

Os sistemas robóticos com pernas possuem uma capacidade de locomoção versátil, graças às técnicas empregadas no controle da caminhada. Entretanto, situações adversas, como uma queda, podem levar o sistema para condições em que ocorre a perda momentânea da locomoção, de modo que se faz necessária alguma estratégia capaz de recuperar sua mobilidade. Esta estratégia é conhecida como *self-righting*, e é dividida em etapas. Uma destas etapas é o *stand up*, e se caracteriza pelo movimento que o robô realiza para ficar em pé novamente. Assim, este trabalho propõe uma solução para o problema de *stand up* de um robô quadrúpede. A solução é baseada em *Reference Governor Control* e implementada via Controlador Preditivo Baseado em Modelo, considerando, no cálculo dos sinais de controle, restrições do robô e do ambiente, garantindo que o sistema retome a capacidade de locomoção de forma segura. A proposta foi testada em ambiente simulado e o desempenho foi avaliado segundo diferentes funções de custo. A solução encontrada se mostrou efetiva para a solução do problema e mostra um grande potencial para tratar de outras subtarefas do *self-righting*.

Orientador: Douglas Wildgrube Bertol

Coorientador: Victor Barasuol

Joinville, 2020

ANO  
2020

AUREO GUILHERME DOBRIKOPF | SELF-RIGHTING DE SISTEMAS ROBÓTICOS COM  
PERNAS SOB UMA PERSPECTIVA DA ARQUITETURA DE GOVERNOR REFERENCE CONTROL  
REALIZADA COM ESTRATÉGIA DE CONTROLE PREDITIVO



UDESC

UNIVERSIDADE DO ESTADO DE SANTA CATARINA – UDESC  
CENTRO DE CIÊNCIAS TECNOLÓGICAS– CCT  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA - PPGEEL

DISSERTAÇÃO DE MESTRADO

**SELF-RIGHTING DE SISTEMAS  
ROBÓTICOS COM PERNAS SOB  
UMA PERSPECTIVA DA  
ARQUITETURA DE GOVERNOR  
REFERENCE CONTROL REALIZADA  
COM ESTRATÉGIA DE CONTROLE  
PREDITIVO**

AUREO GUILHERME DOBRIKOPF

JOINVILLE, 2020

AUREO GUILHERME DOBRIKOPF

***SELF-RIGHTING* DE SISTEMAS ROBÓTICOS COM PERNAS SOB UMA  
PERSPECTIVA DA ARQUITETURA DE *GOVERNOR REFERENCE*  
*CONTROL* REALIZADA COM ESTRATÉGIA DE CONTROLE PREDITIVO**

Dissertação de mestrado, submetida à Universidade do Estado de Santa Catarina como requisito parcial para obtenção do grau de Mestre em Engenharia Elétrica.

Orientador: Douglas Wildgrube Bertol

Co-orientador: Victor Barasuol

Joinville

2020

**Ficha catalográfica elaborada pelo programa de geração automática da  
Biblioteca Setorial do CCT/UDESC,  
com os dados fornecidos pelo(a) autor(a)**

Dobrikopf, Aureo Guilherme  
SELF-RIGHTING DE SISTEMAS ROBÓTICOS COM  
PERNAS SOB UMA PERSPECTIVA DA ARQUITETURA DE  
GOVERNOR REFERENCE CONTROL REALIZADA COM  
ESTRATÉGIA DE CONTROLE PREDITIVO / Aureo Guilherme  
Dobrikopf. -- 2020.  
99 p.

Orientador: Douglas Wildgrube Bertol  
Coorientador: Victor Barasuol  
Dissertação (mestrado) -- Universidade do Estado de Santa  
Catarina, Centro de Ciências Tecnológicas, Programa de  
Pós-Graduação em Engenharia Elétrica, Joinville, 2020.

1. Robô quadrupede. 2. Self-righting. 3. Reference Governor  
Control. 4. Controle Preditivo Baseado em Modelo. I. Bertol,  
Douglas Wildgrube. II. Barasuol, Victor. III. Universidade do  
Estado de Santa Catarina, Centro de Ciências Tecnológicas,  
Programa de Pós-Graduação em Engenharia Elétrica. IV. Título.

***Self-righting* de Sistemas Robóticos com Pernas sob uma Perspectiva da  
Arquitetura de *Governor Reference Control* Realizada com Estratégia de  
Controle Preditivo**

por

**Aureo Guilherme Dobrikopf**

Esta dissertação foi julgada adequada para obtenção do título de

**MESTRE EM ENGENHARIA ELÉTRICA**

Área de concentração em “Sistemas Eletroeletrônicos”  
e aprovada em sua forma final pelo

CURSO DE MESTRADO ACADÊMICO EM ENGENHARIA ELÉTRICA  
DO CENTRO DE CIÊNCIAS TECNOLÓGICAS DA  
UNIVERSIDADE DO ESTADO DE SANTA CATARINA.

Banca Examinadora:

ASSINADO DIGITALMENTE

Prof. Dr. Douglas Wildgrube Bertol  
CCT/UDESC  
(Orientador/Presidente)

VIA VIDEOCONFERÊNCIA

Prof. Dr. Ebrahim Samer El Youssef  
UFSC

ASSINADO DIGITALMENTE

Profa. Dra. Mariana Santos Matos  
Cavalca  
CCT/UDESC

**Joinville, SC, 29 de julho de 2020.**



*À minha familiazinha, por ser minha pedra fundamental*

## AGRADECIMENTOS

Aos meus pais Ede e Simone, por tudo que fizeram e fazem por mim. Vocês são os exemplos da minha vida, o que sou hoje é graças a vocês. A minha irmãzinha Hayane, por todo o carinho e amor. A todos os demais familiares, obrigado por todo carinho e amor dedicados ao longo da minha vida.

Aos meus orientadores Douglas Wildgrube Bertol e Victor Barasuol, por todo o conhecimento transmitido durante esta jornada.

Aos colegas do Laboratório de Pesquisa em Automação e Sistema, pela convivência ao longo destes anos. Em especial ao meu amigo Lucas Schulze, pelas conversas e discussões que proporcionaram novas ideias.

Aos meus amigos da Pós-Graduação: Henrique, Mateus, Renan e Rodolfo, pelas trocas de experiências. Não podemos esquecer do PUBzinho.

Aos meus amigos Germano, Rafael e Ricardo, por esta amizade que perdura desde o ensino médio.

Por fim, mas não menos importante: à minha namora Jéssica, por estar no meu lado todos estes anos, por não me deixar desistir e me apoiar no momentos difíceis. Obrigado por tudo pequena.

*Se você entende como o universo funciona, de  
certa forma pode controlá-lo.*

Stephen Hawking

## RESUMO

Os sistemas robóticos com pernas possuem uma capacidade de locomoção versátil, graças às técnicas empregadas no controle da caminhada. Entretanto, situações adversas, como uma queda, podem levar o sistema para condições em que ocorre a perda momentânea da locomoção, de modo que se faz necessária alguma estratégia capaz de recuperar sua mobilidade. Esta estratégia é conhecida como *self-righting*, e é dividida em etapas. Uma destas etapas é o *stand up*, e se caracteriza pelo movimento que o robô realiza para ficar em pé novamente. Assim, este trabalho propõe uma solução para o problema de *stand up* de um robô quadrúpede. A solução é baseada em *Reference Governor Control* e implementada via Controlador Preditivo Baseado em Modelo, considerando, no cálculo dos sinais de controle, restrições do robô e do ambiente, garantindo que o sistema retome a capacidade de locomoção de forma segura. A proposta foi testada em ambiente simulado e o desempenho foi avaliado segundo diferentes funções de custo. A solução encontrada se mostrou efetiva para a solução do problema e mostra um grande potencial para tratar de outras subtarefas do *self-righting*.

**Palavras-chave:** Robô quadrúpede. *Self-righting*. *Reference Governor Control*. Controle Preditivo Baseado em Modelo.

## ABSTRACT

The legged robot systems has a versatile locomotion capacity, due to the control technics used to the walking task. However, adverse situations, *e.g* a fall, may lead the system in an condition where momentary occurs the lost of locomotion capacity, in a way that makes necessary some recovering mobility strategy. This strategy is known as self-righting, and it is performed in phases. One of these phases is the stand up, and is characterized by the moviment of the robot to become upright. So this work proposes a solution to the stand up problem for a quadrupedal robot. The gotten solution is based in Reference Governor Control and implemented through Model Based Predictive Control, considering the system and environment cosntraints, ensuring that the system recovers the locomotion capacity in a safety way. The proposal has been tested in a simuleted environment and the performance has been rated by differents cost functions. The found solution shown effective to the problem and a potencial to treat others subtasks of self-righting.

**Keywords:** Quadrupedal Robot. Self-righting. Reference Governor Control. Model Based Predictivel Control.

## LISTA DE FIGURAS

FIGURA 1 – Locomoção em diferentes perfis de terreno: pernas vs. rodas. . . . .	18
FIGURA 2 – Etapas observadas durante a reorientação do <i>Eucryp torrhynchus brandti</i> . .	22
FIGURA 3 – Plataforma robótica HReX da <i>Boston Dynamics</i> . . . . .	24
FIGURA 4 – Etapas do <i>self-righting</i> do HyQ2Max. . . . .	25
FIGURA 5 – Etapas da realização do <i>self-righting</i> de um robô simétrico. . . . .	26
FIGURA 6 – Estrutura de controle da tarefa de <i>self-righting</i> do HyQ2Max. . . . .	28
FIGURA 7 – Estrutura de controle proposta para a tarefa de <i>self-righting</i> do HyQ2Max.	30
FIGURA 8 – Estrutura genérica do <i>Reference Governor Control</i> . . . . .	33
FIGURA 9 – Numeração dos corpos de um sistema de base flutuante. . . . .	45
FIGURA 10 – Abstração da dinâmica do centroide. . . . .	48
FIGURA 11 – Numeração dos elos e juntas do HyQ2Max. . . . .	52
FIGURA 12 – Representação das juntas que compõe as patas do robô. . . . .	52
FIGURA 13 – Diagrama de blocos da arquitetura <i>Reference Governor Control</i> proposta para a tarefa de <i>Self-Righting</i> . . . . .	63
FIGURA 14 – RGC-PD+P, função custo da Equação (5.2): trajetórias das juntas $\mathbf{q}$ e das referências moduladas $\mathbf{w}$ . . . . .	67
FIGURA 15 – RGC-PD+P, função custo da Equação (5.2): comportamento do torque. . .	67
FIGURA 16 – RGC-PD+P, função custo da Equação (5.3): trajetórias das juntas $\mathbf{q}$ e das referências moduladas $\mathbf{w}$ . . . . .	68
FIGURA 17 – RGC-PD+P, função custo da Equação (5.3): comportamento do torque. . .	68
FIGURA 18 – RGC-PD+P, função custo da Equação (5.4): trajetórias das juntas $\mathbf{q}$ e das referências moduladas $\mathbf{w}$ . . . . .	69
FIGURA 19 – RGC-PD+P, função custo da Equação (5.4): comportamento do torque. . .	69
FIGURA 20 – RGC-PD+P: comparação das trajetórias das juntas $\mathbf{q}$ para as três funções custo. . . . .	70
FIGURA 21 – RGC-PD+P: comparação das referências moduladas $\mathbf{w}$ para as três funções custo. . . . .	70
FIGURA 22 – RGC-MdlSimp, função custo da Equação (5.3): trajetórias das juntas $\mathbf{q}$ e das referências moduladas $\mathbf{w}$ . . . . .	73

FIGURA 23 – RGC-MdlSimp, função custo da Equação (5.3): comportamento dos torques.	73
FIGURA 24 – RGC-MdlSimp, função custo da Equação (5.4): trajetórias das juntas $\mathbf{q}$ e das referências moduladas $\mathbf{w}$ .	74
FIGURA 25 – RGC-MdlSimp, função custo da Equação (5.4): comportamento do torque.	74
FIGURA 26 – RGC-MdlSimp, função custo da Equação (5.5): trajetórias das juntas $\mathbf{q}$ e das referências moduladas $\mathbf{w}$ .	75
FIGURA 27 – RGC-MdlSimp, função custo da Equação (5.5): comportamento do torque.	75
FIGURA 28 – RGC-MdlSimp: comparação das trajetórias das juntas $\mathbf{q}$ para as três funções custo.	76
FIGURA 29 – RGC-MdlSimp: comparação das referências moduladas $\mathbf{w}$ para as três funções custo.	76
FIGURA 30 – Trajetórias das juntas $\mathbf{q}$ e das referências moduladas $\mathbf{w}$ considerando restrições de torque e força de contato.	78
FIGURA 31 – Comportamento dos torques considerando restrições de torque e força.	78
FIGURA 32 – Comportamento das forças de reação entre os pés e o solo das pernas LF e RF.	80
FIGURA 33 – Comparação das trajetórias dos pés para a situação sem e com restrição de forças de contato.	80
FIGURA 34 – Trajetória dos centro de massa ao considerar restrição de ${}^w r_z$ .	81
FIGURA 35 – Trajetórias das juntas $\mathbf{q}$ e das referências moduladas $\mathbf{w}$ considerando restrições na componente ${}^w r_z$ .	81

## LISTA DE TABELAS

TABELA 1	–	Quadro resumo do estado da arte. . . . .	27
TABELA 2	–	Numeração das juntas de cada pata do robô. . . . .	51
TABELA 3	–	Parâmetros de Denavit-Hartenberg das patas do HyQ2Max. . . . .	53
TABELA 4	–	Deslocamento das origens das juntas HAA em relação ao centro geométrico do HyQ2Max. . . . .	53
TABELA 5	–	Posições dos CoMs e massas dos elos. . . . .	54
TABELA 6	–	RGC-PD+P: Erros médios absolutos . . . . .	71
TABELA 7	–	RGC-MdlSimp: Erros médios absolutos . . . . .	77
TABELA 8	–	Distância percorrida pelos pés. . . . .	79
TABELA 9	–	Abordagem RGC-PD+P: Tempo de cálculo da solução do problema de otimização . . . . .	82
TABELA 10	–	Abordagem RGC-MdlSimp: Tempo de execução da solução do problema de otimização . . . . .	83
TABELA 11	–	Parâmetros do MPC: abordagem RGC-PD+P. . . . .	97
TABELA 12	–	Parâmetros Controlador PD. . . . .	97
TABELA 13	–	Parâmetros Filtro de Kalman. . . . .	97
TABELA 14	–	Abordagem RGC-MdlSimp: parâmetros do MPC. . . . .	98
TABELA 15	–	Abordagem RGC-MdlSimp com restrição de forças: parâmetros do MPC. . . . .	99



## LISTA DE ABREVIATURAS

<b>CMM</b>	<i>Centroidal Momentum Matrix</i>
<b>CoM</b>	<i>Center of Mass</i>
<b>DMC</b>	<i>Dynamic Matrix Control</i>
<b>DoF</b>	<i>Degree of Freedom</i>
<b>GRF</b>	<i>Ground Reaction Force</i>
<b>HAA</b>	<i>Hip Abduction/Adduction</i>
<b>HFE</b>	<i>Hip Flexion/Extension</i>
<b>KFE</b>	<i>Knee Flexion/Extension</i>
<b>LF</b>	<i>Left-Front</i>
<b>LH</b>	<i>Left-Hind</i>
<b>MPC</b>	<i>Model Predictive Control</i>
<b>MPHC</b>	<i>Model Predictive Heuristic Control</i>
<b>OSQP</b>	<i>Operator Splitting Quadratic Program</i>
<b>PD</b>	Proporcional-Derivativo
<b>PI</b>	Proporcional-Integral
<b>PID</b>	Proporcional-Integral-Derivativo
<b>RF</b>	<i>Right-Front</i>
<b>RGC</b>	<i>Reference Governor Control</i>
<b>RH</b>	<i>Right-Hind</i>
<b>SISO</b>	<i>Single Input-Single Output</i>
<b>SR</b>	<i>Self-Righting</i>
<b>TH</b>	Transformação Homogênea
<b>URDF</b>	<i>Unified Robot Description Format</i>

## LISTA DE SÍMBOLOS

$\mathbf{I}_n$	Matriz identidade de ordem $n$
$\mathbf{0}_{n \times m}$	Matriz de zeros de ordem $n \times m$
$\mathbf{q}$	Posições das juntas, [rad]
$\mathbf{w}$	Referência modulada, [rad]
$\mathbf{x}$	Vetor de estados do sistema
$\mathbf{A}$	Matriz de dinâmicas do espaço de estados
$\mathbf{B}$	Matriz de entradas do espaço de estados
$\mathbf{u}$	Vetor de entradas do sistema
$\mathbf{y}$	Vetor de saídas do sistema
$\mathbf{C}$	Matriz de saídas do espaço de estados
$\mathbf{D}$	Matriz de transmissão direta do espaço de estados
$N$	Horizonte de predição
$M$	Horizonte de controle
$\mathbf{G}$	Matriz de predição
$\Phi$	Matriz de resposta livre do sistema
$\mathbf{Q}$	Matriz de ponderação dos estados
$\mathbf{R}$	Matriz de ponderação das entradas
$\mathbf{b}_{\min}$	Vetor dos valores mínimos da restrição
$\mathbf{b}_{\max}$	Vetor dos valores máximos da restrição
$\mathcal{N}_B$	Número de <i>links</i>
$n$	Número de juntas
${}^j\mathbf{T}_i$	Matriz de transformada homogênea
${}^j\mathbf{R}_i$	Matriz de rotação
${}^j\mathbf{p}_i$	Vetor entre as origens do sistema $j$ e $i$
$\nu$	Velocidade linear do efetuador, [m/s]
$\omega$	Velocidade angular do efetuador, [rad/s]
$\mathbf{J}_p$	Jacobiano translacional
$\varphi$	Descrição das orientações do corpo

