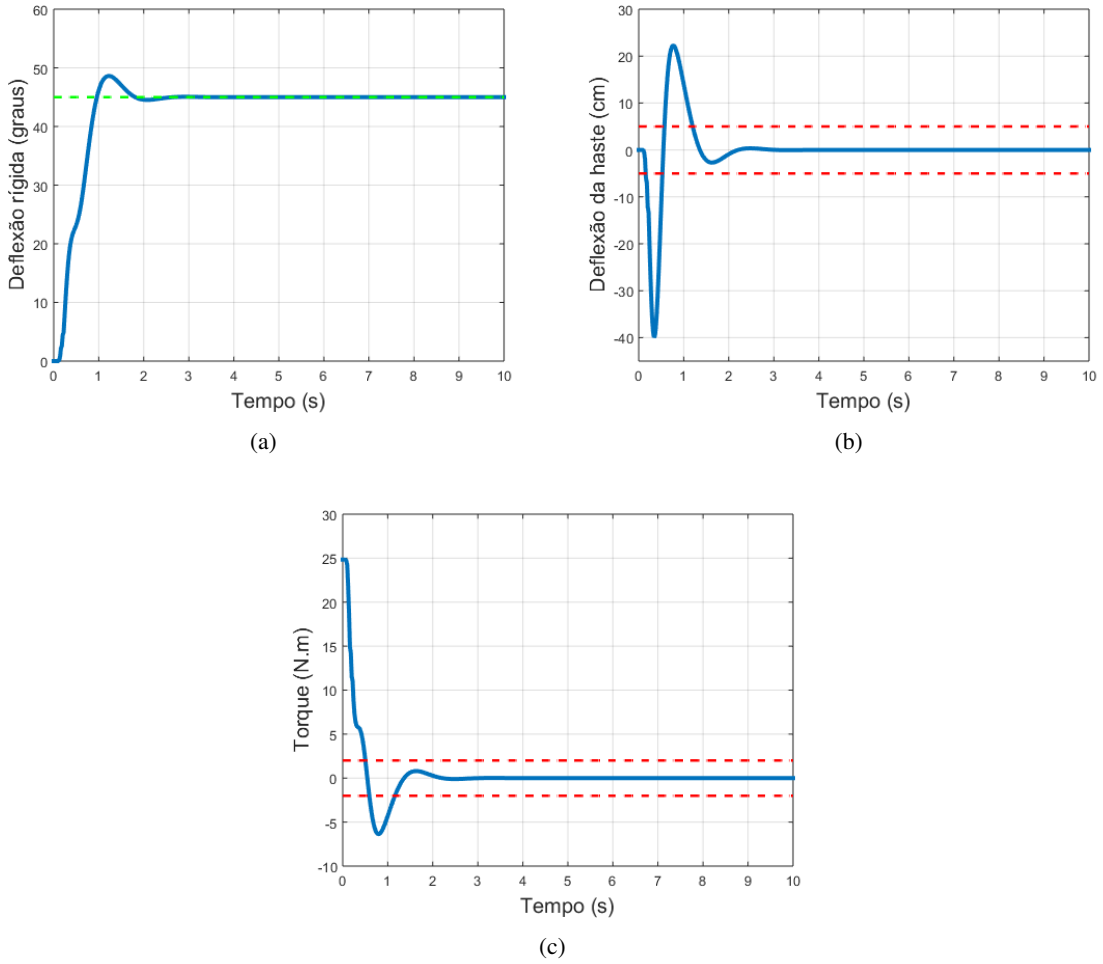


Figura 5.2: Comportamento do satélite utilizando o controlador LQR para o modelo linear em MIL

Nos gráficos mostrados na figura acima, a linha tracejada em verde representa a referência a ser seguida e as linhas tracejadas em vermelho representam os valores de restrição que foram impostas às simulações com o MPC (ver Tabela (5.3)), a termo de comparação.

Abaixo são mostrados os gráficos obtidos do comportamento do sistema linear do satélite ao ser controlado pelo LQR com saturação na variável de comando.

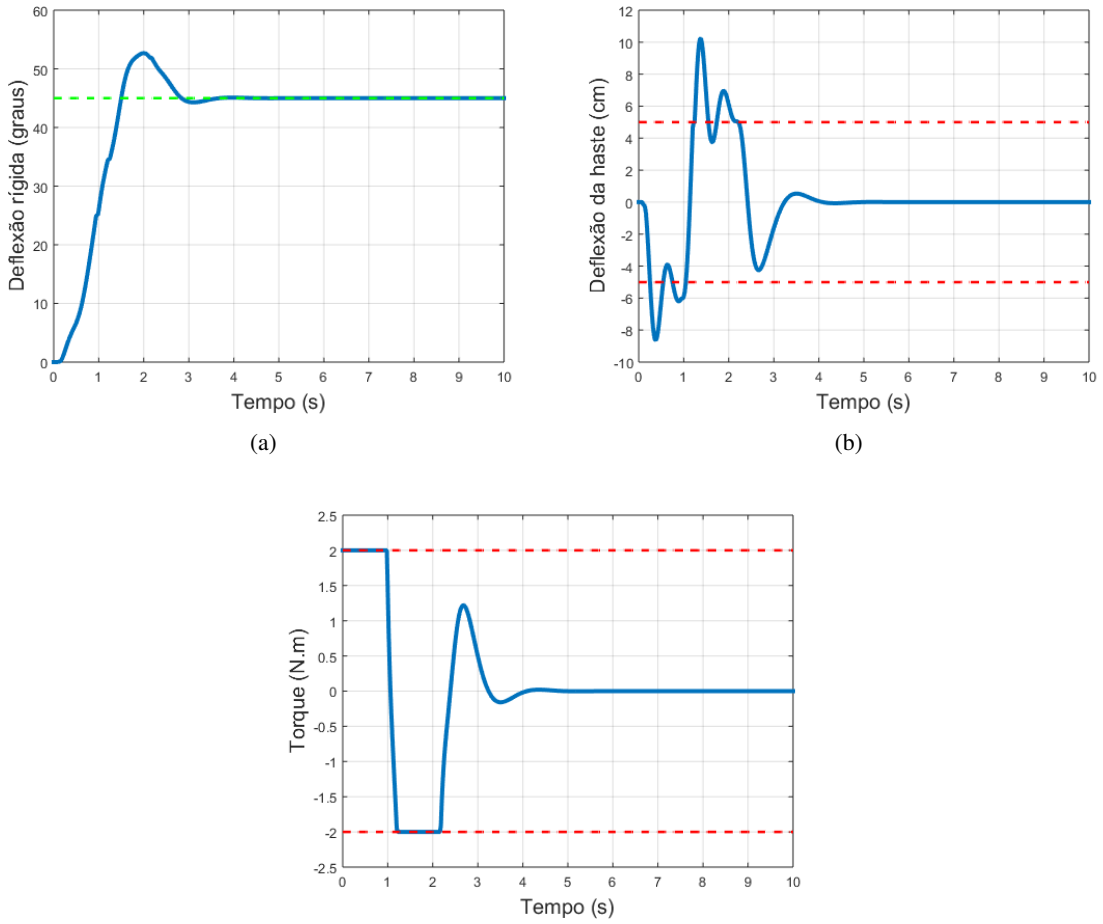


Figura 5.3: Comportamento do satélite utilizando o controlador LQR saturado para o modelo linear em MIL

Neste cenário, o sobressinal atingido foi de 17%, com um tempo de assentamento de 2,54 s. Nessas condições, a haste realizou uma deflexão máxima de 10,2 cm.

5.3.1.2 Simulações com o modelo não-linear

Ao se utilizar o modelo não-linear do satélite, as simulações mostraram os resultados mostrados nos gráficos a seguir.