$\mathbf{J}_r$	Jacobiano rotacional
$\mathbf{J}$	Jacobiano
${}^w\mathbf{c}$	Posição da base descrita no sistema global, [m]
$\chi$	Estados da base flutuante
$\Theta$	Orientação do centro de massa descrito no sistema global, [rad]
${}^w\mathbf{r}$	Posição do centro de massa descrito no sistema global, [m]
${}^b\mathbf{r}$	Posição do centro de massa descrito na base, [m]
$m_T$	Massa total do robô, [kg]
$\mathbf{H}$	Matriz de inércia no espaço de juntas
$\mathcal{C}$	Matriz de inércia no espaço de juntas
$\tau_g$	Torque devido à componente gravitacional, [Nm]
$\boldsymbol{\tau}$	Vetor de torque das juntas, [Nm]
$\mathbf{H}_u$	Parte não atuada da matriz de inércia no espaço de juntas
$\mathbf{H}_{ua}$	Parte inferior dos produtos cruzados da matriz de inércia no espaço de juntas
$\mathbf{H}_{au}$	Parte superior dos produtos cruzados da matriz de inércia no espaço de juntas
$\mathbf{H}_a$	Parte atuada da matriz de inércia no espaço de juntas
$\mathbf{h}_u$	Vetor da parte não atuada das componentes gravitacionais, centrífugas e de Coriolis
$\mathbf{h}_a$	Vetor da parte atuada das componentes gravitacionais, centrífugas e de Coriolis
$\mathbf{J}_{c,u}$	Jacobiano de contato da parte não atuada
$\mathbf{J}_{c,a}$	Jacobiano de contato da parte atuada
$\mathbf{f}$	Vetor de forças externas, [N]

$\mathbf{g}$	Vetor gravitacional, [m/s <sup>2</sup> ]
$\mathbf{f}_i$	Vetor de forças externas aplicada no $i$ -enésimo ponto, [N]
$\mathcal{I}$	Tensor inercial, [kg m <sup>2</sup> ]
$\mathbf{p}_{c,i}$	$i$ -ésimo ponto de contato, [m]
$K_p$	Ganho proporcional
$N_f$	Constante do filtro passa baixa
$K_d$	Ganho derivativo
$\mathbf{K}_p$	Matriz de ganhos proporcional
$\mathbf{K}_d$	Matriz de ganhos derivativo
$\mathbf{x}_r$	Vetor dos estados internos do Controlador PD
$e$	Erro
$\mathbf{Q}_q$	Matriz de ponderação do erro das juntas em relação à referência $\mathbf{q}_r$
$\mathbf{Q}_{\Delta w}$	Matriz de ponderação dos incrementos da referência modulada
$\mathbf{Q}_w$	Matriz de ponderação do erro da referência modulada em relação à referência $\mathbf{q}_r$
$\mathbf{q}_r$	Referências de alto nível das juntas, [rad]
$\mathbf{Q}_e$	Matriz de ponderação do erro da posição das juntas em relação à referência modulada
$t_s$	Tempo de amostragem, [s]

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO . . . . .</b>	<b>18</b>
1.1	CONTRIBUIÇÕES . . . . .	19
1.2	ORGANIZAÇÃO DO TRABALHO . . . . .	20
<b>2</b>	<b>ESTADO DA ARTE . . . . .</b>	<b>21</b>
2.1	PROBLEMA DO <i>SELF-RIGHTING</i> DO HYQ2MAX . . . . .	28
2.2	PROPOSTA DE ARQUITETURA PARA O <i>SELF-RIGHTING</i> . . . . .	29
<b>3</b>	<b>ARCABOUÇO TEÓRICO . . . . .</b>	<b>32</b>
3.1	<i>REFERENCE GOVERNOR CONTROL</i> . . . . .	32
3.1.1	Abordagens de implementação . . . . .	33
3.2	CONTROLE PREDITIVO BASEADO EM MODELO . . . . .	35
3.2.1	Fundamentação teórica do MPC . . . . .	37
3.3	MODELAGEM DE SISTEMAS ROBÓTICOS . . . . .	42
3.3.1	Cinemática . . . . .	42
3.3.2	Dinâmica . . . . .	46
3.4	SÍNTESE DO CAPÍTULO . . . . .	49
<b>4</b>	<b>DESENVOLVIMENTO . . . . .</b>	<b>51</b>
4.1	MODELO CINEMÁTICO . . . . .	51
4.2	MODELOS DE PREDIÇÃO E RESTRIÇÕES . . . . .	54
4.2.1	Primeira abordagem: RGC-PD+P . . . . .	55
4.2.2	Segunda abordagem: RGC-MdlSimp . . . . .	59
4.3	RESTRIÇÃO DA FORÇA DE CONTATO . . . . .	61
4.4	SÍNTESE DO CAPÍTULO . . . . .	62
<b>5</b>	<b>RESULTADOS . . . . .</b>	<b>63</b>
5.1	ABORDAGEM RGC-PD+P . . . . .	64
5.2	ABORDAGEM RGC-MDLSIMP . . . . .	71
5.3	SÍNTESE DO CAPÍTULO . . . . .	82
<b>6</b>	<b>CONCLUSÕES E TRABALHOS FUTUROS . . . . .</b>	<b>84</b>

<b>REFERÊNCIAS</b>	<b>86</b>
<b>APÊNDICES</b>	<b>91</b>
<b>A</b>	<b><i>REFERENCE GOVERNOR CONTROL COM LOOP DE CONTROLE</i></b>
	<b>INTERNO. 91</b>
<b>B</b>	<b>CONSIDERAÇÕES DE MODELAGEM 94</b>
B.1	MATRIZES DE ROTAÇÃO 94
B.2	CONVENÇÃO DE DENAVI-HARTEMBERG 94
<b>C</b>	<b>CONSIDERAÇÕES SOBRE AS SIMULAÇÕES 96</b>
C.1	HARDWARE E SOFTWARE. 96
C.2	ABORDAGEM RGC-PD+P: PSEUDO-CÓDIGO. 96
C.3	ABORDAGEM RGC-PD+P: PARÂMETROS DE SIMULAÇÃO. 97
C.4	ABORDAGEM RGC-MDLSIMP: PSEUDO-CÓDIGO. 98
C.5	PARÂMETROS DE SIMULAÇÃO DA ABORDAGEM RGC-MDLSIMP COM RESTRIÇÃO EM TORQUE 98
C.6	PARÂMETROS DE SIMULAÇÃO DA ABORDAGEM RGC-MDLSIMP COM RESTRIÇÃO EM TORQUE E FORÇAS DE REAÇÃO DO SOLO 99

## 1 INTRODUÇÃO

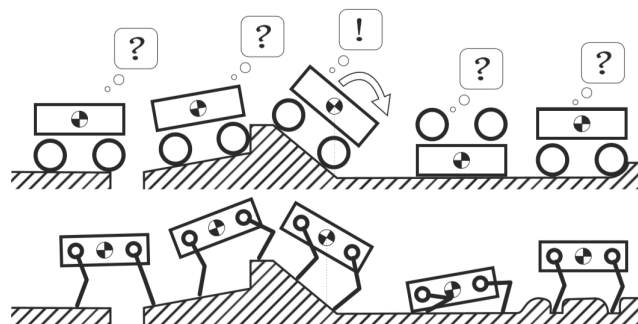
Cada vez mais, robôs fazem parte do cotidiano das pessoas. Podem ser empregados em atividades industriais, como os já consolidados robôs manipuladores, ou em atividades domiciliares, como na limpeza do chão (ACKERMAN, 2019). Robôs são progressivamente utilizados em situações de desastres, naturais ou não, para realizar operações de busca e resgate ou de reparos em situações perigosas (MURPHY, 2012). Existem outras atividades que não são adequadas e até mesmo inviáveis para humanos como, por exemplo, inspeções em locais insalubres (GUIZZO, 2011), ou exploração espacial (ELLERY, 2015), portanto a utilização destes mecanismos se torna natural.

Com exceção dos robôs manipuladores, os demais sistemas devem possuir a capacidade de se locomover nos ambientes. A roda é o sistema mais simples, e com a maior eficiência, utilizada para realizar a movimentação de robôs (SIEGWART; NOURBAKHS; SCARAMUZZA, 2011).

Entretanto, mecanismos robóticos que adotam este tipo de atuação podem enfrentar dificuldades de locomoção, devido às condições do ambiente. Situações corriqueiras, como a travessia de uma escadaria, podem impedir a realização de uma determinada missão para tais sistemas robóticos. Aplicações reais de busca e salvamento, na maioria das vezes, ocorrem em ambiente cobertos de entulhos ou lama, dificultando ou até impossibilitando o deslocamento de um sistema que se locomove por rodas.

Na Figura 1 são ilustradas algumas situações em que robôs com rodas falham ou tem dificuldade de superar obstáculos impostos pelo ambiente se comparado a um sistema robótico movido por pernas.

Figura 1 – Locomoção em diferentes perfis de terreno: pernas vs. rodas.



Fonte: disponível em Remy (2011).

Na natureza, entre animais terrestres, mas não unicamente, pernas são a forma predominante de locomoção. De forma inspiradora, este exemplo da evolução é perspicazmente copiado em sistemas robóticos. Este tipo de locomoção se caracteriza por uma interação de contato intermitente entre o robô e o solo, possibilitando assim uma boa adaptabilidade e manobrabilidade à terrenos irregulares (SIEGWART; NOURBAKHSH; SCARAMUZZA, 2011).

Robôs com pernas normalmente são inspirados em animais como insetos, mamíferos e répteis, portanto podendo ter seis, quatro e duas pernas (SIEGWART; NOURBAKHSH; SCARAMUZZA, 2011). Sistemas com seis patas são mais estáveis, entretanto possuem um maior custo devido ao número de atuadores. Sistemas com duas pernas tem menor custo, novamente devido ao número de atuadores, entretanto o controle destes sistemas é mais complexo, devido a sua essência instável. Assim, um robô quadrúpede aparece como uma opção de custo-benefício a ser considerado: maior número de contatos que um bípede e menos atuadores que um hexápode (SEMINI, 2010).

A locomoção de quadrúpedes já tem um vasto interesse pela comunidade científica (LI et al., 2011). Entretanto, um dos assuntos que padece de soluções se baseia na ocorrência natural de acidentes ou eventos que levam o sistema a uma situação que impossibilite a sua locomoção, por exemplo uma queda, desta maneira, o sistema deve ser capaz de se recuperar. Este problema denomina-se como tarefa de *self-righting*, o que, em termos gerais, significa reerguer-se.

Portanto, esta dissertação tem como objetivo geral estabelecer uma estratégia que permita um robô quadrúpede reerguer-se de maneira adequada, ou seja, realizar o *self-righting*, para isso conta com os seguintes objetivos específicos:

- Levantamento das estratégias atuais empregadas para a realização do *self-righting*;
- Obter o modelo que represente minimamente o comportamento do robô durante o *self-righting*;
- Estudo e projeto da arquitetura para realizar o *self-righting*;
- Implementação da estratégia e simulação da tarefa.

## 1.1 CONTRIBUIÇÕES

Este trabalho propõe uma solução para a tarefa de *self-righting* de robôs quadrúpedes usando a arquitetura *Reference Governor Control*, implementado via Controlador Preditivo



Baseado em Modelo. A estrutura obtida permite modular as referências de alto nível da tarefa, considerando restrições internas do sistema ou, até mesmo, impostas pelo ambiente aonde o robô está inserido.

No que diz respeito às limitações internas, a solução considera as restrições de torque nas juntas do robô, a fim de garantir o funcionamento seguro dos atuadores.

Já em relação às restrições impostas pelo ambiente, dependendo do tipo de solo, o robô pode escorregar ao tentar realizar a tarefa, necessitando levar em consideração as máximas forças de reação entre os pontos de contato do robô com o solo. Também foram consideradas restrições na posição vertical do centro de massa, uma vez que, dependendo do local que o robô se encontra, a postura final pode ser impraticável devido às limitações espaciais do ambiente.

## 1.2 ORGANIZAÇÃO DO TRABALHO

O trabalho está dividido da seguinte maneira: no Capítulo 2 é realizado o levantamento do estado da arte e a definição formal do *self-righting*. No Capítulo 3 estão descritas as técnicas e a arquitetura de controle utilizadas na obtenção da solução de *self-righting*, além das descrições dos passos e considerações tomadas para obter os modelos cinemático e dinâmico empregados no controle de robôs com pernas. No Capítulo 4 estão descritos os modelos utilizado para a implementação da arquitetura de controle proposta. Os resultados obtidos por meio de simulações estão no Capítulo 5. Por último, as conclusões e perspectivas futuras deste trabalho estão discutidas no Capítulo 6.

## 2 ESTADO DA ARTE

Em aplicações reais, robôs com pernas podem sofrer com instabilidades dinâmicas ou estáticas, perturbações rotacionais, perda dos pontos de apoio ou a incapacidade de gerar a força de reação com o solo necessária para manter a postura correta. Estas condições adversas, podem levar o robô a uma queda e, conseqüentemente, provocar um estado que impossibilita a sua locomoção (LI et al., 2016). Logo, é necessário que o sistema seja dotado de alguma estratégia que seja capaz de colocar o robô em pé novamente, de forma autônoma, após uma eventual queda (SEMINI et al., 2016). Tal estratégia é conhecida como *Self-Righting* (SR), que pode ser realizada de forma ativa ou passiva (ZHANG et al., 2012).

O *self-righting* passivo se baseia na construção física do robô para impingir condições de retorno a uma postura inicial segura, sem a necessidade de atuadores. Um exemplo deste tipo é o robô saltador com pernas desenvolvido por Kovač et al. (2010), que exibe um mecanismo de locomoção envolto por uma estrutura ovalada de fibra de carbono, em conjunto com características de distribuição de peso, que geram o comportamento passivo de recomposição após uma queda. Este robô naturalmente rotaciona em seu eixo de simetria e retoma à configuração que permite a retomada da locomoção.

Em Dynamics (2012), é possível constatar o robô quadrúpede LS3 da *Boston Dynamics* realizando o *self-righting* de forma passiva utilizando estruturas arredondadas nas laterais do corpo. Desta forma, depois de uma eventual queda, o sistema robótico passivamente realiza um rolamento no eixo longitudinal, e retoma a sua orientação correta.

Assim, caso o robô seja acometido por alguma situação que o leve a um estado não previsto, o sistema não terá condições de se recuperar, a não ser que possua uma forma ativa de reposicionamento. Portanto o SR ativo, embora requeira mais mecanismos e aparatos para controlá-los, permite a aplicação do robô em uma gama maior de problemas, visto que o torna mais robusto a situações adversas do ambiente (ZHANG et al., 2012).

Na natureza, muitos são os exemplos de animais que têm a capacidade de se recuperar de uma queda, possibilitando se adaptarem a diversos tipos de terreno (ZHANG et al., 2012), logo é natural observar a solução do processo evolutivo para inspirar soluções de *self-righting* para robôs.

Zhang et al. (2012) propõem um robô bípede saltador, que pode realizar o *self-righting*

de forma ativa, utilizando uma terceira perna que possui apenas um grau de liberdade (DoF)<sup>1</sup>, nomeada de *pole leg*. Por meio das observações do comportamento cinemático e dinâmico que a *pole leg* impõe ao robô, foram estabelecidas algumas sequências de configurações, que dependem da postura inicial, para que o robô se reoriente. Neste trabalho, encontra-se a descrição construtiva do robô saltador, considerações sobre sua modelagem e uma comparação com outros robôs saltadores já existentes à época. Ademais, os autores não fazem menção sobre a arquitetura de controle do sistema, apenas que alguns trabalhos futuros são esperados nessa linha de pesquisa. Os testes experimentais da proposta foram realizados em uma superfície plana, para dois estados iniciais distintos.

No trabalho de Peng et al. (2017), foi realizada uma minuciosa análise de como o inseto *Eucryp torrhynchus brandti* realiza o *self-righting*. São identificados três estados principais: apenas as costas ficam em contato com o solo, uma vez que o corpo é muito maior que as pernas; o inseto reposiciona o corpo ficando oblíquo em relação ao solo, de maneira que algumas patas tocam o chão; e o estado quando as patas são divididas em dois grupo: as patas em contato e as sem contato com o solo. Neste último estado, o grupo de pata que toca o solo é responsável por empurrar o corpo para cima, ao passo que as demais se agitam para gerar a força inercial capaz de realizar o rolamento no eixo longitudinal. Estas etapas podem ser visualizadas na Figura 2.

Figura 2 – Etapas observadas durante a reorientação do *Eucryp torrhynchus brandti*.



Fonte: disponível em (PENG et al., 2017).

A partir destas observações, Peng et al. (2017) propõem o *self-righting* do hexápode NOROS. A estratégia é dividida em quatro estágios: na primeira, o robô é colocado em uma

---

<sup>1</sup>do Inglês: *Degree of Freedom* (DoF)

configuração inicial; em seguida, duas patas, as mais externas, criam um eixo de rotação, enquanto as remanescentes geram força de reação com o solo; na terceira, duas das quatro patas que geravam a força de reação perdem o contato, de maneira que seu deslocamento provoca a alteração da posição do centro de massa, levando-o a um ponto crítico; por último, devido ao ponto atingido pelo centro de massa, o robô utiliza da gravidade para terminar seu giro, se reorientando. Em todas as etapas, as trajetórias das juntas são obtidas através da cinemática inversa, sendo que os efeitos dinâmicos das pernas são desconsiderados. A estratégia foi experimentada apenas em superfície plana.

Sequências de movimentos também são obtidas por Perkins, Malchano e Talebinejad (2016) para a realização do *self-righting* em robôs quadrúpedes. Ao longo do documento, os autores dividem a tarefa de SR em algumas etapas. Cada etapa é caracterizada por uma configuração cinemática inicial e final do robô. Entretanto, os autores não deixam claro qual a abordagem utilizada para definir tais posturas. Os autores ainda citam que as sequências de movimentos definidas por eles poderiam ser utilizadas pelo robô WildCat.

Saranli, Rizzi e Koditschek (2004) propõem uma metodologia de controle que possibilita o robô RHex, visto na Figura 3, se reorientar usando as forças geradas pelas patas. A formulação de Newton-Euler é utilizada para realizar a análise de corpo livre em três situações distintas, considerando as restrições que o ambiente impõe no sistema. Em cada um dos cenários, são estabelecidos os modelos dinâmicos do momento linear e angular do robô e, dependendo da configuração em que o robô se encontra, um destes modelos é escolhido para computo do sinal de controle. Como o torque aplicável pelas pernas na realização de movimentos é limitado às capacidades dos atuadores utilizados, os autores propõem uma expressão que restringe os sinais de controle. Assim, um problema de otimização é realizado para obter o valor máximo de torque que respeita todas as restrições. Os autores experimentaram a proposta em seis ambientes distintos, obtendo sucesso em cinco.

Johnson et al. (2012) propõem um modo de realização do *self-righting* aéreo. Caso o robô esteja em queda livre, uma cauda reorienta o robô para atingir o solo com uma postura específica. Esta estratégia de controle é viável graças ao princípio da conservação do momento angular, a mesma adotada pelos gatos ao caírem de pé (WIEBER; TEDRAKE; KUINDERSMA, 2016). Os autores investigaram a influência das características construtivas que a cauda insere ao sistema e reflete nos modelos cinemático e dinâmico do robô. Dois ensaios foram realizados: no primeiro, o robô foi largado em queda livre, com a parte frontal para baixo, de uma altura

Figura 3 – Plataforma robótica HReX da *Bosnton Dynamics*.



Fonte: disponível em (DYNAMICS, 2020).

de 1,36 m do solo; já no segundo, o robô realizava uma caminhada, em linha reta, em direção a um vão de 0,62 m de altura. Em ambos os casos, a inclusão da cauda possibilitou que o robô atingisse o solo com a orientação desejada.

Casarez e Fearing (2017) também propõem a implementação de uma cauda rígida, construída usando uma haste de fibra de carbono, para realizar o *self-righting* de um robô semelhante a uma barata. Inicialmente, foi realizada a obtenção de um modelo dinâmico que estabelece a influência da cauda nos momentos linear e angular do sistema robótico. A partir do modelo, foi efetuado uma investigação de como características construtivas da cauda e as condições físicas do ambiente interferem no torque necessário para rotacionar o robô em seu próprio eixo. Fundamentados nestes resultados, o atuador foi definido e simulações foram realizadas para estabelecer o tempo de SR. Para o controle da posição angular da cauda, um Controlador Proporcional-Integral-Derivativo (PID) foi proposto. Dois ensaios foram executados: no primeiro, o *self-righting* é realizado por meio de uma sequência de movimentos, em malha aberta, da cauda; no segundo ensaio, o robô caminha em um ambiente com diversos obstáculos e degraus e, ao detectar que está com sua orientação invertida, o algoritmo de controle parametriza, com um determinado período e amplitude, a referência da trajetória da haste em relação à posição da cauda e da base. No primeiro caso o desempenho foi verificado em sete superfícies distintas, obtendo sucesso em seis, e nas condições do segundo caso o robô foi capaz de se reorientar.

Nos trabalhos de Li et al. (2016) e Li et al. (2017), é proposta uma solução para o SR de robôs semelhantes às baratas, com atuadores análogos às carapaças que envolvem suas asas. Utilizando uma metodologia semelhante a Peng et al. (2017), os autores observaram,

durante a realização do SR do inseto, dois comportamentos distintos: um simétrico, quando as duas carapaças são abertas em um mesmo intervalo de tempo; e um assimétrico, em que existe uma pequena diferença nos tempos de aberturas. Ainda, análises dos efeitos da geometria e extensão dos movimentos dos atuadores foram realizadas para avaliar a taxa de sucesso e questões energéticas da tarefa. Um modelo dinâmico planar de corpo rígido foi obtido, entretanto, questões do controle da tarefa não foram abordados nos trabalhos.

Semini et al. (2015), descreve o desenvolvimento do quadrúpede HyQ2Max, que é o modelo de robô utilizado nesta dissertação. As juntas do robô são construídas com atuadores hidráulicos e controladores em torque, sendo que a impedância das juntas é determinada por Controladores Proporcional-Derivativo (PD). Nesse trabalho, o *self-righting* do robô é realizado utilizando uma máquina de estados temporais contendo, cada etapa, uma duração e valores finais desejados das juntas. Os valores desejados das juntas foram determinados de forma empírica, utilizando simulações. Durante a transição entre duas posições de juntas, um filtro realiza a suavização entre elas, gerando uma referência para o Controlador PD. Na Figura 4 são mostradas as sequências das etapas do SR.

Figura 4 – Etapas do *self-righting* do HyQ2Max.



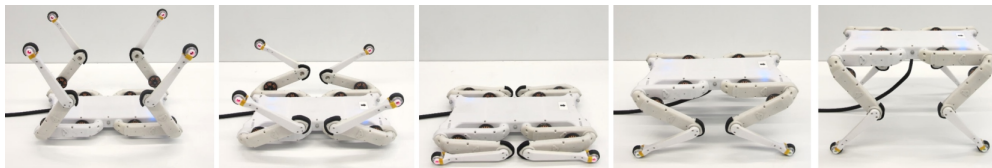
Fonte: disponível em (SEMINI et al., 2015).

Lee, Hwangbo e Hutter (2019), propõem uma estratégia baseada em *model-free Deep Reinforcement Learning* com o objetivo de controlar as manobras de reconfiguração, usando um controlador hierárquico baseado em comportamento. Nesta abordagem, um seletor escolhe a atitude mais adequada dependendo das observações mais recentes do sistema. São três atitudes possíveis: locomoção, levantar-se e *self-righting*. Cada um dos comportamentos anteriores são obtidos através de treinamentos individuais usando *Reinforcement Learning*. Ao final, posições

de juntas são obtidas e usadas como referência do Controlador PD do robô. Esta estratégia é aplicada no robô *ANYmal*. Destaca-se que o experimento foi conduzido em chão plano.

No trabalho de Grimminger et al. (2020), também é descrito o desenvolvimento de um robô quadrúpede. No modelo proposto, a capacidade de locomoção não é afetada caso o robô tombe, uma vez que ele é simétrico em relação ao plano xy. Caso o robô caia com as costas no chão, não é necessário que ele rotacione o torso. As etapas de realização do SR são mostradas na Figura 5.

Figura 5 – Etapas da realização do *self-righting* de um robô simétrico.



Fonte: disponível em (GRIMMINGER et al., 2020).

Para que tal abordagem funcione, é necessário que as juntas tenham uma grande extensão de movimento, de modo que os valores das referências são tratados internamente no software.

Embora todos estes trabalhos acima tratem do *self-righting* para robôs com pernas, esta tarefa não é exclusividade deles. Nos trabalhos de Tsukagoshi et al. (2003) e Stoeter e Papanikolopoulos (2006), a tarefa de SR é resolvida para configurações de robôs com rodas. Estas pesquisas demonstram a amplitude do problema principal desta dissertação, porém, delimitando o assunto, trata-se nesta dissertação apenas de robôs com pernas.

É possível constatar que, na maioria das pesquisas, que propõem soluções de *self-righting*, algum tipo de máquina de estados foi utilizada para determinar qual ação de controle deve ser tomada durante a tarefa. As informações obtidas deste levantamento do estado da arte, pertinentes ao problema desta dissertação, estão sumarizadas na Tabela 1.

Na Tabela 1, é possível observar que para os robôs que possuem pernas com apenas um grau de liberdade, com exceção de Saranli, Rizzi e Koditschek (2004), as soluções de SR foram obtidas através da adição de algum mecanismo extra, demonstrando ser uma maneira simples e eficaz de realização de tal tarefa. Em relação aos atuadores especializados, salvo em Li et al. (2017), todos exibem apenas um grau de liberdade, facilitando a sua construção mecânica. Entretanto, para robôs mais complexos, como os quadrúpedes, a adição de um atuador exclusivamente para realizar a tarefa não é viável, uma vez que tais sistemas robóticos já possuem um grande número mecanismos para atuação.

Tabela 1 – Quadro resumo do estado da arte.

Trabalho	Patas / DoF	Atuador esp.	Modelo	Arq. de controle
Zhang et al. (2012)	2/1	S	C + D	-
Peng et al. (2017)	6/3	N	C	ME
Perkins, Malchano e Talebinejad (2016)	4/3	N	-	ME
Saranli, Rizzi e Koditschek (2004)	6/1	N	D*	ME + PD
Johnson et al. (2012)	6/1	S	D*	-
Casarez e Fearing (2017)	6/1	S	D*	ME + PID
Li et al. (2017)	6/1	S	C	-
Semini et al. (2015)	4/3	N	C	ME + PD
Lee, Hwangbo e Hutter (2019)	4/3	N	RL	ME + PD
Grimminger et al. (2020)	4/3	N	-	-

C: modelo cinemático; D: modelo dinâmico; D\*: modelo dinâmico baseado em momento linear e angular; RL: *reinforcement learning*; ME: máquina de estados.

Fonte: próprio autor

Para robôs que possuem pernas com mais de um grau de liberdade, é possível observar que todas as estratégias de *self-righting* contêm pelo menos três etapas em comum. Considerando inicialmente que o robô está com as costas no chão, na primeira etapa é realizado uma preparação, em que algumas das patas deixam o robô oblíquo ao plano. Na segunda etapa, as patas geram uma força para que o robô role em seu eixo longitudinal. Por último, com o robô já de barriga para baixo, o sistema apenas se levanta, recuperando a sua capacidade de locomoção. Esta última ação é definida como movimento de *stand up*.

As soluções descritas até aqui, apesar de serem funcionais, são muito específicas. Com exceção de Casarez e Fearing (2017) e Saranli, Rizzi e Koditschek (2004), todas as abordagens são realizadas em superfícies planas e ambientes estruturados, o que gera alguns questionamentos: em um terreno com características diferentes, o robô seria capaz de se reorientar? É garantido que durante uma tentativa de *self-righting*, as posturas de referência são factíveis de serem realizadas?

Durante a realização do *self-righting*, características físicas do ambiente podem ser limitantes para o sucesso da tarefa. Dependendo da situação, as forças geradas nos contatos são insuficiente para garantir que o robô se levante, gerando um escorregamento, levando-o a uma nova queda. Desta forma, é necessário considerar as forças de reação entre o solo e a estrutura do robô na realização do SR, e como foi possível averiguar na revisão bibliográfica, para os robôs quadrúpedes, não há uma solução que considere este fator. Outro ponto a ser verificado,



é que em certas situações o ambiente pode proporcionar restrições cinemáticas, de modo que as referências providas pelas máquinas de estados são inatingíveis, obtendo o insucesso da tarefa.

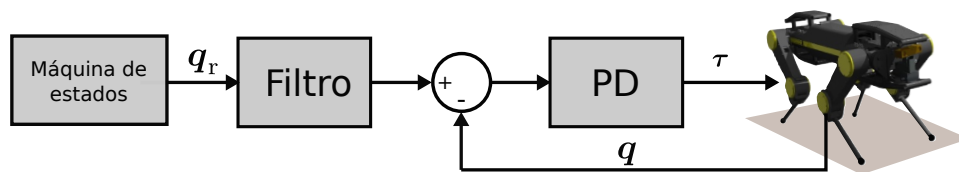
Posto isso, fica evidente a necessidade de conhecer como o ambiente influencia no robô: se insere alguma restrição ao longo do movimento que impossibilite a realização da tarefa ou até mesmo possa contribuir na sua execução. Assim, a solução deve ser capaz de fechar a malha de controle da tarefa, ou seja, ela deve encontrar uma sinal de controle que seja capaz de realizar o SR levando em consideração as limitações impostas pela interação entre o robô e o ambiente.

O *self-righting* de robôs que possuem simetria horizontal é simplificado, pois duas das três etapas, preparação e o rolamento, deixam de existir. Todavia, esta característica deve ser considerada desde o início do projeto mecânico, o que obviamente inviabiliza a sua realização em sistemas já existentes. Ainda mais, tal alternativa não exclui os problemas da influência do ambiente.

## 2.1 PROBLEMA DO *SELF-RIGHTING* DO HYQ2MAX

A partir da revisão bibliográfica é possível elencar alguns problemas que existem nas soluções atuais de *self-righting* de robôs quadrúpedes. Utilizando como base a abordagem descrita em Semini et al. (2015), o diagrama de blocos do *loop* de controle do SR é mostrada de forma detalhada na Figura 6:

Figura 6 – Estrutura de controle da tarefa de *self-righting* do HyQ2Max.



Fonte: próprio autor.

Examinando a Figura 6, observa-se que a partir do momento que o HyQ2Max inicia o *self-righting*, na arquitetura original, não há um *feedback* dos estados internos do robô. Assim, caso ocorra deslizamento em algum dos pés, a máquina de estados não interfere, e continua a fornecer as mesmas referências ao controlador interno, o que pode levar o robô a uma nova queda, e consequentemente provocar danos na estrutura mecânica ou no próprio ambiente.

Essa estrutura também não leva em consideração características do ambiente. Como as posturas de referência foram obtidas de forma empírica em simulações para uma determinado

terreno, qualquer mudança física no ambiente pode impossibilitar a realização da tarefa, uma vez que a estrutura não consegue tratar da interferência do ambiente na tarefa.

Outro problema é em relação aos atuadores: a solução original não trata de forma direta de seus limites. Saturar o sinal gerado pelo Controlador PD pode inserir não linearidades no sistema, e provocar a instabilidade deste, o que não é desejado.

## 2.2 PROPOSTA DE ARQUITETURA PARA O *SELF-RIGHTING*

Como já mencionado na revisão bibliográfica, a tarefa de *self-righting* pode ser dividida em três principais etapas, quando o robô está com as costas no chão: preparação; rolamento; e o *stand up*. Posto isso, esta dissertação tem o objetivo de propor uma arquitetura de controle para a terceira etapa do *self-righting*, uma vez que é comum tanto para robôs simétricos, quanto assimétricos.

A tarefa de *stand up* consiste no movimento realizado pelo robô para que este fique de pé novamente. Durante esta etapa, ocorre apenas movimentos lineares, uma vez que a orientação do robô já está correta. O sistema robótico parte de uma postura em que está com a parte inferior da dorso em contato com o chão, e as pernas recolhidas. Já na configuração final, as pernas estão estendidas, posicionando o centro de massa em uma certa altura. Durante o movimento os pés não devem se mover, pois dependendo das condições do terreno, qualquer descolamento dos pés pode levar o robô a uma nova queda.

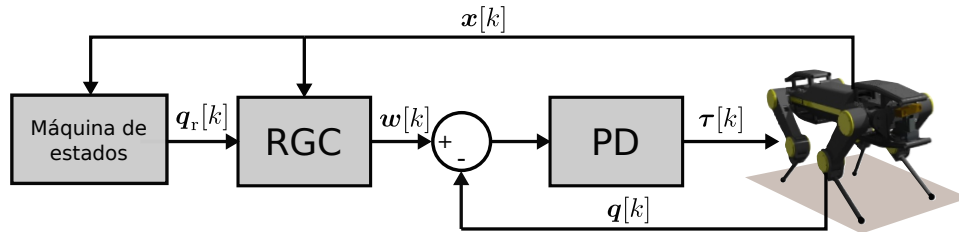
A partir das considerações realizadas na Seção 2.1, uma possível solução para atribuir mais robustez e segurança na realização do *stand up* é incluir no arranjo um algoritmo seja capaz de modificar as referências que a máquina de estados, levando em consideração os estados internos do robô e as características físicas do ambiente em que está inserido. Desta maneira, ao incluir esta nova estrutura, é possível controlar, e consequentemente atribuir maior robustez na realização da tarefa.

Uma solução viável é a inserção de uma estrutura baseada em controladores hierárquicos, que são uma classe de arquiteturas de controle que contém dois ou mais níveis de controladores, de modo que cada controlador envia um sinal de controle para o malha mais interna e ao mesmo tempo pode gerar um *feedback* de um sinal para o nível acima (PARKER, 2003). Uma de suas possíveis implementações é a partir do *Reference Governor Control* (RGC).

O RGC é um algoritmo que é capaz de modificar a referência de um sistema em malha fechada, garantindo desempenho, além de respeitar determinadas restrições (KALABIĆ et al.,

2014). Na Figura 7 está ilustrado, em formato de diagrama de blocos, a solução do problema de *self-righting* do HyQ2Max, proposta por esta dissertação.

Figura 7 – Estrutura de controle proposta para a tarefa de *self-righting* do HyQ2Max.



Fonte: próprio autor.

A técnica de RGC, descrita em detalhes no Capítulo 3, apoia-se na dinâmica de malha fechada do controlador interno e da planta, para obter o valor ótimo da ação de controle que respeite as restrições do sistema como um todo. Uma das possibilidades de implementação desta arquitetura é utilizando Controle Preditivo Baseado em Modelo.

O Controle Preditivo Baseado em Modelo é um tipo de arquitetura de controle que utiliza o modelo dinâmico do processo em um problema de otimização para obter o valor ótimo da ação de controle. Outra característica bastante interessante que é encontrada nas implementações deste tipo de controlador, é a possibilidade de lidar com restrições do sistema (CAMACHO; BORDONS, 2007), que é uma característica desejável na solução do problema de *self-righting*.

Mas, por que empregar uma arquitetura que modifica as referências da malha do atuador interno, e não um controlador preditivo que designe diretamente o torque? A resposta é: a dinâmica do torque em sistemas robóticos com pernas é complexa, além do mais, e necessita uma alta taxa de atualização. Logo, empregar um controlador preditivo neste tipo de problema é inviável. Ademais, modificações na malha de controle do torque seriam necessárias.

Portanto, ao inserir o *Reference Governor Control*, as posturas de referências que foram obtidas em Semini et al. (2016) ainda são válidas, e podem ser utilizadas como referência de alto nível, sendo o RGC responsável por modifica-las com base nos estados do robô e das características e restrições do ambiente. Desta forma, na solução proposta, passaria a existir uma malha de controle da tarefa, garantido robustez para esta, e ainda solucionando um dos problemas das propostas atuais, que é a não consideração das informações do ambiente para a realização do SR.

Para que possa ser realizado, o *Reference Governor Control* via Controlador Preditivo Baseado em Modelo é necessário um modelo do processo dinâmico que seja suficientemente representativo. Em alguns dos trabalhos já citados, os autores avaliaram o comportamento dos momentos linear e angular durante o *self-righting*. Analisando do ponto de vista físico, poder controlar tais variáveis é interessante, uma vez que elas são responsáveis por permitirem que um corpo modifique a sua trajetória.

A afirmativa anterior fica evidente quando é observado o rolamento, que é a etapa mais crítica do *self-righting*. Para que ocorra, dependendo do robô, é necessário que um certo valor de momento linear e angular seja gerado através das forças de reação do solo, para que o centro de massa possa ser colocado em um ponto crítico, deixando com que a gravidade termine a etapa. Assim, ter algum grau de controle sobre estas grandezas, possibilita atribuir mais robustez à tarefa.

Para a realização deste trabalho, assume-se que um algoritmo de alto nível é responsável por determinar um conjunto de referências das juntas do robô para uma certa postura, não sendo o foco do trabalho obtê-las. A partir de um postura inicial, um algoritmo intermediário, o *Reference Governor Control*, fica incumbido de modificar as referências provenientes de uma máquina de estados, obtendo uma referência modulada que é usada no Controlador PD, de modo que este sinal respeita as restrições do sistema ao longo do movimento. Ao final da tarefa, é desejado que a postura obtida seja a mais próxima possível da referência de alto nível. Para isso, no Capítulo 3 todos os detalhes teóricos referente aos aspectos de controle e modelagem de sistemas robóticos são discutidos, para que uma solução de controle de tarefa seja encontrada.