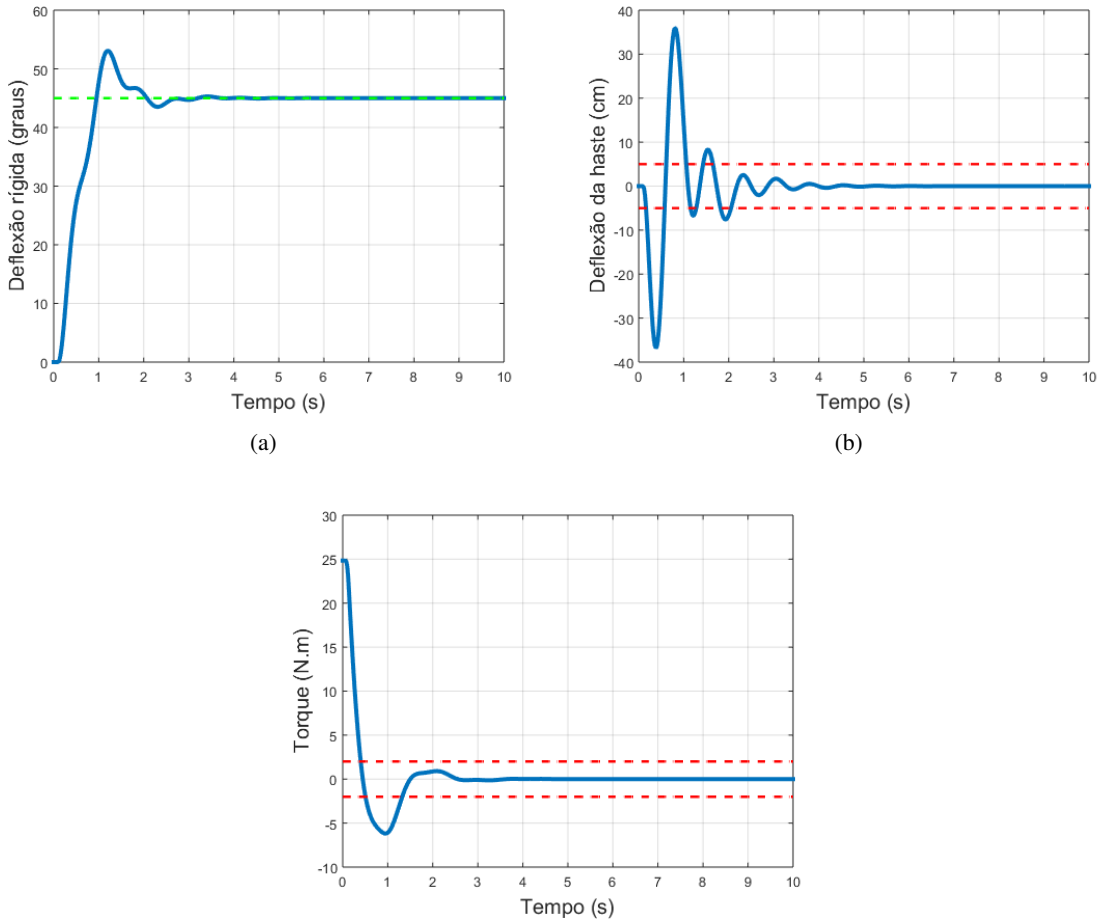


Figura 5.4: Comportamento do satélite utilizando o controlador LQR para o modelo não-linear em MIL

Como a variável de comando (torque) atingiu intensidade muito alta tanto para o modelo linear, quanto para o não-linear, e, na prática, valores tão altos podem não ser disponíveis para controlar um sistema, simulou-se também uma saturação para o torque de entrada do sistema. O valor de saturação foi estipulado em $2N.m$.

Aplicando a saturação na variável de comando para o modelo não-linear do satélite, obteve-se os seguintes gráficos.

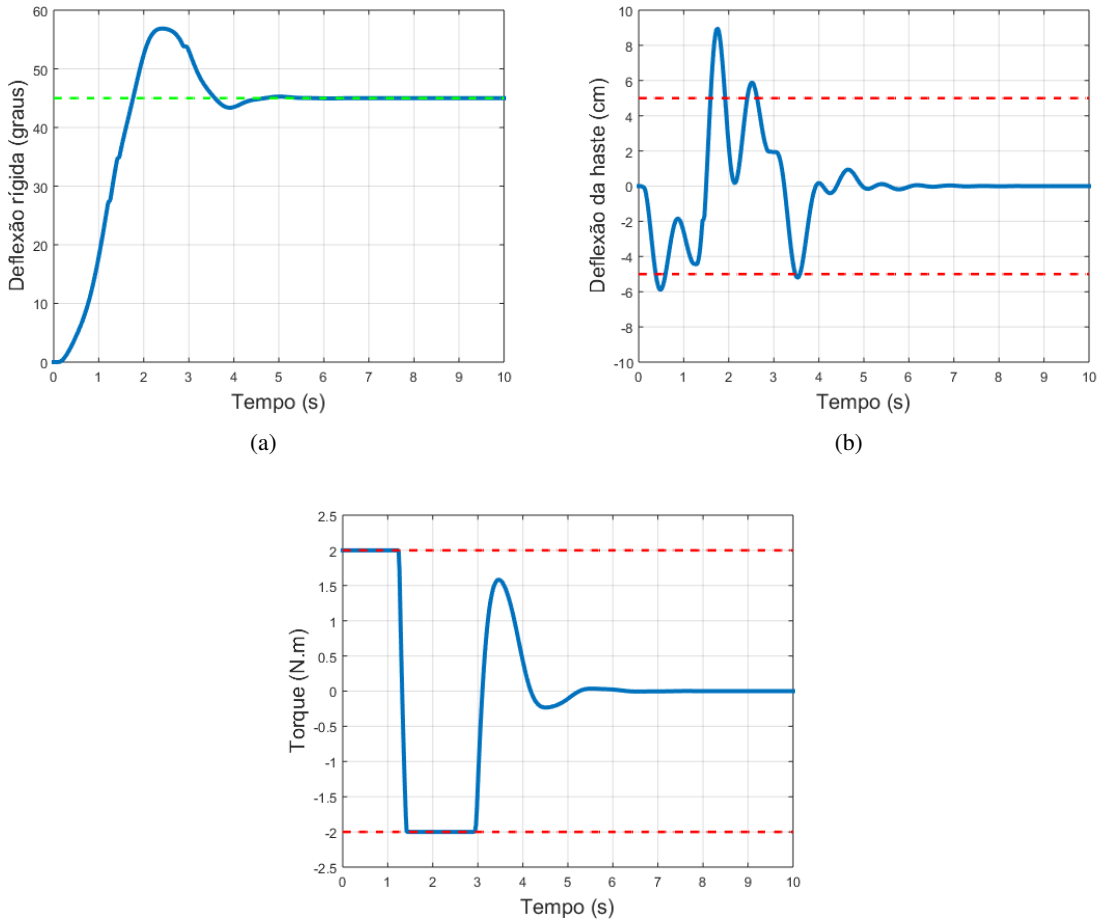


Figura 5.5: Comportamento do satélite utilizando o controlador LQR saturado para o modelo não-linear em MIL

Observa-se que, neste caso, foi atingido um sobressinal de $26,3\%$ com um tempo de assentamento de 4,18 s. A deflexão máxima da ponta da haste foi de 8,9 cm.

5.3.2 Resultados de Simulação para o LQR em HIL

O modelo foi então embarcado na plataforma HIL com o controlador LQR fechando a malha.

As matrizes do sistema linear do satélite, utilizadas no algoritmo do LQR, foram discretizadas com período de amostragem de 20 ms, enquanto que o período de amostragem do *Target*, onde o modelo é simulado, foi fixado em 200 μs, o mínimo aceitável para que o modelo seja simulado o mais próximo possível de um modelo contínuo.

5.3.2.1 Simulações com o modelo linear

Abaixo seguem os gráficos de comportamento do sistema para o modelo linear do satélite.

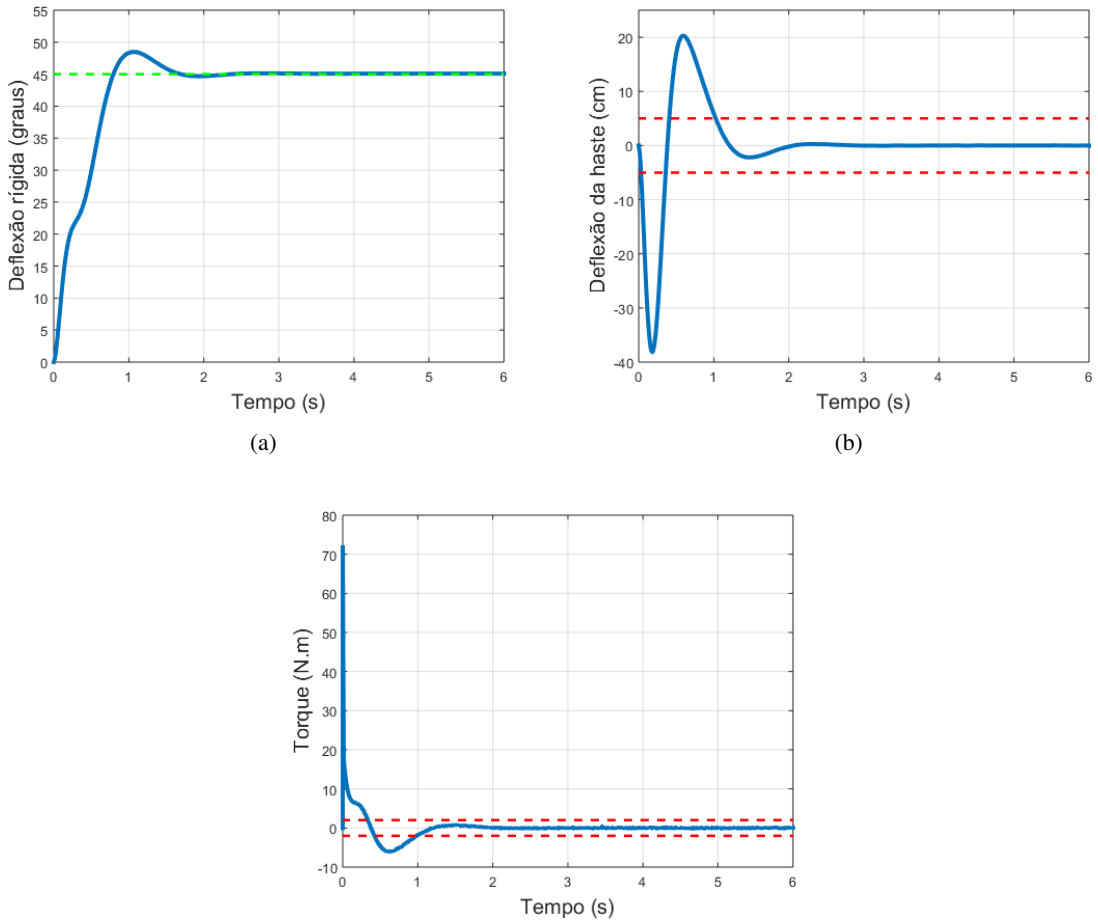
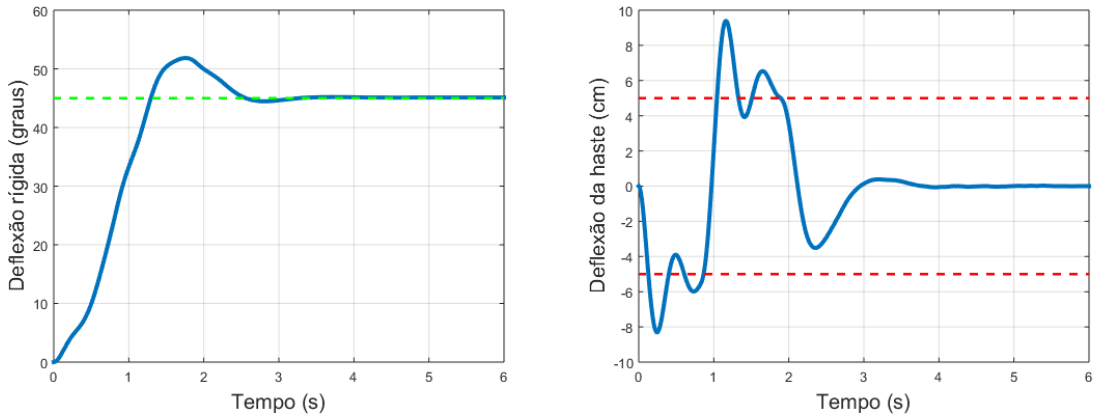


Figura 5.6: Comportamento do satélite utilizando o controlador LQR para o modelo linear em HIL

Percebe-se que houve estabilização do satélite após 1,49 s (tempo de assentamento 2%), gerando um sobressinal de $7,7\%$. Em compensação, a ponta da haste chega a oscilar com deflexão de 38,2 cm.

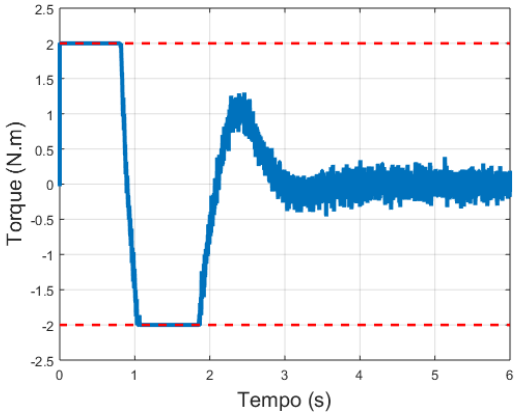
Percebe-se também nos gráficos que o torque demandado pelo satélite é bem alto no início da manobra, zerando após a estabilidade do sistema.

Abaixo seguem gráficos de simulação do modelo linear do satélite, em HIL, para o LQR saturado na variável de comando.



(a)

(b)



(c)

Figura 5.7: Comportamento do satélite utilizando o controlador LQR saturado para o modelo linear em HIL

Percebe-se que, aplicada a saturação na variável de comando τ, o sistema passa a se estabilizar de forma mais lenta, por volta de 2,42 segundos, apresentando também um sobressinal maior, da ordem de $15,2\%$. Nota-se também que a saturação gerou uma diminuição considerável na amplitude de oscilação da haste, que passou a oscilar com amplitudes de 9,4 cm.

5.3.2.2 Simulações com o modelo não-linear

O sistema de controle foi então simulado novamente, desta vez com o modelo não-linear do satélite.

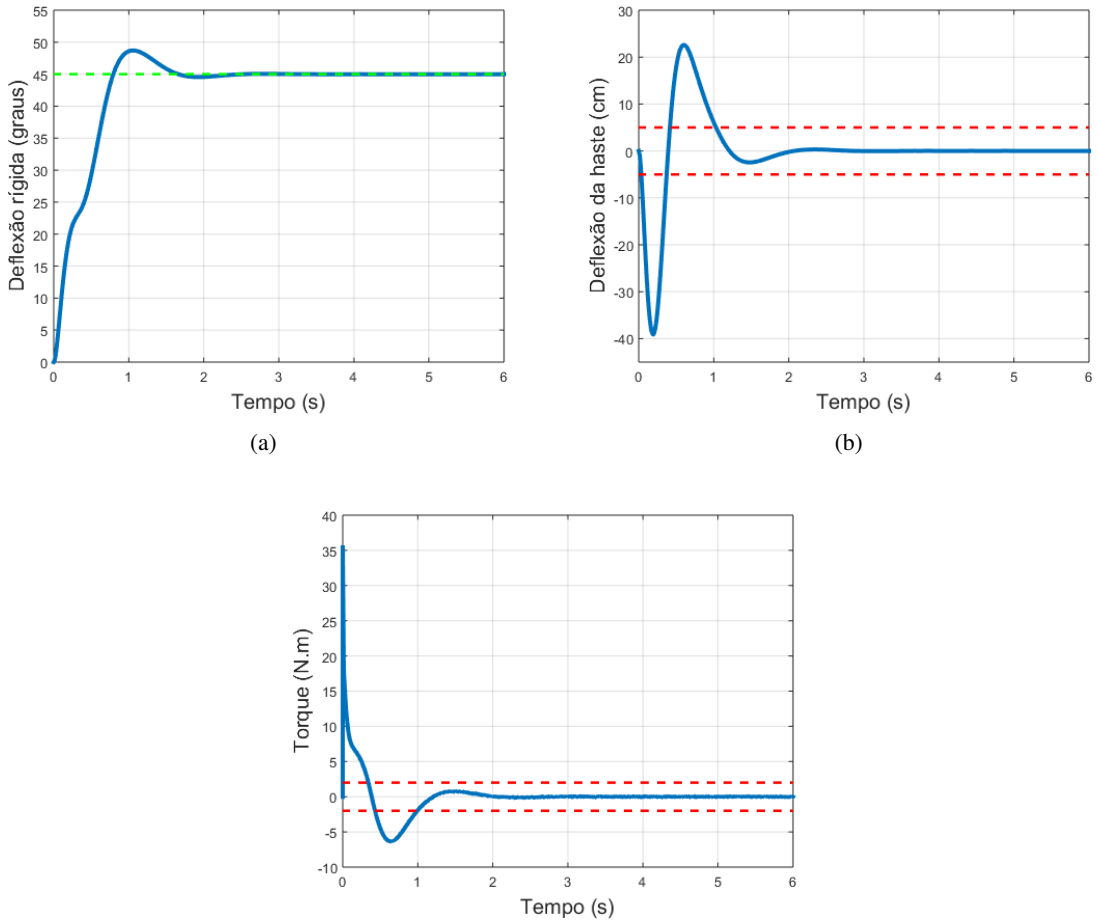


Figura 5.8: Comportamento do satélite utilizando o controlador LQR para o modelo não-linear em HIL

Por fim, simulou-se também o LQR com saturação na variável de comando para o modelo não-linear. Os resultados obtidos são mostrados nos gráficos abaixo.

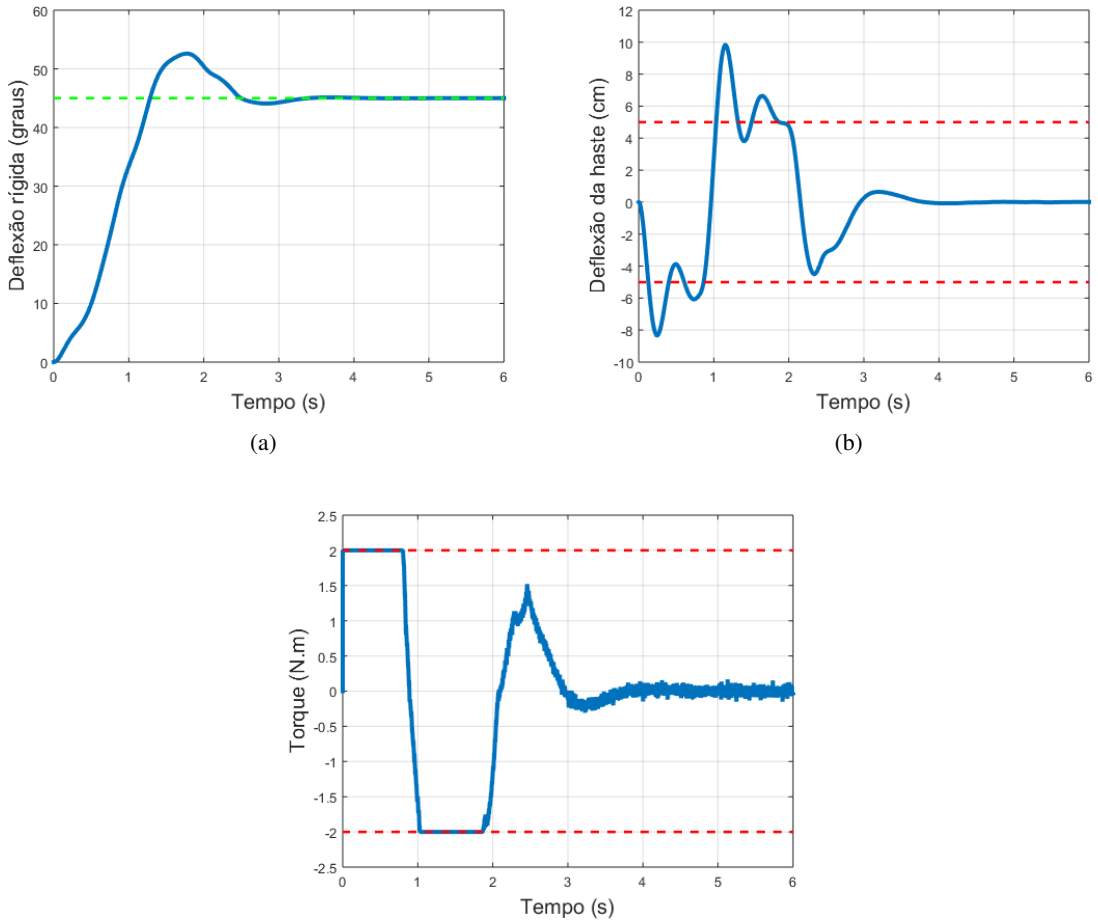


Figura 5.9: Comportamento do satélite utilizando o controlador LQR saturado para o modelo não-linear em HIL

Como pode-se perceber nos gráficos, o comportamento do sistema não-linear novamente assemelhou-se bastante com o do modelo linear.