Por último, destaca-se que o autor desta dissertação não encontrou na literatura trabalhos que abordem a solução do *self-righting* através do uso da arquitetura de *Reference Governor Control*, tornando um solução inédita para o problema.

### 3 ARCABOUÇO TEÓRICO

Todo o embasamento teórico utilizado neste trabalho está descrito neste capítulo: na Seção 3.1 é discorrido sobre a arquitetura *Reference Governor Control* e suas formulações; na Seção 3.2, são detalhadas as principais características e a formulação do Controle Preditivo Baseado em Modelo; e, por último, a modelagem de sistemas robóticos é detalhada na Seção 3.3.

#### 3.1 REFERENCE GOVERNOR CONTROL

Como mencionado no Capítulo 2, o *Reference Governor Control* é um tipo de estrutura inserida para modificar uma referência de alto nível, gerando uma referência modulada que é aplicada ao controlador interno da planta. A vantagem de utilizar um RGC pode ser observada ao considerar duas classes de problemas que são bastante comuns se tratando de soluções de controle: plantas não lineares e/ou instáveis em malha aberta, onde é desejado utilizar algum controlador que utiliza de otimização para a obtenção do sinal de controle; e sistemas simples, que já contém algum tipo de controlador interno.

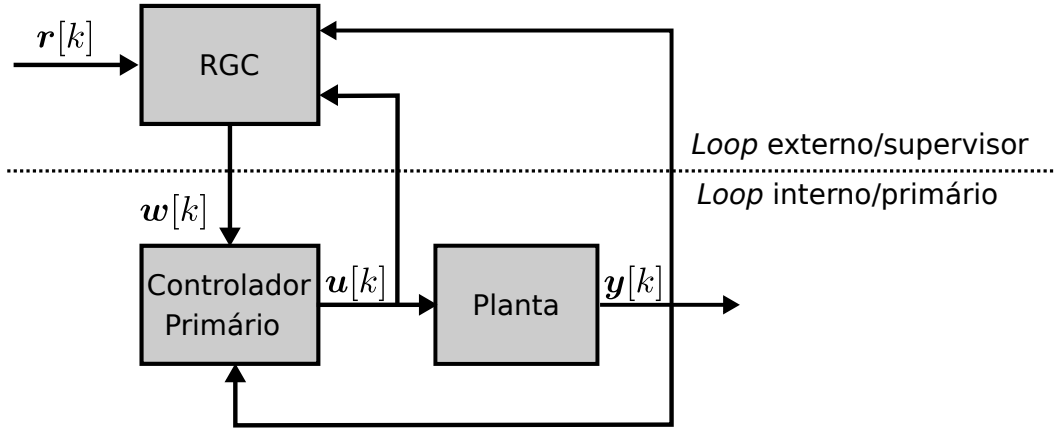
Para o primeiro tipo, é possível que o problema de otimização não consiga encontrar uma solução ótima em tempo hábil, durante o tempo de amostragem do sistema, levando-o a situações de instabilidade. Para contornar este problema, um controlador mais simples pode ser inserido para estabilizar a planta. Desta maneira, o RGC apenas obtém uma referência, enquanto a malha interna garante o seu rastreo (BEMPORAD, 1998).

Na segunda situação, como todo atuador possui limites de operação, as ações de controle podem saturar em algumas condições. Logo, se o *setpoint*  $\mathbf{r}$  varia com o passar do tempo, com este tipo de estrutura básica não é possível garantir que o sistema permanecerá estável. Logo, ao inserir o RGC, considerações acerca das restrições são empregadas na sua implementação, e, portanto, garantindo que a planta opere de forma segura (DO; CHOI, 2005).

Estes dois casos podem ser exemplificados no diagrama de blocos da Figura 8. O controlador interno gera um sinal de controle  $\mathbf{u}$  que é diretamente aplicado à planta. Para controladores do tipo PID, por exemplo, este sinal de controle é obtido através do erro entre a saída  $\mathbf{y}$  da planta com o *setpoint*  $\mathbf{w}$ . A obtenção de  $\mathbf{w}$  é realizada por algum algoritmo, a partir do conhecimento dos estados do *loop* interno. Assim,  $\mathbf{r}$  é modificado ao longo do tempo para que o sistema possa melhorar o desempenho do processo, além de garantir a operação segura

(KLAUČO; KVASNICA, 2019).

Figura 8 – Estrutura genérica do *Reference Governor Control*.



Fonte: adaptado de (KLAUČO; KVASNICA, 2019).

### 3.1.1 Abordagens de implementação

Técnicas que retratam a utilização deste tipo de controlador são descritas desde o começo dos anos 1990. Bemporad e Mosca (1994), propõem um método para que o sinal  $w$  seja obtido através da suavização da referência  $r$ . A cada instante  $k$ , um algoritmo recursivo obtém o valor de uma constante  $\lambda$ , e a partir dessa, é gerado uma sequência finita de ações de controle  $w$ . Esta sequência é aplicada no modelo dinâmico da planta para se observar se as restrições são respeitadas e se o valor do tempo de assentamento é minimizado. Em caso positivo, a primeira ação de controle é aplicada à planta, caso contrário o valor de  $\lambda$  é atualizado e uma nova sequência é recalculada.

Gilbert, Kolmanovsky e Tan (1995) propõem uma abordagem semelhante ao trabalho de Bemporad e Mosca (1994) para plantas não lineares. Para garantir a estabilidade de um sistema com uma única entrada e saída (SISO)<sup>1</sup>, em malha fechada sujeito às restrições, um problema de otimização é realizado para determinar o valor de  $\beta$ , que é capaz de modificar a referência  $w$ , obedecendo a condição para que os estados e a ação de controle pertençam à  $\tilde{\mathcal{O}}_\infty$ . Na otimização,  $\tilde{\mathcal{O}}_\infty$  é o conjunto ligeiramente menor da máxima saída admissível que respeite as restrições impostas à planta. A obtenção de tal conjunto pode ser realizada de forma *offline*

<sup>1</sup>do Inglês: *Single Input-Single Output* (SISO)

ou *online*, sendo que tais procedimentos são detalhados no trabalho de Gilbert, Kolmanovsky e Tan (1995).

A partir do trabalho de Gilbert, Kolmanovsky e Tan (1995), Bemporad (1998) propõe que o sinal  $\mathbf{w}$  seja obtido utilizando o mesmo problema de otimização. Na nova abordagem, é verificado se as restrições são respeitadas em um horizonte finito, e não apenas no instante de amostragem, incorporando a estratégia retrocedente do Controlador Preditivo Baseado em Modelo.

As abordagens de Gilbert, Kolmanovsky e Tan (1995) e Bemporad (1998) são comumente empregadas na obtenção *Reference Governor Control* em diversos tipos de sistemas, destacam-se: o controle de pendulo invertido (CASAVOLA; MOSCA; PAPINI, 2004); o controle de manobras de uma nave espacial (KALABIĆ et al., 2014); o controle de manobras de aviões (SIMON; HÄRKEGÅRD; LÖFBERG, 2017); a utilização em sistemas eletro-mecânicos de automóveis (GARONE; CAIRANO; KOLMANOVSKY, 2017); e na prevenção de capotamento de veículos (BENCATEL et al., 2017).

Do e Choi (2005) descrevem uma abordagem para sistemas já em malha fechada. Uma estratégia é concebida para encontrar o valor da variável  $\lambda^*$ , capaz de modificar a referência  $\mathbf{r}$  de modo que a ação de controle  $\mathbf{u}$  não sature, evitando a perda de estabilidade do sistema.

Observa-se que as abordagens de Bemporad e Mosca (1994), Gilbert, Kolmanovsky e Tan (1995), Bemporad (1998) e Do e Choi (2005) são utilizadas para encontrar o valor de uma variável para escalonar o valor da referência. Ao contrário destes trabalhos, Klaučo e Kvasnica (2019) propõem algumas técnicas formuladas diretamente a partir do Controlador Preditivo Baseado em Modelo, para sistemas que possui algum tipo de controlador interno ao sistema.

Nos trabalhos de Klaučo e Kvasnica (2017) e Klaučo, Kalúz e Kvasnica (2017) é estabelecido uma metodologia para a formulação do RGC em sistemas que contém algum tipo de *loop* de controle interno, usando um controlador clássico. A realização é feita de tal forma que é estabelecido o espaço de estados aumentado:

$$\tilde{\mathbf{x}}[k+1] = \mathbf{A}_{\text{CL}}\tilde{\mathbf{x}}[k] + \mathbf{B}_{\text{CL}}\mathbf{w}[k]; \quad (3.1a)$$

$$\mathbf{u}[k] = \mathbf{C}_{\text{CL},u}\tilde{\mathbf{x}}[k] + \mathbf{D}_{\text{CL},u}\mathbf{w}[k]; \quad (3.1b)$$

$$\mathbf{y}[k] = \mathbf{C}_{\text{CL},y}\tilde{\mathbf{x}}[k] + \mathbf{D}_{\text{CL},y}\mathbf{w}[k]. \quad (3.1c)$$

A Equação (3.1a) estabelece o comportamento da dinâmica do sistema em malha fe-

chada, em que  $\tilde{\mathbf{x}}$  é o vetor de estados, que considera tanto os estados internos do controlador, quanto os estados da planta. Já a Equação (3.1b) é a saída dos controladores, enquanto a Equação (3.1c) é a saída da planta. A partir da Equação (3.1), são estabelecidos os modelos de predição conforme mostrado na Equação (3.3), e um problema de otimização pode ser definido de forma semelhante ao mostrado na Equação (3.5). Utilizando esta abordagem, é necessário a inclusão de observadores para estimar os estados internos dos controladores internos, e se necessário, da planta. O processo de obtenção das matrizes do modelo visto na Equação (3.1) é descrito de forma detalhada no Apêndice A.

A abordagem exposta é utilizada por Klaučo e Kvasnica (2017) para obter o *Reference Governor Control* de controladores Proporcional-Integral (PI) de um *boiler* sujeito à restrições. Já no trabalho de Klaučo, Kalúz e Kvasnica (2017), a mesma solução é realizada para a obtenção do RGC para o controle de sistema de levitação magnética. A diferença é que neste segundo trabalho, o controlador preditivo é realizado de forma explícita.

Observando as abordagens acima descritas, constata-se que a implementação proposta por Klaučo e Kvasnica (2017) possui um potencial de ser utilizada na solução do problema de controle da tarefa de SR, uma vez que o robô HyQ2Max já contém uma malha de controle das posições das juntas, semelhante à malha de controle do *boiler*. Além do mais, ao implementar o RGC usando controlador preditivo, as restrições do mecânicas do robô e do ambiente podem ser levadas em consideração na modulação da referência do controlador interno. As questões acerca da implementação estão descritas no Capítulo 4.

### 3.2 CONTROLE PREDITIVO BASEADO EM MODELO

Controle Preditivo Baseado em Modelo, ou simplesmente MPC<sup>2</sup>, ao contrário do que o próprio nome sugere, não designa uma estratégia específica, mas sim um conjunto de métodos que utilizam explicitamente um modelo da planta na obtenção do sinal de controle. Três conceitos aparecem em todas as famílias de controladores preditivos baseados em modelo: uso explícito do modelo da planta para prever as saídas futuras do processo; cálculo de uma sequência de ações de controle que minimiza uma função custo; e estratégia retrocedente (CAMACHO; BORDONS, 2007).

O MPC foi concebido com o intuito de atender às demandas de controle de usinas e refinarias de petróleo (QIN; BADGWELL, 2003). Para Maciejowski (2002), esta é a única

---

<sup>2</sup>do Inglês: *Model Predictive Control* (MPC)



metodologia de controle avançada que trouxe um impacto significativo nestes setores, pautada nas seguintes razões: capacidade de lidar naturalmente com sistemas multivariáveis; considerar as limitações dos atuadores; possibilidade de operar próximo aos limites das restrições; e taxa de atualização de controle relativamente baixa.

Historicamente, o *Model Predictive Heuristic Control* (MPHC), descrito por Richalet et al. (1978), é considerado o primeiro controlador preditivo. A sua realização é obtida ao descrever um processo multivariável usando o modelo identificado através de um impulso finito. A ação de controle é calculada de forma heurística, em que as futuras ações são calculadas de tal forma que, quando aplicadas ao modelo preditivo, induzem as saídas ao mais próximo possível da trajetória de referência desejada.

A *Shell Oil*, no início dos anos setenta, desenvolveu sua própria tecnologia de MPC (QIN; BADGWELL, 2003), dando origem ao *Dynamic Matrix Control* (DMC), patenteado pela própria empresa (PRETT; RAMAKER; CUTLER, 1982). O DMC utiliza a resposta do sistema ao degrau para obter o modelo de predição da planta. A partir do modelo de predição, uma função custo quadrática é minimizada em um dado horizonte de controle e predição, obtendo um ganho. Este ganho é usado para gerar o sinal de controle que é capaz de levar o sistema cada vez mais próximo da referência. Entretanto, a formulação inicial não considerava restrições.

Para tratar de restrições, anos mais tarde, os engenheiros da própria *Shell* incorporaram o algoritmo do DMC a um problema quadrático, em que as restrições de saídas e entradas apareciam explicitamente (QIN; BADGWELL, 2003). Assim, a cada instante de amostragem, um problema de otimização era resolvido dentro de um horizonte de predição e controle, de modo a obter um sinal de controle ótimo. Um inconveniente é que dependendo da ordem do sistema e do algoritmo usado na solução da otimização, o problema pode se tornar inviável computacionalmente.

Apesar destas técnicas de MPC serem atraentes para aplicação em soluções práticas, elas possuem uma desvantagem: a incapacidade de trabalhar explicitamente com incertezas (KOTHARE; BALAKRISHNAN; MORARI, 1996). Como o objetivo é minimizar uma função que depende de um único modelo linear, não variante no tempo, para prever os estados futuros, quando incertezas são atribuídas ao sistema o desempenho do controlador fica comprometido. Neste sentido, os autores supracitados, propõem uma abordagem em que desigualdades matriciais são resolvidas a cada instante, garantindo a robustez do controlador. Nesta abordagem

também são consideradas as restrições da planta.

Como já mencionado, dependendo do tamanho e da dinâmica da planta, não é possível atender os requisitos de tempo real, sem que aumente consideravelmente a capacidade computacional que abriga o MPC. Nesse sentido, de forma a reduzir a necessidade de alta quantidade de processamento em tempo de execução, como resultado direto do trabalho de Kothare, Balakrishnan e Morari (1996), no trabalho de Wan e Kothare (2003), o controlador é obtido em um procedimento *offline*, gerando um vetor de ganhos, que são aplicados na planta de acordo com o estado atual. Uma segunda opção é a utilização de um MPC Explícito (BEMPORAD et al., 2002), que trata o problema de forma multiparamétrica em um processo *offline*, obtendo um conjunto de regiões delimitadas pelos valores dos estados. Em cada uma destas regiões, há uma lei de controle descrita em função dos estados. Na etapa *online*, um algoritmo de busca é utilizado para encontrar uma solução de controle dado o estado atual do sistema.

As técnicas descritas até aqui são utilizadas em sistemas linearizados, e que tenham o seu comportamento bem definido. Entretanto, existem alguns sistemas que estas soluções não exibem um desempenho aceitável ou até se tornam inviáveis, necessitando que alguma técnica não linear seja adotada (BLET et al., 2002). Tais técnicas não serão discutidas neste trabalho, todavia, no trabalho de Magni, Raimondo e Allgöwer (2009), é realizado um compilado de soluções utilizando estas abordagens, bem como perspectivas futuras no que tange este tipo de problema.

Como pode ser constatado com este histórico, inicialmente, a técnica de controle preditivo era exclusividade em plantas classificadas como processos de dinâmica lenta, pois eram facilmente alocáveis nos sistemas computacionais comuns da época. Entretanto, com o passar do tempo, o avanço de técnicas e popularização de sistemas computacionais mais potentes, os algoritmos de MPC passam a ser utilizados em processos ditos de dinâmica rápida (LEE, 2011). Nesta linha, este trabalho traz resultados de MPC em um problema intrínseco a robôs móveis com pernas, que se caracterizam como sistemas de dinâmica rápidas.

### 3.2.1 Fundamentação teórica do MPC

Como já mencionado, a implementação do MPC necessita de um modelo que descreva o comportamento da dinâmica da planta. Uma das possíveis formas de realização do modelo é utilizando a formulação por espaço de estados, que permitem a representação de sistemas com uma ou mais variáveis, além de possibilitar a extrapolação para sistemas não lineares

(CAMACHO; BORDONS, 2007).

Considerando o modelo de espaço de estados discreto:

$$\mathbf{x}[k+1] = \mathbf{A}\mathbf{x}[k] + \mathbf{B}\mathbf{u}[k], \quad (3.2a)$$

$$\mathbf{y}[k] = \mathbf{C}\mathbf{x}[k] + \mathbf{D}\mathbf{u}[k]. \quad (3.2b)$$

A Equação (3.2a) é conhecida como equação dinâmica do espaço de estados,  $\mathbf{x} \in \mathbb{R}^n$  é o vetor de estados,  $\mathbf{A} \in \mathbb{R}^{n \times n}$  é matriz das dinâmicas do sistema,  $\mathbf{B} \in \mathbb{R}^{n \times p}$  é a matriz de entrada do sistema e  $\mathbf{u} \in \mathbb{R}^p$  é o vetor de entradas de controle. Já a Equação (3.2b) é denominada equação de saída,  $\mathbf{y} \in \mathbb{R}^q$  é o vetor de saídas do sistema,  $\mathbf{C} \in \mathbb{R}^{q \times n}$  é a matriz de saída e  $\mathbf{D} \in \mathbb{R}^{q \times p}$  é a matriz de transmissão direta.