Visto que, mesmo com a saturação no torque de entrada do sistema, a haste flexível ainda oscilou com amplitude considerável, simulou-se então uma malha de controle com o controlador MPC, o qual lida bem com restrições nas variáveis do sistema.

A seção a seguir mostra os resultados obtidos com o controlador MPC.

5.4 Resultados com MPC

Nesta seção, serão mostrados os resultados de simulação utilizando o MPC como controlador para o sistema.

Definiu-se como restrição para a deflexão da haste um valor limite de 5 cm de deflexão em sua ponta, para qualquer sentido de giro. Além disso, impôs-se também a restrição no torque de entrada em 2 N.m,

assim como feito para o LQR saturado.

5.4.1 Resultados de Simulação para o MPC em MIL

5.4.1.1 SImulações com o modelo linear

Os gráficos a seguir mostram os resultados obtidos para a simulação do sistema linear do satélite em MIL utilizando o MPC como controlador para o sistema, com horizonte de predição 20.

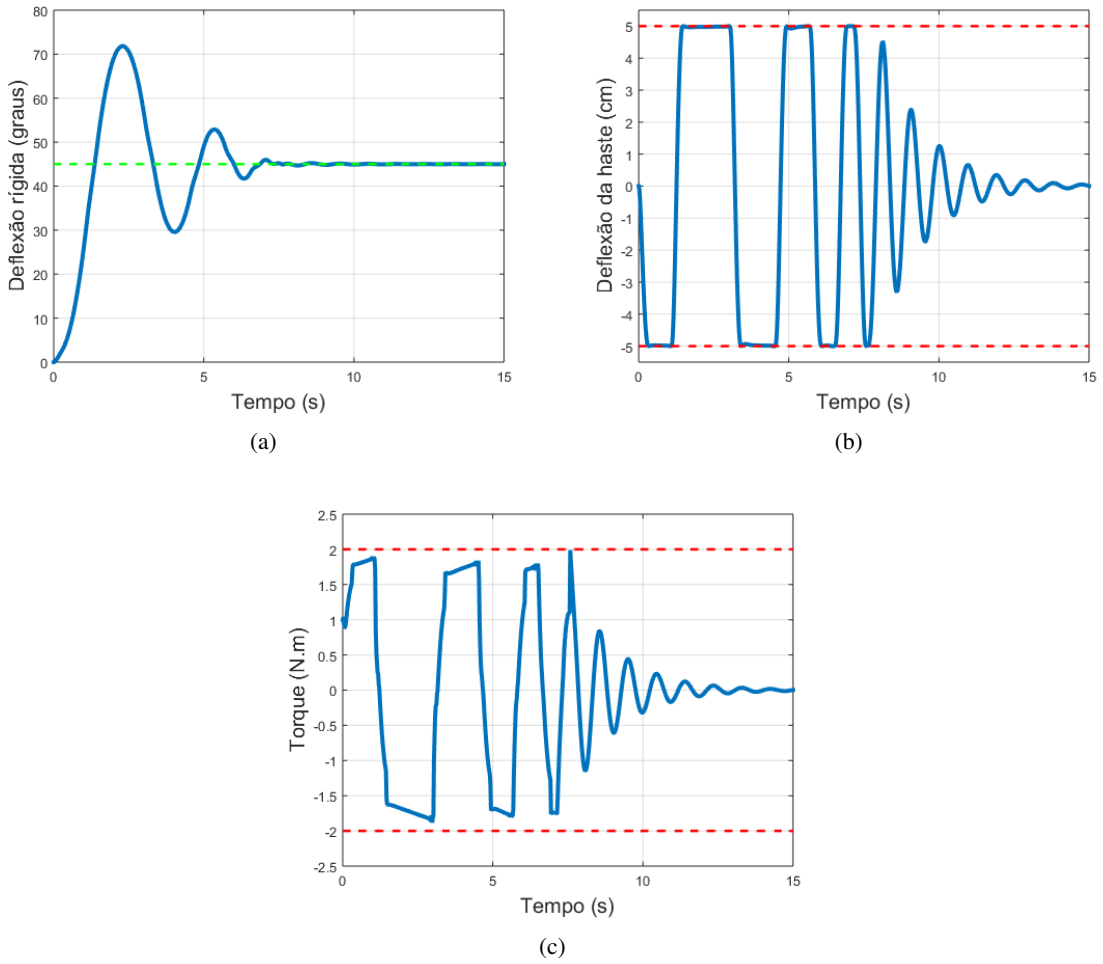


Figura 5.10: Comportamento do satélite utilizando o controlador MPC para o modelo linear em MIL, com um horizonte de predição de 20

Percebe-se destes gráficos que o controlador MPC consegue manter a deflexão da ponta da haste e o torque dentro das faixas pré-determinadas, marcadas com a linha tracejada vermelha.

O horizonte de predição foi alterado para 60, e o comportamento do sistema nessas condições está mostrado nos gráficos das figuras a seguir.

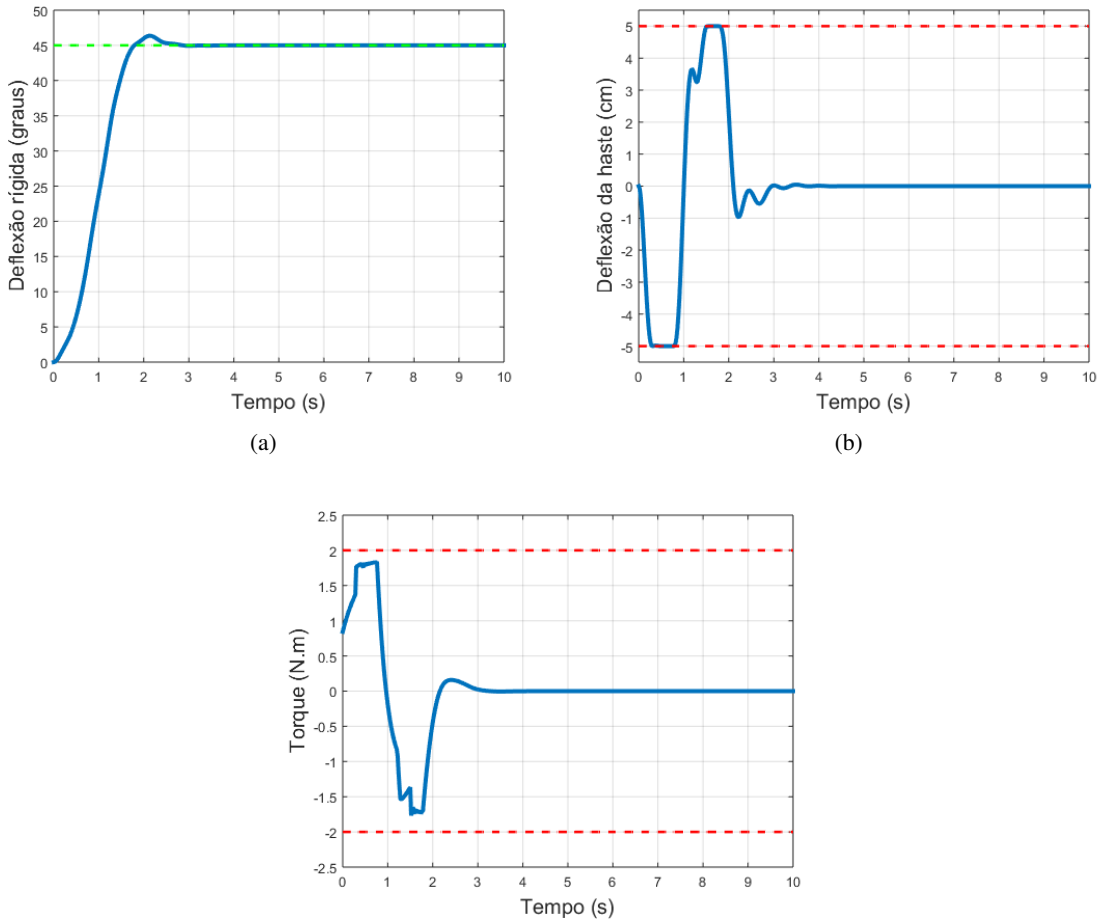
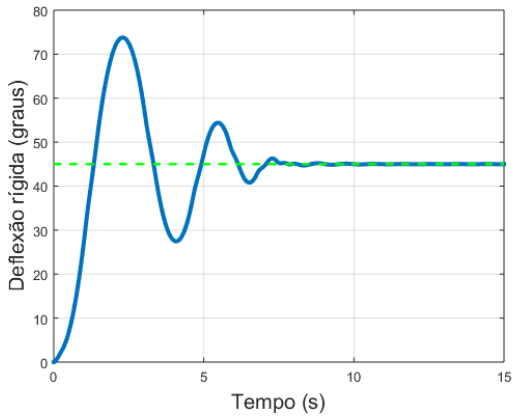


Figura 5.11: Comportamento do satélite utilizando o controlador MPC para o modelo linear em MIL, com um horizonte de predição de 60

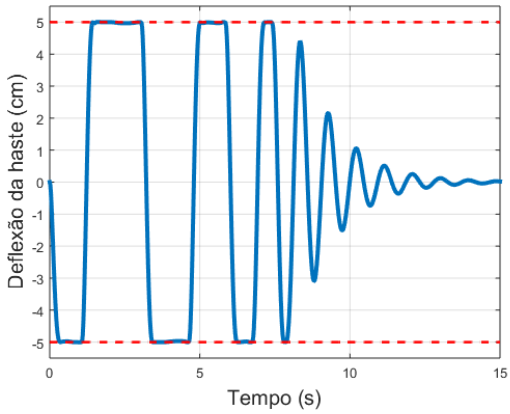
Percebe-se que o efeito de aumentar o horizonte de predição é a diminuição do sobressinal (de $59,6\%$ para 3%) e do tempo de assentamento 2% (de 7,12 s para 2,3 s).

5.4.1.2 Simulações com o modelo não-linear

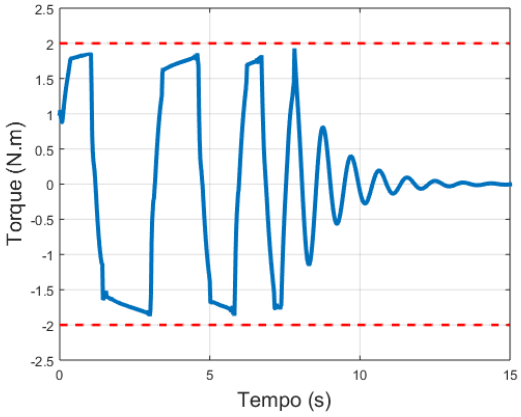
Realizando as simulações com o modelo não-linear do satélite, foram obtidos os gráficos a seguir, para um horizonte de predição de 20.



(a)



(b)



(c)

Figura 5.12: Comportamento do satélite utilizando o controlador MPC para o modelo não-linear em MIL, com um horizonte de predição de 20

Por fim, alterando o horizonte de predição para 60, ainda com o modelo não-linear, obteve-se os seguintes gráficos.

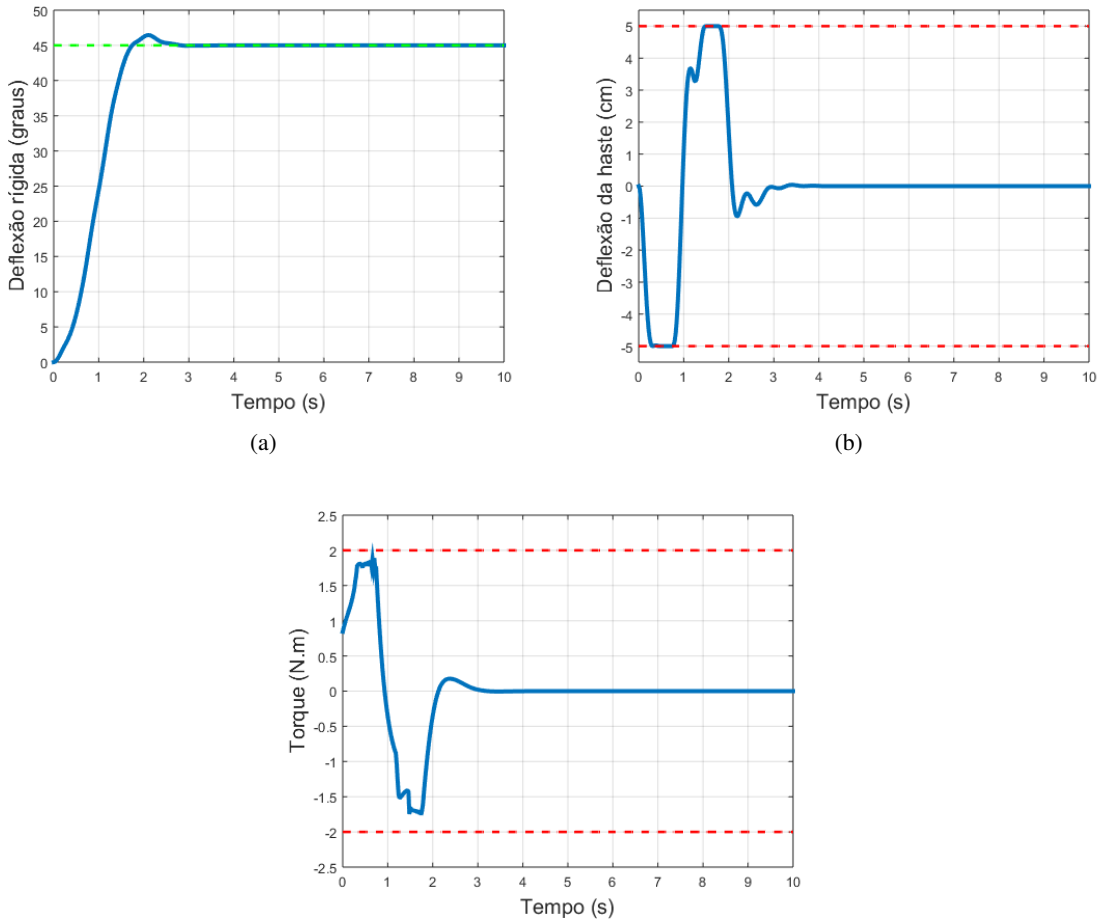


Figura 5.13: Comportamento do satélite utilizando o controlador MPC para o modelo não-linear em MIL, com um horizonte de predição de 60

Observa-se que, também para o modelo não-linear, o aumento do horizonte de predição favorece os índices sobressinal (diminui de $63,9\%$ para $3,2\%$) e tempo de assentamento 2% (diminui de 7,4 s para 2,28 s).

5.4.2 Resultados de Simulação para o MPC em HIL

A seguir, serão mostrados os resultados obtidos em HIL para o modelo trabalhando com o controlador MPC.

É importante lembrar que as matrizes do sistema linear do satélite, utilizadas no algoritmo do MPC, foram discretizadas com período de amostragem de 20 ms, enquanto que o período de amostragem do *Target*, onde o modelo é simulado, foi fixado em 200 μs, o mínimo aceitável para que o modelo seja simulado o mais próximo possível de um modelo contínuo.

5.4.2.1 Simulações com o modelo linear

Primeiramente, simulou-se o modelo linear com um horizonte de predição de 20 passos. Os gráficos abaixo mostram o comportamento da deflexão rígida, da deflexão da ponta da haste e da variável de comando para esse cenário.

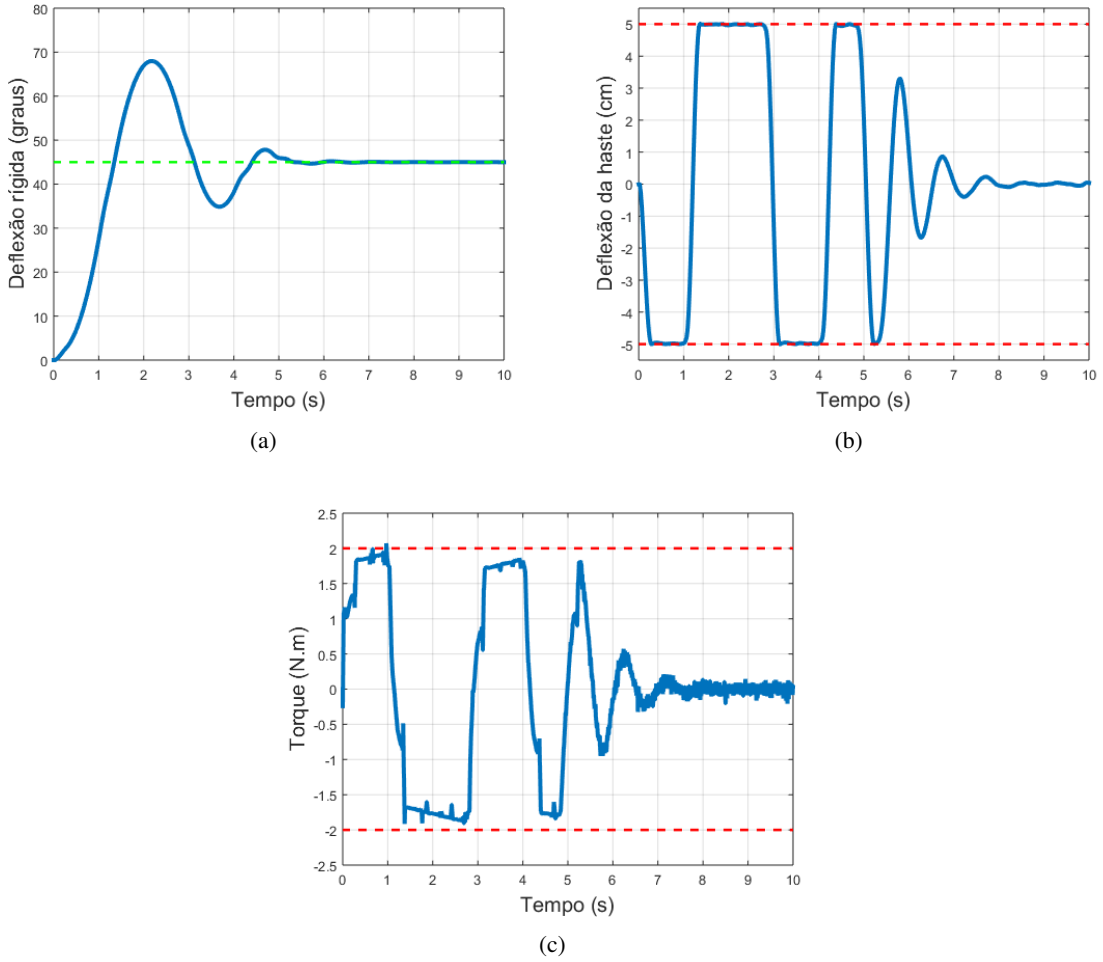


Figura 5.14: Comportamento do satélite utilizando o controlador MPC para o modelo linear em HIL, com um horizonte de predição de 20