Como descrito por Rossiter (2017), para descrever a evolução do sistema em um horizonte de predição  $N$  e de controle  $M$ , com  $M \leq N$ , é possível reescrever o sistema da Equação (3.2) como:

$$\bar{\mathbf{y}} = \mathbf{G}\bar{\mathbf{u}} + \mathbf{\Phi}\mathbf{x}[k|k], \quad (3.3)$$

sendo  $\bar{\mathbf{y}} \in \mathbb{R}^{qN}$  o vetor de saídas preditas,  $\mathbf{G} \in \mathbb{R}^{qN \times pM}$  uma matriz triangular inferior, responsável por descrever o comportamento do sistema ao longo do horizonte de predição,  $\bar{\mathbf{u}} \in \mathbb{R}^{pM}$  o vetor de entradas preditas do sistema,  $\mathbf{\Phi} \in \mathbb{R}^{qN \times n}$  a matriz de resposta livre e  $\mathbf{x}[k|k]$  são os

estados da planta no instante  $k$ . Os termos  $\bar{\mathbf{y}}$ ,  $\mathbf{G}$ ,  $\bar{\mathbf{u}}$  e  $\Phi$  são descritos como:

$$\bar{\mathbf{y}} = \begin{bmatrix} \mathbf{y}[k+1|k] \\ \mathbf{y}[k+2|k] \\ \vdots \\ \mathbf{y}[k+N|k] \end{bmatrix}, \quad (3.4a)$$

$$\mathbf{G} = \begin{bmatrix} \mathbf{CB} & \mathbf{0}_{q \times p} & \dots & \mathbf{0}_{q \times p} \\ \mathbf{CAB} & \mathbf{CB} & \dots & \mathbf{0}_{q \times p} \\ \vdots & \vdots & \ddots & \mathbf{0}_{q \times p} \\ \mathbf{CA}^{(N-1)}\mathbf{B} & \mathbf{CA}^{(N-2)}\mathbf{B} & \dots & \mathbf{CA}^{(N-M)}\mathbf{B} \end{bmatrix}, \quad (3.4b)$$

$$\bar{\mathbf{u}} = \begin{bmatrix} \mathbf{u}[k|k] \\ \mathbf{u}[k+1|k] \\ \vdots \\ \mathbf{u}[k+M|k] \end{bmatrix}, \quad (3.4c)$$

$$\Phi = \begin{bmatrix} \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^N \end{bmatrix}. \quad (3.4d)$$

Nas Equações (3.4a) e (3.4c) os índices  $[k+i|k]$ , com  $i = 0, 1, 2, \dots, N$ , significam os valores das variáveis preditas no instante  $k+i$  dado os valores medidos/estimados no instante  $k$ .

Para um problema de regulação, a partir do modelo de predição, a ação de controle é calculada através da solução de um problema de otimização:

$$\begin{aligned} \min_{\bar{\mathbf{u}}} \quad & \Gamma(\bar{\mathbf{y}}, \bar{\mathbf{u}}) \\ \text{s.t.} \quad & \bar{\mathbf{y}} = \mathbf{G}\bar{\mathbf{u}} + \Phi \mathbf{x}[k|k] \end{aligned}, \quad (3.5)$$

em que  $\bar{\mathbf{u}}$  é a variável a ser otimizada,  $\Gamma(\bar{\mathbf{y}}, \bar{\mathbf{u}})$  é uma função de ponderação quadrática, definida como:

$$\Gamma(\bar{\mathbf{y}}, \bar{\mathbf{u}}) = \|\bar{\mathbf{y}}\|_{\mathbf{Q}} + \|\bar{\mathbf{u}}\|_{\mathbf{R}}, \quad (3.6)$$

em que:

$$\|\bar{\mathbf{y}}\|_{\mathbf{Q}} = \bar{\mathbf{y}}^T \mathbf{Q} \bar{\mathbf{y}}, \quad (3.7)$$

e:

$$\|\bar{\mathbf{u}}\|_{\mathbf{R}} = \bar{\mathbf{u}}^T \mathbf{R} \bar{\mathbf{u}}, \quad (3.8)$$

são os termos da função quadrática que penalizam a saída do sistema e a ação de controle, respectivamente. O termo  $\mathbf{Q}$  é uma matriz simétrica, semi-definida positiva, que pondera os termos das saídas do sistema, ao passo que  $\mathbf{R}$  é a matriz simétrica, positiva definida, que pondera as ações de controle. Destaca-se que qualquer função de ponderação pode ser utilizada, desde que seja dependente da variável a ser otimizada.

Devido ao conceito de horizonte retrocedente, apenas os primeiros  $p$  valores do vetor  $\bar{\mathbf{u}}$  são usados como ação de controle, ou seja, os valores de  $\mathbf{u}[k|k]$ .

Apesar da Equação (3.5) fornecer uma ação de controle que obedece algum critério minimizado pela função custo  $\Gamma$ , é possível incluir restrições na sua solução. Na prática, todo e qualquer processo possui restrições, sejam elas operacionais ou físicas (CAMACHO; BORDONS, 2007).

Restrições físicas são aquelas que surgem devido às condições impostas pela planta, como máximo *range* do atuador ou o seu *slew-rate*, ao passo que as restrições de operação são, por exemplo, valores de segurança na qual a planta opera (CAMACHO; BORDONS, 2007).

É possível considerar na formulação de um MPC, uma sistemática que permite lidar com restrições da melhor forma possível (ROSSITER, 2017). Para isso, as restrições devem ser escritas deixando explícito as variáveis de otimização:

$$\mathbf{b}_{\min} \leq \mathbf{A}_{\text{in}} \bar{\mathbf{u}} \leq \mathbf{b}_{\max}, \quad (3.9)$$

de modo que o problema seja tratado posteriormente no otimizador.

Na Equação (3.9),  $\mathbf{A}_{\text{in}}$  descreve como as variáveis de otimização  $\bar{\mathbf{u}}$  afetam determinada característica do sistema, sendo que esta possui  $\mathbf{b}_{\max}$  e  $\mathbf{b}_{\min}$  como seus valores máximos e mínimos, respectivamente.

Assim, o problema de otimização da Equação (3.5) pode ser reescrito como:

$$\begin{aligned} \min_{\bar{\mathbf{u}}} \quad & \Gamma(\bar{\mathbf{y}}, \bar{\mathbf{u}}) \\ \text{s.t.} \quad & \bar{\mathbf{y}} = \mathbf{G}\bar{\mathbf{u}} + \Phi\mathbf{x}[k|k], \\ & \mathbf{A}_{\text{in}}\bar{\mathbf{u}} \leq \mathbf{b}_{\text{in}} \end{aligned} \quad (3.10)$$

desta forma, além de considerar o desempenho obtido pela minimização da função custo  $\Gamma$ , a ação de controle também está sujeita as restrições levadas em conta no projeto do controlador preditivo.

O problemas de rastreo também podem ser realizados utilizando controladores preditivos. Em sistemas que já possuem tal característica intrínseca à sua natureza, basta obter o modelo de predição de acordo com a Equação (3.3). Já nos casos em que a planta não a possua, é necessário reescrever o espaço de estados para incorporá-la. Na literatura são encontradas algumas possibilidades: Wang (2009) insere este comportamento ao reescrever o sistema em termos de  $\Delta\mathbf{x}$  e  $\Delta\mathbf{u}$ , já para Rossiter (2017), insere a ação integral ao descrever o espaço de estados apenas em termos de  $\Delta\mathbf{u}$ , como:

$$\begin{bmatrix} \mathbf{x}[k+1] \\ \mathbf{u}[k] \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0}_{p \times n} & \mathbf{I}_p \end{bmatrix}}_{\hat{\mathbf{A}}} \underbrace{\begin{bmatrix} \mathbf{x}[k] \\ \mathbf{u}[k-1] \end{bmatrix}}_{\hat{\mathbf{x}}} + \underbrace{\begin{bmatrix} \mathbf{B} \\ \mathbf{I}_p \end{bmatrix}}_{\hat{\mathbf{B}}} \Delta\mathbf{u}[k], \quad (3.11a)$$

$$\mathbf{y}[k] = \underbrace{\begin{bmatrix} \mathbf{C} & \mathbf{D} \end{bmatrix}}_{\hat{\mathbf{C}}} \hat{\mathbf{x}}[k] + \mathbf{D}\Delta\mathbf{u}[k] + \mathbf{d}, \quad (3.11b)$$

em que  $\mathbf{d}$  é um distúrbio. Em regime permanente, se  $\mathbf{y} = \mathbf{r}$ ,  $\Delta\mathbf{u} = \mathbf{0}$ . Caso haja variações em  $\mathbf{d}$ ,  $\Delta\mathbf{u}$  tem a liberdade para modificar seu valor, levando o sistema a novos estados  $\hat{\mathbf{x}}$  que são capazes de corrigir o erro.

Desta maneira, para obter o modelo de predição que incorpore a ação integral, basta utilizar as matrizes do sistema  $\hat{\mathbf{A}}$ ,  $\hat{\mathbf{B}}$  e  $\hat{\mathbf{C}}$  e os vetores  $\hat{\mathbf{x}}$  e  $\Delta\bar{\mathbf{u}}$  nas Equações (3.4). Deve-se atentar que  $\bar{\mathbf{u}}$  passa a ser  $\Delta\bar{\mathbf{u}}$ , logo a ação de controle que deverá ser aplicada na planta no instante  $k$  é:

$$\mathbf{u}[k|k] = \mathbf{u}[k-1|k] + \Delta\mathbf{u}[k|k]. \quad (3.12)$$

### 3.3 MODELAGEM DE SISTEMAS ROBÓTICOS

Os sistemas robóticos podem ser descritos por modelos cinemáticos e dinâmicos. A cinemática de um mecanismo robótico diz ao estudo dos movimentos do corpo, sem se preocupar com as forças e torques que os geram (WALDRON; SCHMIEDELER, 2016). Por sua vez, a dinâmica fornece uma relação entre atuação e forças de contato no mecanismo, com as acelerações e trajetórias dos movimentos resultantes (FEATHERSTONE; ORIN, 2016).

Assim, nesta seção estão descritas as técnicas utilizadas para obter os modelos utilizados no trabalho, sendo dividida da seguinte forma: na Subseção 3.3.1 são exibidas as abordagens para a descrição e modelagem cinemático de robôs; já na Subseção 3.3.2 é tratada a obtenção do modelo dinâmico deste tipo de sistemas.

#### 3.3.1 Cinemática

Tradicionalmente, um robô pode ser esquematicamente representado, do ponto de vista mecânico, como sendo uma cadeia de  $\mathcal{N}_B$  corpos rígidos, denominados *links* ou elos, conectados por  $n$  juntas de revolução ou prismáticas (SICILIANO et al., 2010). O conjunto formado pelas posições e orientações destes corpos no espaço são denominados de postura. Portanto, a cinemática descreve a postura de um corpo e as demais derivadas desta em relação a uma determinada base (WALDRON; SCHMIEDELER, 2016).

A cinemática pode ser classificada como direta e inversa. A cinemática direta permite expressar a posição e orientação de um efetuador como função das posições das juntas que o compõem. Normalmente, esta descrição é realizada utilizando álgebra linear. Por sua vez, a cinemática inversa determina as posições de juntas baseadas na posição final do efetuador (SICILIANO et al., 2010). A posição e orientação do efetuador determinam a postura que esse ocupa no espaço.

No espaço Euclidiano, o número mínimo de variáveis necessárias para determinar a localização de um corpo é seis, sendo três determinam a posição e as demais a orientação. Ademais, a postura de um corpo é sempre expressa em relação a um sistema de coordenadas que pode estar em um segundo corpo, ou até mesmo fixo no espaço (WALDRON; SCHMIEDELER, 2016). Denomina-se como sistema de base inercial fixo, um sistemas de coordenadas que é fixo no espaço, sendo que todos os movimentos realizados pelo robô são em relação a ele.

Desta forma, um ponto  $\mathbf{r}_i$  localizado no corpo  $i$  pode ser descrito no sistema de coorde-

nadas  $j$ :

$$\begin{bmatrix} \mathbf{r}_j \\ 1 \end{bmatrix} = {}^j\mathbf{T}_i \begin{bmatrix} \mathbf{r}_i \\ 1 \end{bmatrix}, \quad (3.13)$$

sendo que o termo  ${}^j\mathbf{T}_i$  é conhecido como Transformação Homogênea (TH), no qual é possível combinar as rotações e translações em uma notação compacta (WALDRON; SCHMIEDELER, 2016). A construção de  ${}^j\mathbf{T}_i$  é dada por:

$${}^j\mathbf{T}_i = \begin{bmatrix} {}^j\mathbf{R}_i & {}^j\mathbf{p}_i \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}, \quad (3.14)$$

em que  ${}^j\mathbf{R}_i$  é a matriz de rotação da base  $i$  em relação à base  $j$ , e  ${}^j\mathbf{p}_i$  é o vetor entre a origem do sistema  $j$  e a origem de  $i$ .

A representação de um ponto, que está alocado no  $n$ -ésimo corpo de um robô, pode ser obtido através de uma cadeia cinemática. Desta forma, a representação matemática do ponto no espaço, em relação a uma base fixa é realizada através de transformadas entre dois *links* subsequentes que compõe todo o mecanismo e, no final, é realizado o produto entre eles, como a seguir:

$${}^0\mathbf{T}_n = {}^0\mathbf{T}_1 {}^1\mathbf{T}_2 \dots {}^{n-2}\mathbf{T}_{n-1} {}^{n-1}\mathbf{T}_n. \quad (3.15)$$

Quando ocorre apenas a translação ou rotação, a transformação homogênea pode ser simplificada e escrita como:

$$Trans(\mathbf{d}) = \begin{bmatrix} \mathbf{I}_3 & \mathbf{d}^T \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}, \quad (3.16)$$

$$Rot(\hat{\cdot}, \theta) = \begin{bmatrix} \mathbf{R}_{\{\cdot\}}(\theta) & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}. \quad (3.17)$$

Na Equação (3.16),  $\mathbf{d}$  é o deslocamento. Já na Equação (3.17),  $\mathbf{R}_{\{\cdot\}}$  representa a rotação de  $\theta$  no eixo  $\hat{\cdot}$ . As matrizes de rotação em  $\hat{x}$ ,  $\hat{y}$  e  $\hat{z}$  foram obtidas de Waldron e Schmiedeler (2016), estão mostradas no Apêndice B.1.

As bases que são utilizadas para definir as transformadas entre dois *links* podem ser determinadas de forma arbitrária, entretanto estas escolhas podem acarretar dificuldades para computar as transformações (SICILIANO et al., 2010). Para contornar este problema, é estabelecido um conjunto de regras, a convenção de Denavit-Hartenberg, que propõem uma metodologia para obter as matrizes de transformação homogêneas entre dois *links* consecutivos



(WALDRON; SCHMIEDELER, 2016).

Utilizando esta metodologia, são necessárias apenas quatro variáveis para determinar a matriz de transformação homogênea do sistema. A convenção adotada neste trabalho foi obtida de Wit, Siciliano e Bastin (2012), sendo conhecida como notação modificada de Denavit-Hartenberg, e pode ser vista no Apêndice B.2.

Para corpos que se movimentam, a cinemática diferencial estabelece o comportamento da velocidade linear  $\boldsymbol{\nu}$  e angular  $\boldsymbol{\omega}$  no espaço, em relação a uma base fixa, dada a uma variação de posição das juntas (WALDRON; SCHMIEDELER, 2016). Existem duas formas para estabelecer esta grandeza. A primeira é através de abordagens geométricas, enquanto a segunda ocorre de forma analítica (ETH, 2017).

Caso a posição e orientação de um corpo é especificada em termos do número mínimo de parâmetros no espaço operacional, é possível obter a matriz Jacobiana diferenciando as equações de cinemática direta (WIT; SICILIANO; BASTIN, 2012), uma vez que o Jacobiano é por definição uma matriz de todas as primeiras derivadas parciais de uma função vetorial, em relação à outro vetor (KIEBOOM, 2019). Logo, para cada configuração do conjunto de juntas  $\mathbf{q}$ , sendo  $\mathbf{q} = [q_1 \ q_2 \ \dots \ q_n]^T$ , as matrizes Jacobianas representam a relação dos deslocamentos das juntas com o deslocamento da posição do robô no espaço (TZAFESTAS, 2013).

Desta forma, utilizando a componente translacional  ${}^j\mathbf{p}_i$  da matriz de transformadas homogêneas da Equação (3.14):

$${}^j\boldsymbol{\nu}_i = \frac{\partial {}^j\mathbf{p}_i}{\partial \mathbf{q}} \dot{\mathbf{q}} = \mathbf{J}_p(\mathbf{q}) \dot{\mathbf{q}}. \quad (3.18)$$

O termo  $\mathbf{J}_p$  obtido na Equação (3.18) é conhecido como *Jacobiano Translacional*.

Para a obtenção do *Jacobiano Rotacional* é necessário escrever uma função  $\boldsymbol{\varphi}$  da orientação do corpo em relação a uma base fixa, e realizar o mesmo procedimento adotado anteriormente:

$${}^j\boldsymbol{\omega}_i = \frac{\partial {}^j\boldsymbol{\varphi}_i}{\partial \mathbf{q}} \dot{\mathbf{q}} = \mathbf{J}_r(\mathbf{q}) \dot{\mathbf{q}}. \quad (3.19)$$

Concatenando as duas expressões, é obtida a matriz do Jacobiano:

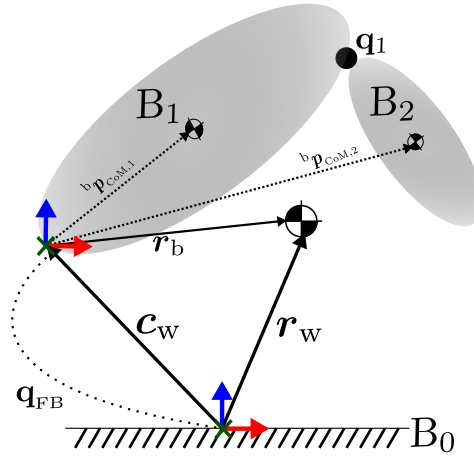
$$\begin{bmatrix} {}^j\boldsymbol{\nu}_i \\ {}^j\boldsymbol{\omega}_i \end{bmatrix} = \begin{bmatrix} \mathbf{J}_p(\mathbf{q}) \\ \mathbf{J}_r(\mathbf{q}) \end{bmatrix} \dot{\mathbf{q}} = \mathbf{J}(\mathbf{q}) \dot{\mathbf{q}}. \quad (3.20)$$

Até aqui, todas as definições realizadas foram feitas para os chamados robôs fixos, entre-

tanto os robôs quadrúpedes fazem parte de outra categoria: robôs móveis, também conhecidos como *robôs de base flutuante*.