Percebe-se que o sistema se estabiliza em 5 segundos, apresentando um sobressinal de 51%. Em contrapartida, tanto a deflexão da haste como a variável de comando foram mantidas dentro de seus limites, como desejado.

Simulou-se então a malha de controle com horizonte de predição de 60 passos, para tentar melhorar o sobressinal e o tempo de assentamento. Os resultados são mostrados abaixo.

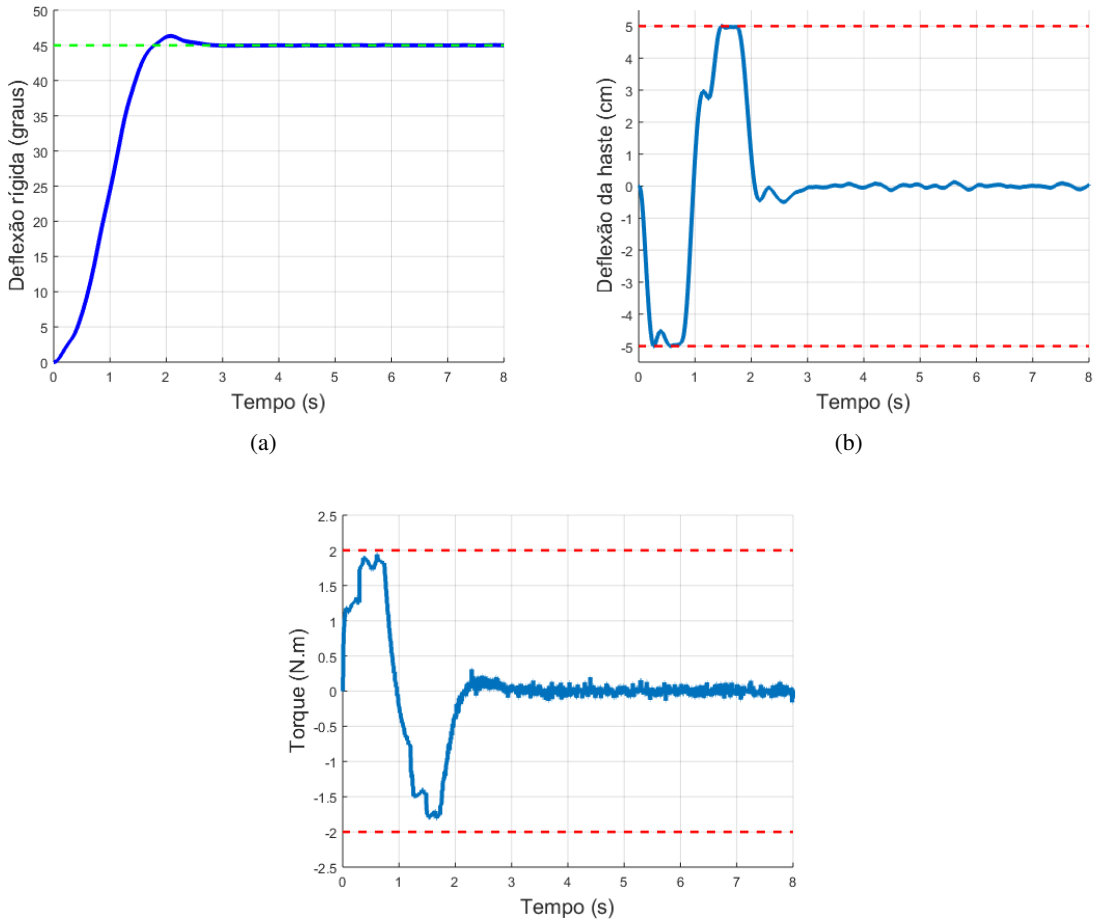


Figura 5.15: Comportamento do satélite utilizando o controlador MPC para o modelo linear em HIL, com um horizonte de predição de 60

Com o aumento do horizonte de predição, conseguiu-se diminuir consideravelmente o tempo de assentamento do sistema, o qual nesse cenário é de 2,23 s. Além disso, nota-se que o sobressinal também foi reduzido, para $2,9\%$.

5.4.2.2 Simulações com o modelo não-linear

O modelo não-linear do satélite também foi utilizado nos mesmos cenários que o modelo linear, para horizontes de predição iguais a 20 e 60 passos. Os gráficos abaixo mostram os resultados obtidos.

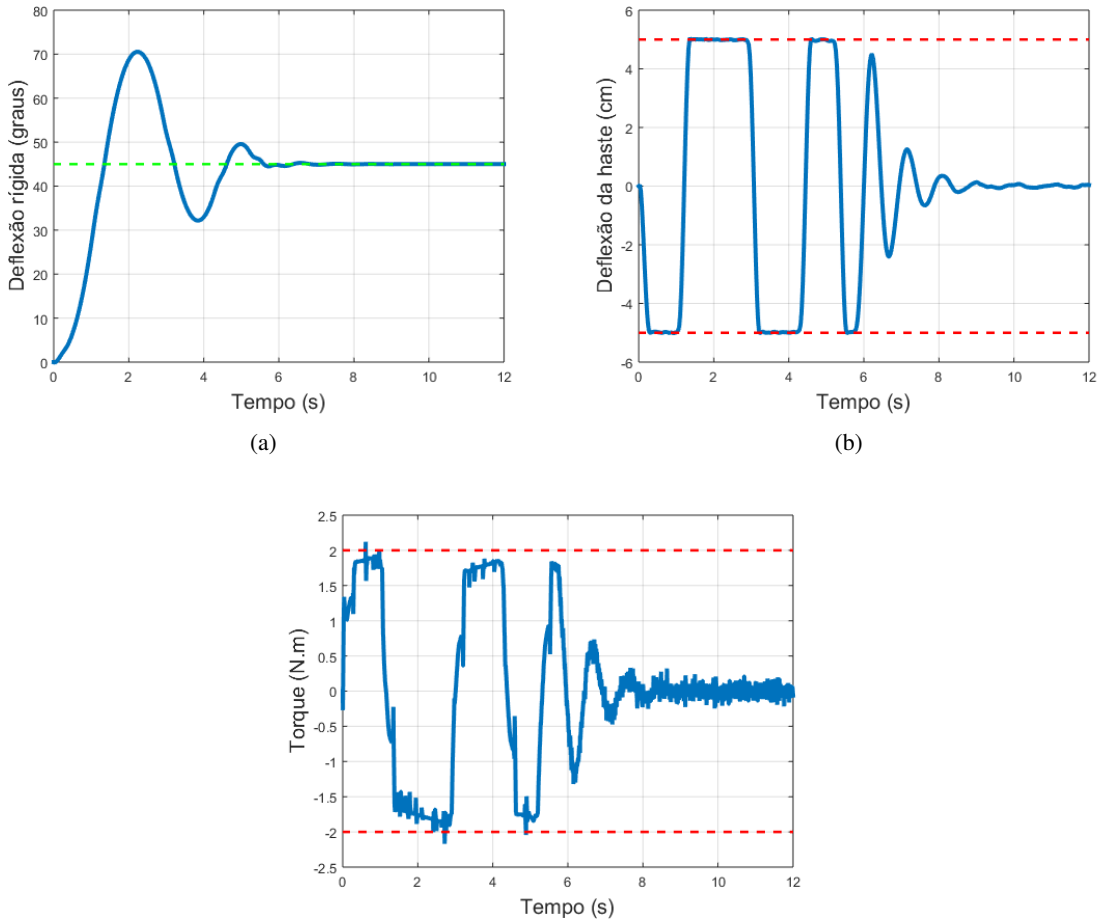


Figura 5.16: Comportamento do satélite utilizando o controlador MPC para o modelo não-linear em HIL, com um horizonte de predição de 20