As afirmações permanecem válidas, mas algumas novas considerações precisam ser feitas. Robôs com pernas possuem  $\mathcal{N}_B = n + 1$  elos<sup>3</sup>, conectados por um conjunto de  $n$  juntas. Os movimentos do sistema são mesurados em relação a uma base inercial fixa, representada por  $B_0$ . Um dos elos, normalmente o torso, é denotado como a base flutuante e enumerado como  $B_1$ . A posição da base no sistema de coordenadas inercial é denotada por  ${}^w\mathbf{c}$ . Os elos restantes são enumerados de  $B_2$  até  $B_{n+1}$  de tal maneira que o antecessor do corpo  $i$ ,  $\mathcal{P}(i)$ , seja sempre menor que  $i$  (WENSING; PALMER; ORIN, 2015), como exemplificado na Figura 9.

Figura 9 – Numeração dos corpos de um sistema de base flutuante.



Fonte: próprio autor.

Na Figura 9, a ligação virtual,  $q_{FB}$  que une  $B_0$  ao  $B_1$  possui seis graus de liberdade, sendo responsável por conectar a base inercial à flutuante (WENSING; PALMER; ORIN, 2015), possibilitando descrever a posição e orientação do robô no espaço. Estes estados são descritos como:

$$\chi = \begin{bmatrix} {}^w\mathbf{c} & \Theta \end{bmatrix}^T = \begin{bmatrix} {}^wc_x & {}^wc_y & {}^wc_z & \phi & \theta & \psi \end{bmatrix}^T. \quad (3.21)$$

A posição do centro de massa, descrita no sistema de coordenadas global, pode ser obtida utilizando (KWON; OH, 2009):

$${}^w\mathbf{r} = {}^w\mathbf{c} + {}^w\mathbf{R}_b {}^b\mathbf{r}, \quad (3.22)$$

<sup>3</sup>No decorrer do texto os elos serão representados por  $B_n$

sendo que  ${}^b\mathbf{r}$  é a posição do Centro de Massa (CoM)<sup>4</sup> descrita no sistema de coordenadas da base, e é determinado como:

$${}^b\mathbf{r} = \sum_{i=1}^{\mathcal{N}_B} \frac{m_i}{m_T} {}^b\mathbf{p}_{\text{CoM},i}, \quad (3.23)$$

em que  $m_T$  é a massa total do robô,  $m_i$  é a massa do  $i$ -ésimo elo e  ${}^b\mathbf{p}_{\text{CoM},i}$  é a posição do CoM do elo, descrito no sistema de referência da base, respectivamente.

### 3.3.2 Dinâmica

Utilizando as definições estabelecidas na Subseção 3.3.1, é possível saber como um corpo rígido se movimenta pelo espaço.

De maneira semelhante à cinemática, a dinâmica também pode ser dividida entre direta e inversa. A dinâmica direta descreve a aceleração a partir de torques e forças externas aplicadas a um corpo. Por sua vez, a dinâmica inversa descreve o torque ou força a partir da aceleração de todos os corpos da cadeia (FEATHERSTONE; ORIN, 2016). Existem inúmeros modelos matemáticos que descrevem o comportamento dinâmico de robôs (WINKLER, 2018), com diferentes graus de abstração.

Sistemas mecânicos podem ser descritos através da dinâmica de corpos rígidos. Ao fazê-lo, obtém-se uma formulação em espaço de juntas, conhecido como forma canônica (FEATHERSTONE; ORIN, 2016). Para sistemas robóticos com base flutuante a equação canônica é escrita como:

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathcal{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \boldsymbol{\tau}_g(\mathbf{q}) = \mathbf{S}\boldsymbol{\tau} + \mathbf{J}_c^T(\mathbf{q})\mathbf{f}. \quad (3.24)$$

Na Equação (3.24), o termo  $\mathbf{H} \in \mathbb{R}^{6+n \times 6+n}$ , conhecido como matriz de inércia no espaço de juntas, possui uma estrutura simétrica positiva, que descreve a distribuição das massas do robô. A matriz  $\mathcal{C} \in \mathbb{R}^{6+n \times 6+n}$ , é conhecida como matriz de Coriolis. Já o termo  $\boldsymbol{\tau}_g \in \mathbb{R}^{6+n}$ , representa as componentes gravitacionais que influenciam o sistema. Por fim,  $\boldsymbol{\tau} \in \mathbb{R}^n$ , representa os torques que atuam sobre o sistema (FEATHERSTONE; ORIN, 2016).  $\mathbf{S}$  é a matriz de seleção que determina em quais estados os torques das juntas influenciam.  $\mathbf{J}_c$  é o Jacobiano de contato, responsável por mapear a influência das forças externas no  $\mathbf{f}$  na dinâmica do sistema.

Todos os coeficientes da Equação (3.24) são variáveis no tempo, uma vez que dependem

---

<sup>4</sup>do Inglês: *Center of Mass* (CoM)

das configurações e/ou velocidades das juntas, ou seja, o modelo é não linear, o que dificulta o desenvolvimento de aplicações de controle. Normalmente, tal abordagem é usada para simular o sistema robótico. Posto isso, é desejado o uso de modelos simplificados para as tarefas de controle.

Um modelo que é amplamente utilizado, é a aproximação da dinâmica do CoM com a de um pêndulo invertido, e posteriormente realizar a linearização. Com esta aproximação, o modelo é simplificado em um sistema linear, entretanto são necessárias a inclusão de algumas restrições (WINKLER, 2018). Como citado em Orin, Goswami e Lee (2013), é possível derivar outros modelos a partir do primeiro, como, por exemplo, o pêndulo invertido telescópico, pêndulo invertido com impedância variável, de modo a considerar algumas não linearidades.

A abordagem conhecida como *car-table* também é uma simplificação encontrada na literatura. Em tal aproximação, o carro representa a massa do robô, enquanto as pernas da mesa equivalem às do robô. Desta maneira, o movimento é gerado através da força provocada pela movimentação de peso sobre uma mesa (BACHELIER; CHEMORI; KRUT, 2008).

Devido aos seus graus de abstração, os modelos supracitados geralmente são usados para gerar trajetórias, uma vez que não incorporam aspectos das juntas dos robôs (ORIN; GOSWAMI; LEE, 2013). Para incorporar tal aspecto considerações cinemáticas devem ser realizadas.

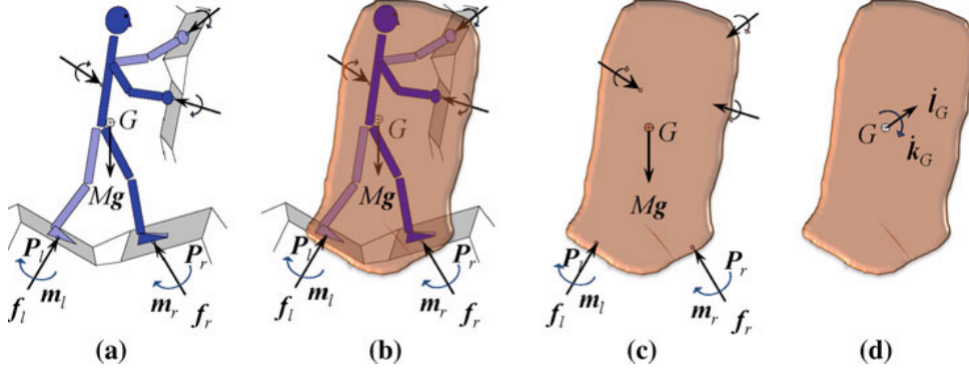
Nos trabalhos de Saranli, Rizzi e Koditschek (2004), Johnson et al. (2012) e Casarez e Fearing (2017), os autores consideraram o momento angular e linear do robô. A partir deste tipo de abordagem, é possível estabelecer o modelo dinâmico do centroide. Sabe-se que as variações destas grandezas físicas são iguais às somas de todas as forças aplicadas no sistema (FEATHERSTONE; ORIN, 2016). Na Figura 10, é vista a abstração que pode ser realizada do ponto de vista físico ao utilizar a dinâmica do centroide. Logo, ao controlar a variação dos momentos, é possível considerar aspectos externos no controle.

Orin e Goswami (2008) mostram que o momento de um robô pode ser obtido através da soma dos momentos individuais de cada elo do sistema, depois da projeção deles no centro de massa do robô. Desta maneira, o momento do sistema robótico pode ser obtido através de uma função das velocidades com a Matriz de Momentos da Centroide (CMM)<sup>5</sup>. Os autores ainda descrevem, os procedimentos para a obtenção da CMM, além de estabelecer a sua relação com a matriz de inércia no espaço de juntas.

---

<sup>5</sup>do Inglês: *Centroidal Momentum Matrix* (CMM)

Figura 10 – Abstração da dinâmica do centroide.



Fonte: disponível em (ORIN; GOSWAMI; LEE, 2013) .

No trabalho de Orin, Goswami e Lee (2013), a abordagem da obtenção da CMM é realizada com um algoritmo recursivo, para o controle de balanço do robô, durante uma caminhada. Li, Yu e Liu (2013) também utilizam a dinâmica do centroide neste mesmo escopo para um robô sujeito à perturbações externas.

Liu et al. (2019) propõem um método que utiliza a dinâmica do centroide, para a tarefa de locomoção de um quadrúpede. A partir de uma sequência de posições dos pés, definidas de acordo com o padrão de movimento, é obtida a referência de momento linear e angular. Um problema de otimização quadrático, que considera as restrições dos atuadores e das forças externas, é utilizado para obter as acelerações das juntas e as forças de reação com o solo, que, por sua vez, são utilizadas em um algoritmo de dinâmica inversa para a computação do toque de referência do *loop* de controle interno.

Outra abordagem que pode ser utilizada para obter o modelo do centroide pode ser observada no trabalho de Villarreal (2020). Quando se trata de robôs quadrúpedes, a dinâmica descrita pela Equação (3.24) pode ser reescrita de modo a deixar explícito a dinâmica do CoM e a dos atuadores:

$$\begin{bmatrix} \mathbf{H}_u & \mathbf{H}_{ua} \\ \mathbf{H}_{au} & \mathbf{H}_a \end{bmatrix} \begin{bmatrix} \ddot{\chi} \\ \ddot{q} \end{bmatrix} + \begin{bmatrix} \mathbf{h}_u \\ \mathbf{h}_a \end{bmatrix} = \begin{bmatrix} \mathbf{J}_{c,u}^T \\ \mathbf{J}_{c,a}^T \end{bmatrix} \mathbf{f}. \quad (3.25)$$

Na Equação (3.25), os termos  $\mathbf{H}_u \in \mathbb{R}^{6 \times 6}$  e  $\mathbf{H}_a \in \mathbb{R}^{n \times n}$  são as matrizes inerciais da parte não atuada e atuada do robô, respectivamente. As matrizes  $\mathbf{H}_{au} \in \mathbb{R}^{6 \times n}$  e  $\mathbf{H}_{ua} \in \mathbb{R}^{n \times 6}$  são os termos correspondentes aos termos cruzados da matriz de inércia.  $\mathbf{h}_u \in \mathbb{R}^6$  e  $\mathbf{h}_a \in \mathbb{R}^n$  incorpora a influência das forças gravitacionais, centrífuga e de Coriolis. Os termos  $\mathbf{J}_{c,u} \in \mathbb{R}^{6n_c \times 6}$  e  $\mathbf{J}_{c,a} \in \mathbb{R}^{6n_c \times n}$  são os Jacobianos de contato.  $\mathbf{f}$  é o vetor de forças externas, expressas na base

inercial de referência, que atuam sobre o robô (VILLARREAL, 2020).

A dinâmica do centroide pode ser obtida através da primeira linha da Equação (3.25):

$$\mathbf{H}_u \ddot{\boldsymbol{\chi}} + \mathbf{H}_{ua} \ddot{\mathbf{q}} + \mathbf{h}_u = \mathbf{J}_{c,u}^T \mathbf{f}, \quad (3.26)$$

de modo que a Equação (3.26) pode ser reescrita separando a dinâmica linear angular:

$$m_T {}^w\ddot{\mathbf{r}} + m_T \mathbf{g} = \sum_{i=1}^{n_c} \mathbf{f}_i, \quad (3.27a)$$

$$\mathcal{I} \dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathcal{I} \boldsymbol{\omega} + \mathbf{H}_{ua} \ddot{\mathbf{q}} = \sum_{i=1}^{n_c} ({}^w\mathbf{r} - \mathbf{p}_{c,i}) \times \mathbf{f}_i. \quad (3.27b)$$

Na Equação (3.27a),  ${}^w\ddot{\mathbf{r}}$  é a aceleração do centro de massa, descrito na base inercial fixa,  $\mathbf{g}$  é o vetor da gravidade,  $n_c$  são os contatos entre o robô e a superfície e  $\mathbf{f}_i$  é o vetor das forças de reação entre o ponto de contato e o robô. Tal equação diz respeito ao movimento linear que o sistema realiza, quando submetido às forças externas. Já na Equação (3.27b),  $\mathcal{I}$  é o tensor de inércia do robô,  $\dot{\boldsymbol{\omega}}$  e  $\boldsymbol{\omega}$  são, respectivamente, as acelerações e velocidades angulares do centro de massa e  $\mathbf{p}_{c,i}$  é a localização do  $i$ -ésimo ponto de contato. Desta forma, é possível descrever o comportamento angular do robô.

A Equação (3.27) é utilizada como base para a obtenção do modelo de predição a ser empregado no RGC, descrita no Capítulo 4.

### 3.4 SÍNTESE DO CAPÍTULO

Ao longo deste Capítulo, conceitos e abordagens foram definidas para encontrar uma possível solução de controle da tarefa de *self-righting* do robô HyQ2Max. Utilizando as definições e abordagens mostradas na Seção 3.2 e 3.1, constatou-se que a utilização do *Reference Governor Control*, implementado via Controle Preditivo Baseado em Modelo, se mostra uma potencial solução deste problema, uma vez que esta arquitetura poderia lidar com restrições do ambiente e do próprio mecanismo, condição que as soluções atuais não tratam como apontado no Capítulo 2. Ao empregar a solução proposta, também é possível garantir certo desempenho, uma vez que é necessário estabelecer uma função custo, de modo que é ponderar questões de rastreamento de estados, ou até mesmo energéticas dos atuadores.

Para implementar o MPC, é necessário obter o modelo dinâmico do sistema. Posto isso, a partir das definições da Seção 3.3, foi possível estabelecer o modelo dinâmico que é

suficientemente representativo, utilizados para a predição. Este modelo é pautado na dinâmica do centroide, uma vez que é possível considerar as forças externas que atua sobre o sistema robótico. As questões que envolvem a implementação podem ser vistas no Capítulo 4.

## 4 DESENVOLVIMENTO

Neste Capítulo, duas abordagens para a realização da tarefa de *self-righting* utilizando a arquitetura de *Reference Governor Control*, implementado a partir de controlador MPC são descritas. Como já discutido no Capítulo 2, será implementado o controlador para realizar a sub tarefa de *stand up* do robô.

Desta forma, as considerações construtivas do HyQ2Max são utilizadas para obtenção do modelo cinemático na Seção 4.1. Já na Seção 4.2, são definidos os modelos de predição e restrição usados no MPC. Por último, na Seção 4.3 é definido a restrição de força de contato.

### 4.1 MODELO CINEMÁTICO

Primeiramente, é necessário realizar algumas definições de nomenclaturas a serem utilizadas no decorrer do trabalho. As quatro patas do robô são nomeadas como  $LF^1$ ,  $RF^2$ ,  $LH^3$ ,  $RH^4$ , sendo que cada uma possui três elos: *hip assembly*; *upper leg*; e *lower leg*.

O *hip assembly* é unido ao torso do robô pela junta  $q_{HAA}^5$ . Já o *link upper leg* é unido ao *hip assembly* pela junta  $q_{HFE}^6$ . Por último, a *lower leg* é unida à *upper leg* através da junta  $q_{KFE}^7$ . A numeração das juntas utilizadas para obter o modelo cinemático é mostrada na Tabela 2:

Tabela 2 – Numeração das juntas de cada pata do robô.

Junta	LF	RF	LH	RH
$q_{HAA}$	$q_1$	$q_4$	$q_7$	$q_{10}$
$q_{HFE}$	$q_2$	$q_5$	$q_8$	$q_{11}$
$q_{KFE}$	$q_3$	$q_6$	$q_9$	$q_{12}$

Fonte: próprio Autor.

Destaca-se que ao longo do trabalho, os eixos dos sistemas de coordenadas são representados da seguinte forma: o eixo  $\hat{x}$  é reapresentado pelo vetor em vermelho; o eixo  $\hat{y}$  é reapresentado pelo vetor em verde; e, o eixo  $\hat{z}$  é reapresentado pelo vetor em azul. Posto isso,

<sup>1</sup>do Inglês: *Left-Front* (LF)

<sup>2</sup>do Inglês: *Right-Front* (RF)

<sup>3</sup>do Inglês: *Left-Hind* (LH)

<sup>4</sup>do Inglês: *Right-Hind* (RH)

<sup>5</sup>do Inglês: *Hip Abduction/Adduction* (HAA)

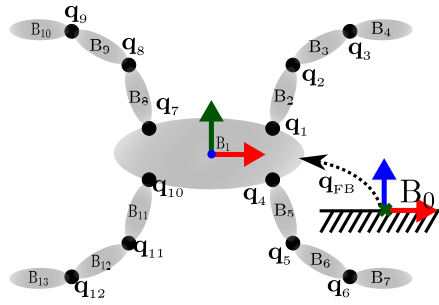
<sup>6</sup>do Inglês: *Hip Flexion/Extension* (HFE)

<sup>7</sup>do Inglês: *Knee Flexion/Extension* (KFE)



segundo a notação descrita na Seção 3.3.1, o diagrama cinemático do HyQ2Max é mostrado na Figura 11.

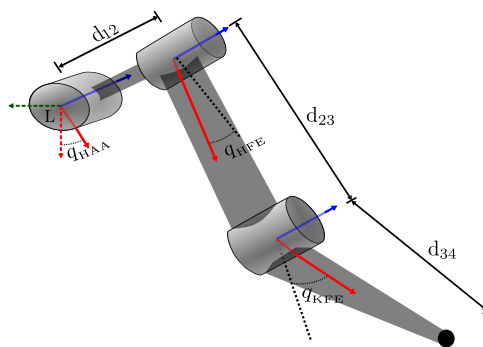
Figura 11 – Numeração dos elos e juntas do HyQ2Max.



Fonte: próprio autor.

Na Figura 12 é mostrado uma simplificação do mecanismo da perna do robô, a qual é utilizada para a obtenção do modelo cinemático da perna. Para a obtenção dos parâmetros de Denavit-Hartenberg, pode-se considerar que cada perna é um manipulador, e portanto utilizar as definições contidas no Apêndice B.2. Ademais, insere-se um sistema de coordenadas auxiliar,  $L$ , que é fixo em relação à base do robô, sendo representado pelos vetores pontilhados na junta HAA, de modo que o eixo  $\hat{z}$  é colinear com o eixo de rotação desta junta. Os parâmetros de Denavit-Hartenberg obtidos são mostrados na Tabela 3.

Figura 12 – Representação das juntas que compõe as patas do robô.



Fonte: próprio autor.

Na Tabela 3, os valores das distancias  $d_{12}$  e  $d_{23}$  são 0,177 m e 0,360 m, respectivamente. Ao longo do trabalho, é estabelecido o sinal negativo para rotações no sentido horário.

Para estabelecer a posição dos pés em relação à base do robô, é necessário obter a

Tabela 3 – Parâmetros de Denavit-Hartenberg das patas do HyQ2Max.

Patras	$a_i$	$\alpha_i$	$d_i$	$\theta_i$
LF & LH	0	0	0	$q_{HAA}$
	0	$\pi/2$	$d_{12}$	$q_{HFE}$
	$d_{23}$	0	0	$q_{KFE}$
RF & RH	0	0	0	$q_{HAA}$
	0	$-\pi/2$	$-d_{12}$	$q_{HFE}$
	$d_{23}$	0	0	$q_{KFE}$

Fonte: próprio Autor.

transformação homogênea entre o joelho e o pé, que é a translação  $d_{34}$  na Figura 12 sobre o eixo  $\hat{x}$  da junta KFE, sendo  $d_{34} = 0,38$  m.

As posições e orientações do sistema de coordenadas L das juntas HAA são constantes em relação à base do robô, logo as transformações homogêneas são definidas através das sequencias de translações e rotações descritas na Tabela 4.

Tabela 4 – Deslocamento das origens das juntas HAA em relação ao centro geométrico do HyQ2Max.

Pata	$d_x$	$d_y$	$\hat{y}$	$\hat{z}$
LF	$d_{FH}/2$	$d_{LR}/2$	$-\pi/2$	$-\pi$
RF	$d_{FH}/2$	$-d_{LR}/2$	$\pi/2$	0
LH	$-d_{FH}/2$	$d_{LR}/2$	$-\pi/2$	$-\pi$
RH	$-d_{FH}/2$	$-d_{LR}/2$	$\pi/2$	0

Fonte: próprio Autor.

As distâncias  $d_{FH}$  e  $d_{LR}$  são de 0,887 m e 0,194 m, respectivamente.

A partir destas informações, foi possível estabelecer todas as THs, sendo então obtido o modelo cinemático do HyQ2Max. A partir da parte linear das transformações homogêneas entre a base e os pés, utilizando a Equação (3.18), foram estabelecidos os Jacobianos de contato  $\mathbf{J}_{c,i}(\mathbf{q})$ .

Ainda foram determinados os Jacobianos dos centro de massa de cada elo. Para obtê-los, foi necessário descrever a posição  $\mathbf{p}_{CoM}$  de cada membro em relação a sua origem. Desta maneira, a transformação homogênea se caracteriza por uma translação entre o eixo de rotação da junta e a posição do CoM.

Considerando as THs das juntas, foi possível compor a cadeia cinemática entre a base do robô e cada um dos doze CoMs dos elos que compõem o robô. Novamente, utilizando a

Equação (3.18), e a parte linear das transformações, foram obtidos os Jacobianos do centro de massa  $\mathbf{J}_{\text{CoM},i}(\mathbf{q})$ . Na Tabela 5, são descritas as posições dos centro de massa de cada um dos elo, além de suas respectivas massas, além de tais informações a cerca do dorso do robô.

Tabela 5 – Posições dos CoMs e massas dos elos.

Membros	$\mathbf{p}_{\text{CoM}}$ [m]	m [kg]
Dorso	$\begin{bmatrix} 6,06 \times 10^{-2} & 5,98 \times 10^{-4} & -7,59 \times 10^{-3} \end{bmatrix}$	39,71
<i>Hip assembly</i>	$\begin{bmatrix} -6,05 \times 10^{-4} & -1,34 \times 10^{-3} & 1,96 \times 10^{-4} \end{bmatrix}$	3,54
<i>Upper leg</i>	$\begin{bmatrix} 1,44 \times 10^{-1} & -2,07 \times 10^{-3} & -5,09 \times 10^{-3} \end{bmatrix}$	4,95
<i>Lower leg</i>	$\begin{bmatrix} 1,09 \times 10^{-1} & 4,21 \times 10^{-2} & 2,19 \times 10^{-4} \end{bmatrix}$	1,71

Fonte: próprio Autor.

Os termos  $\mathbf{J}_{c,i}(\mathbf{q})$  e  $\mathbf{J}_{\text{CoM},i}(\mathbf{q})$  encontrados nesta seção foram utilizados na Seção 4.2 para a obtenção dos modelos de predição utilizados na solução do problema desta dissertação.

## 4.2 MODELOS DE PREDIÇÃO E RESTRIÇÕES

A primeira arquitetura de RGC utilizada, é descrita na Subseção 4.2.1, e adota a metodologia de modelagem proposta por Klaučo, Kalúz e Kvasnica (2017), onde as dinâmicas do controlador interno e da planta são determinadas de modo a obter dois espaços de estados distintos. A partir destes, é obtido a dinâmica de malha fechada do sistema, e, consequentemente o modelo de predição. Ao longo do trabalho, ao ser referir a esta abordagem será utilizado o termo *RGC-PD+P*.

Já na segunda, descrita na Subseção 4.2.2, a dinâmica de malha fechada do sistema foi obtida de forma direta, apenas com manipulações algébricas. No decorrer do texto, para se referir a esta abordagem, será utilizado o termo *RGC-MdlSimp*.

Considera-se na obtenção dos modelos: que todos os pés estão em contato com o solo; a velocidade da base e das juntas podem ser mensuradas; é possível estimar a velocidade do CoM; e que a matrizes do Jacobianos são constantes ao longo do horizonte de predição e controle.

#### 4.2.1 Primeira abordagem: RGC-PD+P

Utilizando a Equação (3.27a), a aceleração do centro de massa do robô, no sistema de coordenadas fixa, é descrita como como:

$${}^w\ddot{\mathbf{r}} = -\mathbf{g} + \frac{1}{m_T} \sum_{i=1}^{nc} \mathbf{f}_i. \quad (4.1)$$

Como estabelecido em Featherstone e Orin (2016), é possível relacionar a força que age sobre um efetuador, no caso os pés, com os torques aplicados nas juntas utilizado o Jacobiano. Portanto, o vetor de forças  $\mathbf{f}_i$  de contato, expresso no sistema de coordenadas fixa, pode ser escrito como:

$$\mathbf{f}_i = -{}^w\mathbf{R}_b \mathbf{J}_{c,i}^{-T} \boldsymbol{\tau}, \quad (4.2)$$

no qual  $\mathbf{J}_{c,i}$  é o Jacobianos entre a base e o  $i$ -ésimo ponto de contato e  $\boldsymbol{\tau}$  são os torques aplicados pelos atuadores. A operação  $(.)^{-T}$  representa a transposta inversa de uma matriz. O sinal negativo na Equação (4.2) se deve a Força de Reação do Solo (GRF)<sup>8</sup>. Ao considerar que os quatro pés estão em contato, logo  $i = \{\text{LF}, \text{RF}, \text{LH}, \text{RH}\}$ , o termo que diz respeito ao somatório da forças da Equação (4.1) pode ser escrito como:

$$\sum_{i=\text{LF}}^{\text{RH}} \mathbf{f}_i = \underbrace{\begin{bmatrix} \mathbf{I}_3 & \mathbf{I}_3 & \mathbf{I}_3 & \mathbf{I}_3 \end{bmatrix}}_{\mathbf{I}_\Sigma} \begin{bmatrix} \mathbf{f}_{\text{LF}} \\ \mathbf{f}_{\text{RF}} \\ \mathbf{f}_{\text{LH}} \\ \mathbf{f}_{\text{RH}} \end{bmatrix}. \quad (4.3)$$

Substituindo cada um dos termos  $\mathbf{f}_i$  da Equação (4.3) pela Equação (4.2) e substituindo na Equação (4.1), a aceleração do CoM é reescrita como:

$${}^w\ddot{\mathbf{r}} = -\mathbf{g} - \frac{\mathbf{I}_\Sigma}{m_T} {}^w\mathbf{J}_c^{-T} \boldsymbol{\tau}, \quad (4.4)$$

em que

$${}^w\mathbf{J}_c = \begin{bmatrix} {}^w\mathbf{R}_b \mathbf{J}_{c,\text{LF}} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & {}^w\mathbf{R}_b \mathbf{J}_{c,\text{RF}} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & {}^w\mathbf{R}_b \mathbf{J}_{c,\text{LH}} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & {}^w\mathbf{R}_b \mathbf{J}_{c,\text{RH}} \end{bmatrix}. \quad (4.5)$$

---

<sup>8</sup>do Inglês: *Ground Reaction Force* (GRF)

Já a velocidade do centro de massa, também no sistema de coordenadas fixa, pode ser obtido através da derivada da Equação (3.22):

$${}^w\dot{\mathbf{r}} = {}^w\dot{\mathbf{c}} + {}^w\dot{\mathbf{R}}_b {}^b\mathbf{r} + {}^w\mathbf{R}_b {}^b\dot{\mathbf{r}}, \quad (4.6)$$

em que o termo  ${}^w\dot{\mathbf{c}}$  é a velocidade da base no sistema de coordenadas fixa, ao passo que  ${}^w\dot{\mathbf{r}}_b$  é a velocidade do centro de massa no sistema de coordenadas da base. Durante a tarefa, o robô realiza apenas movimentos lineares, portanto, é assumido que a orientação do robô se mantém constante ao longo do movimento, logo  ${}^w\dot{\mathbf{R}}_b = \mathbf{0}_{3 \times 3}$ . Devido às restrições impostas pelos contatos,  ${}^w\dot{\mathbf{c}}$  pode ser determinada em termos das velocidades das juntas como:

$$\boldsymbol{\nu}_{nc} = {}^w\dot{\mathbf{c}} + {}^w\mathbf{R}_b \mathbf{J}_{c,nc} \dot{\mathbf{q}}_{nc}, \quad (4.7)$$

sendo  $\boldsymbol{\nu}_{nc}$  a velocidade dos pés. Uma vez que esta grandeza é nula por existir contato, a velocidade da base devido a contribuição do movimento realizado pela pata pode ser descrita como:

$${}^w\dot{\mathbf{c}} = -{}^w\mathbf{R}_b \mathbf{J}_{c,nc} \dot{\mathbf{q}}_{nc}. \quad (4.8)$$

Considerando que cada uma das quatro patas contribui de forma igual no movimento da base, é possível estabelecer uma relação entre as velocidades de todas as juntas e a velocidade da base:

$${}^w\dot{\mathbf{c}} = \frac{{}^w\mathbf{R}_b}{4} \begin{bmatrix} -\mathbf{J}_{c,LF} & -\mathbf{J}_{c,RF} & -\mathbf{J}_{c,LH} & -\mathbf{J}_{c,RH} \end{bmatrix} \dot{\mathbf{q}}. \quad (4.9)$$

Por sua vez, a velocidade do centro de massa  $\dot{\mathbf{r}}_b$  pode ser relacionada aos movimentos das juntas usando:

$${}^w\dot{\mathbf{r}}_b = \sum_{i=1}^n \frac{m_i}{m_T} \mathbf{J}_{CoM,i} \dot{\mathbf{q}}, \quad (4.10)$$

sendo que  $\mathbf{J}_{CoM,i}$  é o Jacobiano entre a base e o centro de massa do  $i$ -ésimo elo. Desta maneira substituindo as Equações (4.9) e (4.10) na Equação (4.6) e reorganizando os termos, é obtido uma relação entre a velocidade do centro de massa e as velocidades das juntas como:

$$\dot{\mathbf{q}} = \mathbf{J}^\dagger {}^w\dot{\mathbf{r}}, \quad (4.11)$$

em que  $(.)^\dagger$  representa a pseudo-inversa da matriz. A matriz  $\mathbf{J}$  é descrita por:

$$\mathbf{J} = \begin{bmatrix} {}^w\mathbf{R}_b \left( \mathbf{J}_{\text{CoM, LF}} - \frac{1}{4} \mathbf{J}_{\text{c, LF}} \right) \\ {}^w\mathbf{R}_b \left( \mathbf{J}_{\text{CoM, RF}} - \frac{1}{4} \mathbf{J}_{\text{c, RF}} \right) \\ {}^w\mathbf{R}_b \left( \mathbf{J}_{\text{CoM, LH}} - \frac{1}{4} \mathbf{J}_{\text{c, LH}} \right) \\ {}^w\mathbf{R}_b \left( \mathbf{J}_{\text{CoM, RH}} - \frac{1}{4} \mathbf{J}_{\text{c, RH}} \right) \end{bmatrix}^T. \quad (4.12)$$

Utilizando as Equações (4.4) e (4.11), é possível obter a dinâmica do sistema escrita em espaço de estados contínuo, como:

$$\begin{bmatrix} {}^w\ddot{\mathbf{r}} \\ \dot{\mathbf{g}} \\ \dot{\mathbf{q}} \end{bmatrix} (t) = \begin{bmatrix} \mathbf{0}_{3 \times 3} & -\mathbf{I}_3 & \mathbf{0}_{3 \times 12} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 12} \\ \mathbf{J}^\dagger & \mathbf{0}_{12 \times 3} & \mathbf{0}_{12 \times 12} \end{bmatrix} \begin{bmatrix} {}^w\dot{\mathbf{r}} \\ \mathbf{g} \\ \mathbf{q} \end{bmatrix} (t) + \begin{bmatrix} -\frac{\mathbf{I}_\Sigma}{m_T} {}^w\mathbf{J}_c^{-T} \\ \mathbf{0}_{3 \times 12} \\ \mathbf{0}_{12 \times 12} \end{bmatrix} \tau(t). \quad (4.13)$$

Considerando as posições das juntas como saída:

$$\mathbf{y}(t) = \begin{bmatrix} \mathbf{0}_{12 \times 3} & \mathbf{0}_{12 \times 3} & \mathbf{I}_{12} \end{bmatrix} \begin{bmatrix} {}^w\dot{\mathbf{r}} \\ \mathbf{g} \\ \mathbf{q} \end{bmatrix} (t). \quad (4.14)$$

Para esta forma de implementação, é necessário determinar a dinâmica dos Controladores PD em espaço de estados, de modo a obter a estrutura mostrada no Apêndice A. Considerando a função de transferência deste tipo de controlador:

$$\frac{u(t)}{e(t)} = \frac{(K_p + N_f K_d)s + K_p N_f}{s + N_f}. \quad (4.15)$$

Observa-se que a Equação (4.15) representa a formulação do controlador mais um filtro passa baixa. A necessidade do filtro se deve ao fato de não ser possível escrever a dinâmica do Controlador PD tradicional em espaço de estados, uma vez que esta é uma função imprópria.

Rescrevendo a Equação (4.15) no formato da Equação (A.1), os termos  $\mathbf{A}_r$ ,  $\mathbf{B}_r$ ,  $\mathbf{C}_r$  e  $\mathbf{D}_r$

de cada um dos Controladores PD são estabelecidos como:

$$\mathbf{A}_r = -N_f; \quad (4.16a)$$

$$\mathbf{B}_r = 1; \quad (4.16b)$$

$$\mathbf{C}_r = -K_d N_f^2; \quad (4.16c)$$

$$\mathbf{D}_r = K_d N_f + K_p. \quad (4.16d)$$

Obtidas as representações em espaços de estados das duas dinâmicas e, depois de discretizá-las, é estabelecido o comportamento em malha fechada do sistema, conforme descrito no Apêndice A. Desta maneira, a partir da Equação (3.1), são obtidos os modelos de predição das saídas da planta  $\bar{\mathbf{q}}$  e das ações de controle  $\bar{\boldsymbol{\tau}}$  como:

$$\bar{\mathbf{q}} = \mathbf{G}_y \Delta \bar{\mathbf{u}} + \Phi_y \hat{\mathbf{x}}[k|k]; \quad (4.17a)$$

$$\bar{\boldsymbol{\tau}} = \mathbf{G}_u \Delta \bar{\mathbf{u}} + \Phi_u \hat{\mathbf{x}}[k|k]. \quad (4.17b)$$

Sob estas condições, é possível reescrever a dinâmica, obtendo um sistema aumentado de modo a incorporar a ação integral. Existem duas possibilidades: rastrear a posição das juntas  $\mathbf{q}$  ou a referência modulada  $\mathbf{w}$ . Independente da escolha, a inclusão é realizada utilizando a abordagem descrita na Seção 3.2, onde os modelos dinâmicos devem ser reescritos em termos de  $\Delta \mathbf{u}$ .

O torque  $\boldsymbol{\tau}$  aplicado ao sistema têm seus valores limitados, portanto, é necessário estabelecer as restrições para estas grandezas. Este tipo de restrição, conhecida como restrição de saída, é estabelecida de forma direta a partir da Equação (4.17b):

$$\boldsymbol{\tau}_{\min} \leq \mathbf{G}_u \Delta \bar{\mathbf{u}} + \Phi_u \hat{\mathbf{x}}[k|k] \leq \boldsymbol{\tau}_{\max}. \quad (4.18)$$

Dito isso, é estabelecido o seguinte problema de otimização:

$$\begin{aligned}
& \min_{\Delta \bar{\mathbf{w}}} \quad \Gamma \\
& \text{s.t.} \quad \bar{\mathbf{q}} = \mathbf{G}_y \Delta \bar{\mathbf{w}} + \Phi_y \hat{\mathbf{x}}[k|k] \\
& \quad \bar{\boldsymbol{\tau}} = \mathbf{G}_u \Delta \bar{\mathbf{w}} + \Phi_u \hat{\mathbf{x}}[k|k] \\
& \quad \begin{bmatrix} \mathbf{G}_u \\ -\mathbf{G}_u \end{bmatrix} \Delta \bar{\mathbf{u}} \leq \begin{bmatrix} \boldsymbol{\tau}_{\max} - \Phi_u \hat{\mathbf{x}}[k|k] \\ \boldsymbol{\tau}_{\min} + \Phi_u \hat{\mathbf{x}}[k|k] \end{bmatrix}
\end{aligned} \tag{4.19}$$

A escolha da função custo  $\Gamma$  é discutida na Seção 5.