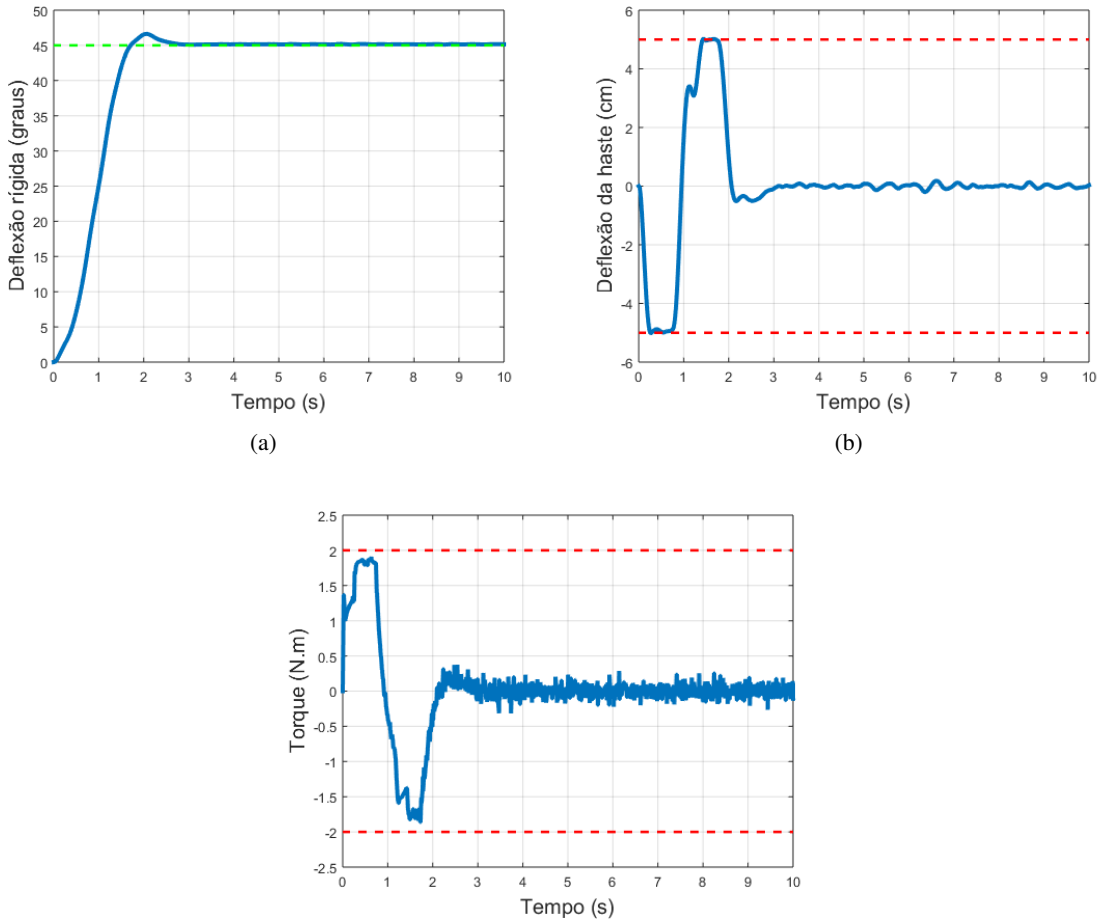


Figura 5.17: Comportamento do satélite utilizando o controlador MPC para o modelo não-linear em HIL, com um horizonte de predição de 60

Como se pode perceber, o comportamento do modelo não-linear é muito parecido com o comportamento do modelo linear, para todos os cenários simulados.

5.5 Análise dos Resultados Obtidos

Foram medidos, a nível de *software*, os tempos necessários para o algoritmo realizar uma iteração completa do algoritmo de controle, ou seja, o tempo levado para a placa *Beaglebone*, a partir do recebimento dos valores de estados atuais, calcular a variável de comando ótima e enviá-la ao modelo. Os tempos medidos estão mostrados na tabela a seguir.

Tabela 5.4: Tempos de cálculo da variável de comando pelo controlador

Controlador	**Modelo**	**Tempo de cálculo**
LQR	Linear	$500\mu s$
LQR	Não-linear	$500\mu s$
MPC (N = 20)	Linear	$2,7ms$
MPC (N = 20)	Não-linear	$2,7ms$
MPC (N = 60)	Linear	$6,1ms$
MPC (N = 60)	Não-linear	$5,9ms$

Percebe-se que o tempo de cálculo para o LQR é muito inferior ao tempo de cálculo para o MPC, em qualquer um dos cenários simulados. Esse resultado é uma evidência da baixa complexidade do LQR, se comparada com o MPC, já que os ganhos de realimentação do LQR somente são calculados uma vez *off-line* para gerar a variável de comando, enquanto o algoritmo do MPC deve realizar uma rotina de otimização diferente a cada iteração para calcular a variável de comando, gerando um custo computacional mais alto.

Além disso, percebe-se que há um aumento considerável no tempo de cálculo quando se aumenta o horizonte de predição do MPC de 20 para 60, o que já era esperado, já que o horizonte de predição impacta na complexidade do algoritmo de controle.

As tabelas a seguir mostram os indicadores de desempenho utilizados para avaliar as simulações realizadas neste trabalho. Os indicadores escolhidos foram o sobressinal (M_p), o tempo de assentamento de 2% (T_a) associados à deflexão rígida (θ) e a máxima deflexão da ponta da haste durante a simulação (w_{max}).

Tabela 5.5: Indicadores de desempenho para as simulações realizadas em MIL para o controlador LQR

Modelo	M_p	T_a	w_{max}
Linear	$8,0\%$	$1,64s$	$39,9cm$
Não-linear	$17,9\%$	$2,48s$	$36,8cm$
Linear com saturação	$17,0\%$	$2,54s$	$10,2cm$
Não-linear com saturação	$26,3\%$	$4,18s$	$8,9cm$

Tabela 5.6: Indicadores de desempenho para as simulações realizadas em HIL para o controlador LQR

Modelo	M_p	T_a	w_{max}
Linear	$7,7\%$	$1,49s$	$38,2cm$
Não-linear	$8,2\%$	$1,49s$	$39,1cm$
Linear com saturação	$15,2\%$	$2,42s$	$9,4cm$
Não-linear com saturação	$16,8\%$	$2,87s$	$9,8cm$

Tabela 5.7: Indicadores de desempenho para as simulações realizadas em MIL para o controlador MPC

Modelo	M_p	T_a	w_{max}
Linear (N=20)	$59,6\%$	$7,12s$	$5cm$
Linear (N=60)	$3,0\%$	$2,3s$	$5cm$
Não-linear (N=20)	$63,9\%$	$7,4s$	$5cm$
Não-linear (N=60)	$3,2\%$	$2,28s$	$5cm$

Tabela 5.8: Indicadores de desempenho para as simulações realizadas em HIL para o controlador MPC

Modelo	M_p	T_a	w_{max}
Linear (N=20)	$51,0\%$	$5,0s$	$5cm$
Linear (N=60)	$2,9\%$	$2,23s$	$5cm$
Não-linear (N=20)	$56,7\%$	$5,5s$	$5cm$
Não-linear (N=60)	$3,6\%$	$2,3s$	$5cm$

Ao se comparar os resultados do LQR comum com os do LQR com saturação na variável de comando, pode-se constatar que a inclusão da saturação faz aumentar o tempo de assentamento e o sobressinal do sistema. Por outro lado, reduz-se a amplitude de oscilação da haste flexível.

Quanto aos resultados obtidos com o controlador MPC, percebe-se que um aumento no horizonte de predição faz reduzir bastante o tempo de assentamento e o sobressinal, mantendo-se a deflexão da ponta da haste restrita no valor de 5 cm.

Em todas as simulações, tanto para o LQR, quanto para o MPC, notou-se que o comportamento do sistema linear e o comportamento do sistema não-linear se assemelham bastante, revelando uma característica interessante da dinâmica do sistema: pouca influência dos termos não-lineares. Isso explica também o fato de o controlador MPC ter sido capaz de controlar o sistema não-linear, apesar de ser um controlador baseado em modelo linear.

Comparando os dois controladores utilizados, percebe-se que o LQR possui um bom desempenho ao estabilizar o sistema com maior rapidez que o MPC. O custo para o LQR controlar o sistema com menor tempo de assentamento é a grande demanda no torque para movimentar o satélite e as grandes amplitudes de vibração da haste flexível, além do maior sobressinal apresentado. Saturando-se a saída do controlador LQR, consegue-se diminuir bastante a vibração, mas ainda se observa amplitudes indesejáveis da ordem de 10 cm.

A técnica de MPC aparece como boa alternativa para limitar as amplitudes de vibração, já que seu algoritmo lida diretamente com este tipo de restrição. Como se pôde perceber, para um horizonte de predição de 20 passos, a amplitude de vibração ficou bem controlada em 5 centímetros. Para o cenário com horizonte de predição 60, obtém-se sobressinal bem menor que aqueles obtidos para o LQR e para o cenário com horizonte de predição 20. O tempo de assentamento para este caso é apenas um pouco maior que aquele obtido para o LQR, estando dentro de limites aceitáveis.

O cenário para um horizonte de predição de 60 passos oferece então um bom *trade-off* para o sistema, pois estabiliza o sistema relativamente rápido, com sobressinal baixo, e com a haste oscilando dentro das restrições de 5 centímetros imposta.

Capítulo 6

Conclusões

Este capítulo apresenta as conclusões do presente trabalho, assim também como sugestões para trabalhos futuros que possam complementar a linha de pesquisa desenvolvida até aqui com a plataforma HIL para validação de controladores para satélites com estruturas flexíveis.

6.1 Conclusões do Trabalho

Este trabalho teve como objetivo projetar controladores de atitude para um modelo de satélite rígido flexível e validá-los numa plataforma HIL. A partir dos resultados obtidos e também ao longo do desenvolvimento do trabalho, pôde-se observar algumas conclusões, que serão listadas a seguir.

- A plataforma HIL montada mostrou-se satisfatória, tendo validado os controladores LQR e MPC para diversos cenários montados, tanto para o modelo linear quanto para o não-linear. Tal validação pode ser atestada pela similaridade entre as simulações em software e as simulações na plataforma HIL.
- Conseguiu-se implementar numa plataforma HIL de baixo custo e de processamento limitado o algoritmo do MPC parametrizado obtendo bons resultados.
- O comportamento do modelo não-linear assemelha-se bastante com o do modelo linearizado. Isso acontece porque as não-linearidades não são fortes o suficiente, consistindo de produtos entre estados do sistema.
- A proximidade de comportamento entre os modelos linear e não-linear permitiu que mesmo uma técnica baseada em modelo linear, como a do MPC, pudesse controlar o sistema com modelo não-linear.
- No cenário com horizonte de predição 60, o controlador MPC apresenta melhor desempenho que o LQR, quando se leva em consideração a vibração da haste flexível, pois esse controlador consegue manter a amplitude de vibração da haste flexível dentro de uma faixa predeterminada, com menor sobressinal e com tempo de assentamento relativamente baixo.

6.2 Sugestões para Trabalhos Futuros

A seguir são apresentadas algumas sugestões para trabalhos futuros a serem desenvolvidos, continuando o estudo de controle de atitude para o modelo de satélite rígido-flexível utilizado neste trabalho:

- Implementar um observador de estados para observar os estados η_1, η_2, bem como suas derivadas, já que esses estados não possuem significados físicos mensuráveis através de sensores como os demais, θ e $\dot{\theta}$.
- Em vez de utilizar conexão Ethernet/UDP, utilizar conexões analógicas na plataforma HIL, tanto para enviar o comando do controlador para a planta, quanto para enviar os estados da planta para o controlador, aproximando mais a arquitetura de uma malha de controle com o processo físico real.
- Testar outras técnicas de controle (PID, H_∞, SDRE, etc.) para o mesmo modelo de satélite, comparando seus desempenhos com aqueles obtidos neste trabalho.
- Utilizar outras placas microcontroladoras, tais como *Raspberry Pi*, *Arduino* ou *Odroid*, para rodar os algoritmos de controle desenvolvidos para o modelo em questão, a fim de analisar seus impactos na dinâmica do sistema.
- Utilizar um sistema operacional de tempo real na *Beaglebone*, tal como o *Xenomai* ou o *FreeRTOS*, para realizar a plataforma HIL inteiramente em tempo real, aproximando-a mais de uma aplicação com o processo físico real.
- Incluir no modelo de satélite as equações do atuador, um sistema que tenha como entrada uma tensão vinda do controlador, e, como saída, o torque aplicado ao satélite, como por exemplo um motor.
- Embarcar o controlador em um protótipo de satélite com estrutura flexível real, para validar as técnicas de controle implementadas em HIL.

REFERÊNCIAS BIBLIOGRÁFICAS

ALAMIR, M. *A Pragmatic Story of Model Predictive Control: Self-Contained Algorithms and Case Studies*. Primeira. [S.l.]: CreateSpace, 2013.

BAYAT, F. Conceptual design of a low-cost real-time hardware-in-the-loop simulator for satellite attitude control system. *Turkish Journal of Electrical Engineering and Computer Sciences*, v. 23, p. 789–803, 2015.

CAO, Y.; CHEN, W. Variable sampling-time nonlinear model predictive control of satellites using magneto-torquers. *Systems Science and Control Engineering*, v. 2, n. 1, p. 593–601, 2014.

CASTRO, J. *Estudo experimental da dinâmica e do sistema de controle de um satélite rígido-flexível*. Dissertação (Mestrado), 2009.

CHARBONNEL, C. h_∞ and lmi attitude control design: towards performances and robustness enhancement. *Acta Astronautica*, v. 54, p. 307–314, 2002.

CHEGENI, E.; ZANDIEH, M.; EBRAHIMI, J. Attitude control of satellite with pulse-width pulse-frequency (pwpf) modulator using generalized incremental predictive control. *Majlesi Journal of Electrical Engineering*, v. 8, n. 3, p. 25–31, 2014.

COLEY, G. *BeagleBone Black System Reference Manual*. [S.l.], 2013.

CORPORATION, M. C. *PCI-DAC6703 User's Guide*. [S.l.], 2009.

CROMWELL, C. *Development and Analysis of a Small Satellite Attitude Determination and Control System Testbed*. Dissertação (Mestrado), 2011.

EREN, U. et al. Model predictive control in aerospace systems: Current state and opportunities. *Journal of guidance, control and dynamics*, 2017.

FENILI, A.; SOUZA, L. Control of a nonlinear slewing flexible beam. In: *International congres of theoretical and applied mechanics*. [S.l.: s.n.], 2004. p. 2.

FERREAU, H. J. *qpOASES User's Manual*. [S.l.], 2014.

GERVINI, V. *Controle adaptativo de estruturas flexíveis*. Dissertação (Mestrado), 2003.

GORINEVSKY, D.; VUKOVICH, G. Nonlinear input shaping control of flexible spacecraft reorientation maneuver. *AIAA - Journal of Guidance, Control, and Dynamics*, v. 21, n. 2, p. 264–270, 1998.

GRAVADE, M. *Modelagem e controle ativo de vibrações de painéis flexíveis de satélites artificiais empregando transdutores piezoelétricos*. Dissertação (Mestrado), 2009.

GU, D.; PETKOV, P.; KONSTANTINOV, M. *Robust Control Design with MATLAB*. [S.l.]: Springer-Verlag, 2005.

KUKRETI, S. et al. Genetic algorithm based lqr for attitude control of a magnetically actuated cubesat. *AIAA SciTech*, 2015.

LOPES, I. *Controle de atitude de satélites rígido-flexíveis usando a otimização extrema generalizada com abordagem multi-objetivo*. Dissertação (Mestrado), 2008.

MAYNE, D. et al. Constrained model predictive control: Stability and optimality. *Automatica*, n. 36, p. 789–814, 2000.

OGATA, K. *Engenharia de Controle Moderno*. Quinta. [S.l.]: Pearson, 2010.

PEIXOTO, P.; MURILO, A.; SOUZA, L. Controle preditivo baseado em modelo para satélites com estruturas flexíveis. In: *Congresso Ibero Latino Americano de Métodos Computacionais em Engenharia - CILAMCE*. [S.l.: s.n.], 2016.

PINHEIRO, E.; SOUZA, L. Design of the microsatellite attitude control system using the mixed h_2/h_∞ method via lmi optimization. *Mathematical Problems in Engineering*, 2013.

PITTET, C.; MIGNOT, J.; FALLET, C. Lmi based multi objective h_∞ control of flexible microsatellite. In: *Conference on Decision and Control*. [S.l.: s.n.], 2000. p. 4000–4005.

RODRIGUES, R. et al. Sistema gps: uma órbita em evolução - número de satélites e periodicidade. *Energia na Agricultura*, v. 29, n. 2, p. 115–120, 2014.

SAAD, M.; AKHRIF, O.; SAYDY, L. Analytical model of one flexible link system with nonlinear kinematics. *Journal of Vibration and Control*, v. 19, p. 1795–1806, 2012.

SALES, T.; RADE, D.; SOUZA, L. Passive vibration control of flexible spacecraft using shunted piezoelectric transducers. *Aerospace Science and Technology*, v. 29, n. 1, p. 403–412, 2013.

SILVA, R. *Projeto de controladores para um sistema de direção elétrica utilizando a metodologia de projeto baseado em modelos*. Dissertação (Mestrado), 2017.

SILVA, S. *Controle ativo de vibração de um satélite rígido-flexível durante manobras de atitude*. Dissertação (Mestrado), 2000.

SOUZA, A. *Estudo do uso de métodos de controle robusto em sistemas espaciais rígidos - flexíveis*. Thesis, 2017.

SOUZA, L. *Dynamics and robust control for uncertain flexible space systems*. Thesis, 1992.

SOUZA, L.; ARENA, V. Design of satellite attitude control algorithm based on the sdre method using gas jets and reaction wheels. *Journal of Engineering*, v. 2013, 2012.

SOUZA, L.; GALVÃO, B. Experimental design of robust controller for a rigid flexible satellite. In: *Congresso Internacional de Engenharia Mecânica - COBEM2015*. [S.l.: s.n.], 2015. p. 8.

STINGA, F. et al. Optimal and mpc control of the quanser flexible link experiment. In: *International Conference on Dynamical and Control*. [S.l.: s.n.], 2008. p. 6.

TAFAZOLI, S.; KHORASANI, K. Nonlinear control and stability analysis of spacecraft attitude recovery. *IEEE Transactions on Aerospace and Electronic Systems*, v. 42, n. 3, p. 825–845, 2006.

VALDIVIA, R. *Influência da flexibilidade no desempenho de um sistema de controle de atitude de um satélite rígido-flexível*. Dissertação (Mestrado), 2007.

VICENTE, A.; RIBEIRO, J. Simulação hil do controle de atitude de satélites artificiais. In: *Congresso Nacional de Engenharia Mecânica - CONEM2014*. [S.l.: s.n.], 2014.

ZHANG, G. et al. Nonlinear control of flexible spacecraft based on lmi constrained convex optimization. In: *International Symposium on Systems and Control in Aerospace and Astronautics*. [S.l.: s.n.], 2008. p. 1–5.