#### 4.2.2 Segunda abordagem: RGC-MdlSimp

Para a segunda abordagem, o modelo de impedância das juntas pode ser descrito como:

$$\boldsymbol{\tau}(t) = \mathbf{K}_p(\mathbf{w}(t) - \mathbf{q}(t)) + \mathbf{K}_d(\dot{\mathbf{w}}(t) - \dot{\mathbf{q}}(t)), \tag{4.20}$$

em que  $\mathbf{w}$  e  $\dot{\mathbf{w}}$  são as referências de posição e velocidade das juntas, respectivamente.  $\mathbf{K}_p$  é a matriz que contém os ganhos proporcionais em sua diagonal principal, enquanto  $\mathbf{K}_d$ , os ganhos derivativos. No decorrer das manipulações algébricas, os termos  $(t)$  serão suprimidos.

Substituindo a Equação (4.20) na Equação (4.4) e reorganizando os termos:

$${}^w\ddot{\mathbf{r}} = -\mathbf{g} - \frac{\mathbf{I}_\Sigma \mathbf{J}_c \mathbf{K}_p \mathbf{w}}{m_T} + \frac{\mathbf{I}_\Sigma \mathbf{J}_c \mathbf{K}_p \mathbf{q}}{m_T} - \frac{\mathbf{I}_\Sigma \mathbf{J}_c \mathbf{K}_d \dot{\mathbf{w}}}{m_T} + \frac{\mathbf{I}_\Sigma \mathbf{J}_c \mathbf{K}_d \dot{\mathbf{q}}}{m_T}. \tag{4.21}$$

O termo  $\dot{\mathbf{q}}$  da Equação (4.21) pode ser substituído pela Equação (4.11), uma vez que as considerações usadas para a obter permanecem válidas. Reescrevendo a equação resultante, em espaço de estados, e considerando que  $\dot{\mathbf{w}} = \mathbf{0}$ , uma vez que é desejado um postura estática ao final da tarefa:

$$\begin{bmatrix} {}^w\ddot{\mathbf{r}} \\ \dot{\mathbf{g}} \\ \dot{\mathbf{q}} \end{bmatrix} (t) = \begin{bmatrix} \frac{\mathbf{I}_\Sigma \mathbf{J}_c \mathbf{J}^\dagger \mathbf{K}_d}{m_T} & -\mathbf{I}_3 & \frac{\mathbf{I}_\Sigma \mathbf{J}_c \mathbf{K}_p}{m_T} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{12 \times 3} \\ \mathbf{J}^\dagger & \mathbf{0}_{12 \times 3} & \mathbf{0}_{12 \times 12} \end{bmatrix} \begin{bmatrix} {}^w\dot{\mathbf{r}} \\ \mathbf{g} \\ \mathbf{q} \end{bmatrix} (t) + \begin{bmatrix} -\frac{\mathbf{I}_\Sigma \mathbf{J}_c \mathbf{K}_p}{m_T} \\ \mathbf{0}_{3 \times 12} \\ \mathbf{0}_{12 \times 12} \end{bmatrix} \mathbf{w}(t). \tag{4.22}$$

A equação de saída deste modelo é a mesma da Equação (4.14). Para a implementação do MPC, o espaço de estados deve ser discretizado.

A inclusão da ação integral também é realizada ao reescrever o espaço de estados aumentado mostrado na Seção 3.2. Nesta formulação, também é possível rastrear tanto  $\mathbf{q}$ , quanto



$\mathbf{w}$ .

Usando esta abordagem, a obtenção da restrição do torque não é realizada de forma direta. Para obtê-la, é necessário observar o modelo de impedância da juntas, descrito de forma discreta:

$$\boldsymbol{\tau}[k] = \mathbf{K}_p(\mathbf{w}[k] - \mathbf{q}[k]) - \mathbf{K}_d\dot{\mathbf{q}}[k]. \quad (4.23)$$

Substituindo o termo  $\mathbf{w}[k]$  da Equação (4.23) pela Equação (3.12), e  $\dot{\mathbf{q}}[k]$  pela Equação (4.11):

$$\boldsymbol{\tau}[k] = \mathbf{K}_p\mathbf{w}[k-1] + \mathbf{K}_p\Delta\mathbf{w}[k] - \mathbf{K}_p\mathbf{q}[k] - \mathbf{K}_d\mathbf{J}^\dagger \mathbf{w}\dot{\mathbf{r}}[k]. \quad (4.24)$$

É possível observar que os termos  $\mathbf{w}\dot{\mathbf{r}}[k]$ ,  $\mathbf{q}[k]$  e  $\mathbf{w}[k-1]$  são parte do vetor de estados aumentado  $\hat{\mathbf{x}}[k]$ , portanto é possível reescrever a Equação (4.24) como:

$$\boldsymbol{\tau}[k] = \underbrace{\begin{bmatrix} -\mathbf{K}_d\mathbf{J}^\dagger & \mathbf{0}_{12 \times 1} & -\mathbf{K}_p & \mathbf{K}_p \end{bmatrix}}_{\mathbf{L}} \hat{\mathbf{x}}[k] + \mathbf{K}_p\Delta\mathbf{w}[k]. \quad (4.25)$$

Já no instante  $k+2$ , o torque é definido como:

$$\boldsymbol{\tau}[k+2] = \mathbf{L}\hat{\mathbf{x}}[k+1] + \mathbf{K}_p\Delta\mathbf{w}[k+1]. \quad (4.26)$$

Substituindo  $\hat{\mathbf{x}}[k+1]$  da Equação (4.26) pela Equação (3.11a) e realizando as devidas manipulações, o torque no instante  $k+2$  é reescrito como:

$$\boldsymbol{\tau}[k+2] = \mathbf{L}\hat{\mathbf{A}}\hat{\mathbf{x}}[k] + \begin{bmatrix} \mathbf{L}\hat{\mathbf{B}} & \mathbf{K}_p \end{bmatrix} \begin{bmatrix} \Delta\mathbf{w}[k] \\ \Delta\mathbf{w}[k+1] \end{bmatrix}. \quad (4.27)$$

Extrapolando estes procedimentos para os demais instantes, é observado um padrão, de modo que é viável descrever o torque ao longo dos  $N$  horizontes de predição, como:

$$\bar{\boldsymbol{\tau}} = \underbrace{\begin{bmatrix} \mathbf{K}_p & \mathbf{0}_{p \times p} & \mathbf{0}_{p \times p} & \mathbf{0}_{p \times p} \\ \mathbf{L}\hat{\mathbf{B}} & \mathbf{K}_p & \mathbf{0}_{p \times p} & \mathbf{0}_{p \times p} \\ \vdots & \vdots & \ddots & \mathbf{0}_{p \times p} \\ \mathbf{L}\hat{\mathbf{A}}^{N-1}\hat{\mathbf{B}} & \mathbf{L}\hat{\mathbf{A}}^{N-1}\hat{\mathbf{B}} & \dots & \mathbf{K}_p \end{bmatrix}}_{\mathbf{K}_c} \Delta\bar{\mathbf{w}} + \underbrace{\begin{bmatrix} \mathbf{L} \\ \mathbf{L}\hat{\mathbf{A}} \\ \vdots \\ \mathbf{L}\hat{\mathbf{A}}^{N-1} \end{bmatrix}}_{\mathbf{P}_c} \hat{\mathbf{x}}[k|k]. \quad (4.28)$$

De modo semelhante à primeira abordagem, o torque deve ficar restrito em um *range*.

Posto isso, define-se o seguinte problema de otimização:

$$\begin{aligned}
& \min_{\Delta \bar{\mathbf{w}}} \quad \Gamma \\
& \text{s.t.} \quad \bar{\mathbf{y}} = \mathbf{G} \Delta \bar{\mathbf{w}} + \Phi \hat{\mathbf{x}}[k|k] \\
& \quad \begin{bmatrix} \mathbf{K}_c \\ -\mathbf{K}_c \end{bmatrix} \Delta \bar{\mathbf{w}} \leq \begin{bmatrix} \tau_{\max} - \mathbf{P}_c \hat{\mathbf{x}}[k|k] \\ -\tau_{\min} + \mathbf{P}_c \hat{\mathbf{x}}[k|k] \end{bmatrix}.
\end{aligned} \tag{4.29}$$

### 4.3 RESTRIÇÃO DA FORÇA DE CONTATO

Dependendo da velocidade e aceleração, as forças resultantes podem levar o sistema à instabilidade. Desta forma, é necessário a inclusão da restrição de fricção, que garante que os pés do robô permaneçam em contato com o solo, de modo que o sistema não perca o equilíbrio (DINI; MAJD, 2019).

Considerando  $f_{t,i}$  e  $f_{n,i}$  a magnitude de força tangencial e normal, respectivamente, descritas no sistema de coordenadas fixo, aplicada ao  $i$ -ésimo ponto de contato entre o robô e o solo, e  $\mu$  o coeficiente de atrito estático. A condição  $f_{t,i} \leq \mu f_{n,i}$  deve ser respeitada para que não ocorram deslizamentos dos pés (KHORRAM; MOOSAVIAN, 2017). Sem nenhum prejuízo,  $f_{t,i}$  pode ser decomposta nas componentes  $x$  e  $y$ , de modo que a restrição pode ser representada geometricamente como um cone de fricção (LI; YU; LIU, 2013):

$$\sqrt{f_{x,i}^2 + f_{y,i}^2} \leq \mu f_{n,i}, \tag{4.30}$$

sendo que  $f_{n,i}$  é paralelo ao eixo  $\hat{z}$ , do sistema de coordenadas inercialmente fixo, e pode ser representada com  $f_{z,i}$ .

Para evitar a separação entre os pés e o solo, a força normal deve ser  $f_z \geq 0$  (KHORRAM; MOOSAVIAN, 2017). Apesar desta condição já ser suficiente, Villarreal (2020), utiliza a seguinte condição:

$$f_{z,\min} \leq f_z \leq f_{z,\max}, \tag{4.31}$$

com  $f_{z,\min} > 0$ , desta forma, é garantido que sempre há uma força de reação entre o pé e o chão, ou seja, o pé está em contato com o solo.

Como a Equação (4.30) é não linear, o cone de fricção pode ser aproximado por uma pirâmide. Considerando a condição da Equação (4.31), para que o pé não deslize, deve-se

garantir a seguinte relação entre as forças:

$$\underbrace{\begin{bmatrix} 1 & 0 & -\mu \\ -1 & 0 & -\mu \\ 0 & 1 & -\mu \\ 0 & -1 & -\mu \\ 0 & 0 & 1 \\ 0 & 0 & -1 \end{bmatrix}}_{\mathbf{k}_f} \underbrace{\begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix}}_{\mathbf{f}_i} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -f_{z,\max} \\ f_{z,\min} \end{bmatrix}}_{\mathbf{f}_c} \leq \mathbf{0}_{6 \times 1}. \quad (4.32)$$

Desta forma, considerando a condição que os quatro pés do robô estão em contato com o chão, a Equação (4.32) é reescrita como:

$$\begin{bmatrix} \mathbf{k}_f & \mathbf{0}_{6 \times 3} & \mathbf{0}_{6 \times 3} & \mathbf{0}_{6 \times 3} \\ \mathbf{0}_{6 \times 3} & \mathbf{k}_f & \mathbf{0}_{6 \times 3} & \mathbf{0}_{6 \times 3} \\ \mathbf{0}_{6 \times 3} & \mathbf{0}_{6 \times 3} & \mathbf{k}_f & \mathbf{0}_{6 \times 3} \\ \mathbf{0}_{6 \times 3} & \mathbf{0}_{6 \times 3} & \mathbf{0}_{6 \times 3} & \mathbf{k}_f \end{bmatrix} \begin{bmatrix} \mathbf{f}_{LF} \\ \mathbf{f}_{RF} \\ \mathbf{f}_{LH} \\ \mathbf{f}_{RH} \end{bmatrix} + \begin{bmatrix} \mathbf{f}_c \\ \mathbf{f}_c \\ \mathbf{f}_c \\ \mathbf{f}_c \end{bmatrix} \leq \mathbf{0}_{24 \times 1}. \quad (4.33)$$

Uma vez que força e torque podem ser relacionados pela Equação (4.2), a restrição das forças de contato podem ser consideradas no problema de otimização. Para isso, basta utilizar os modelos de predição do torque estabelecidos nas Equações (4.17b) ou (4.28).

#### 4.4 SÍNTESE DO CAPÍTULO

A partir das considerações realizadas neste capítulo, foi possível estabelecer os modelos de predição usados no MPC para a implementação do *Reference Governor Control*. Também foram definidas as restrições de torques das juntas, bem como as restrições de forças de contatos.

Desta forma, as duas soluções propostas foram implementadas e simuladas de modo a verificar sua efetividade na realização do *stand up* do Hyq2Max, sendo que os resultados obtidos estão mostrados em detalhes no Capítulo 5.

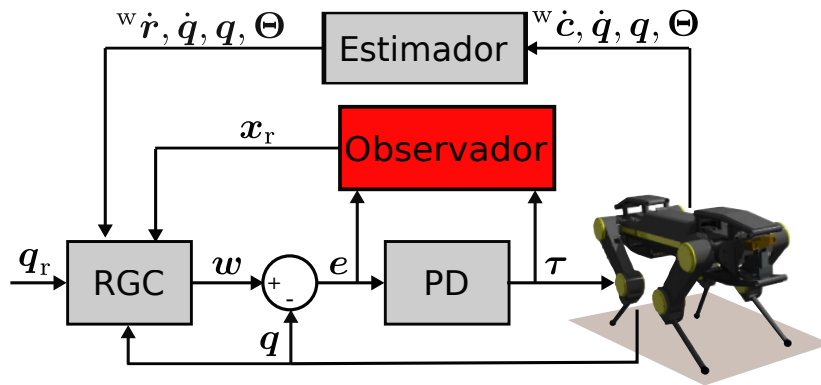
## 5 RESULTADOS

A partir dos modelos do Capítulo 4, as soluções propostas usando a arquitetura de *Reference Governor Control*, implementada via MPC, para a realização do *stand up* foram simuladas e testadas de modo que os resultados estão descritos ao longo desse capítulo.

As abordagens propostas foram implementadas no Matlab e simuladas usando o ambiente de simulação *Simscape Multibody*. O modelo do HyQ2Max foi obtido a partir dos arquivos *Unified Robot Description Format* (URDF) disponibilizados em IIT-DLSLab (2015).

No diagrama de blocos da Figura 13 está ilustrado o arranjo utilizado para as simulações:

Figura 13 – Diagrama de blocos da arquitetura *Reference Governor Control* proposta para a tarefa de *Self-Righting*



Fonte: próprio autor.

O diagrama de blocos é válido para as duas abordagens propostas, entretanto, para a implementação usando a abordagem RGC-PD+P, é necessário a inclusão de um observador de estados, em vermelho na Figura 13, para estimar os valores dos estados internos  $\mathbf{x}_r$  do controlador PD, uma vez que compõem o vetor de estados aumentado  $\tilde{\mathbf{x}}$  da dinâmica de malha fechada do sistema, como mostrado na Equação (3.1a). Neste bloco, os estados  $\mathbf{x}_r$  são estimados utilizando um Filtro de Kalman, a partir do modelo de espaço de estados descrito na Equação (4.17b), dos valores da ação de controle,  $\tau$ , e do erro da posição das juntas,  $e$ .

Os estados  ${}^w\dot{\mathbf{r}}$ , referentes às velocidades do CoM, são estimados a partir da Equação (4.6), uma vez que não é possível mensurar esta grandeza de forma direta. Como os modelos dependem dos Jacobianos, a cada amostragem essas matrizes são atualizadas e consideradas constantes ao longo do horizonte de predição e controle.

Em todos os ensaios de simulação, o robô parte da mesma posição inicial em uma postura

$\mathbf{q}_0$  com atuação apenas dos Controladores PD de juntas. No instante de simulação de 1 s, o *Reference Governor Control* passa a modular  $\mathbf{q}_r$ , obtendo o sinal  $\mathbf{w}$ , que são as referências para os Controladores PD.

Para as soluções dos problemas de otimização, foi utilizado o *solver Quadprog*, disponibilizado no Matlab. Para observar questões acerca do tempo de computação, cada simulação foi repetida dez vezes. Informações sobre o hardware e versões dos softwares utilizados, encontram-se no Apêndice C.1.

### 5.1 ABORDAGEM RGC-PD+P

Nessa primeira abordagem, para o problema de otimização da Equação (4.19), foram considerados três casos com funções de custo distintas. Já em relação às restrições, foi avaliado apenas o caso que considera os limites de torque nas juntas. A frequência do MPC foi de 100 Hz.

Em relação aos torques aplicados na juntas, é inserido uma restrição operacional, garantindo que estas operem em uma região segura. Considerando a situação em que o robô está estático e as quatro patas estão em contato com o solo, idealmente, a força exercida por cada uma é equivalente a um quarto da força peso. Assim, como a massa total do robô é de 80,51 kg, e considerando a gravidade como  $9,81 \text{ m/s}^2$ , cada membro deve proporcionar uma força de aproximadamente 197,45N. Em uma análise simplificada, o toque  $\tau$  aplicado a uma junta pode ser escrito como:

$$\tau = d f \cos \theta, \quad (5.1)$$

sendo  $f$  o módulo da força exercida em um corpo,  $d$  a distância entre o eixo de rotação e  $f$ , denominado braço de alavanca, e  $\theta$  o ângulo formado entre  $f$  e o vetor normal ao braço de alavanca.

A junta que mais exerce torque no sistema é a do joelho, com base nesta informação foi obtido o torque máximo e mínimo das juntas. Considerando o braço de alavanca entre o pé e o joelho sendo 0,36 m, e que no estado inicial  $\cos \theta \approx 1$ , uma vez que a perna inferior está paralela ao solo, nestas condições, o torque necessário para que robô modifique sua postura é de aproximadamente 72 Nm. Entretanto, pode ocorrer situações em que uma das pernas exercem mais força que outra, assim o valor da restrição foi modificada para possibilitar que as juntas apliquem um toque de até 80 Nm.

Três funções custo distintas foram testadas com o objetivo de observar como elas interferiam no controle da tarefa. Para uma melhor visualização, primeiramente serão definidas as três funções custo utilizadas nos ensaios desta primeira arquitetura. No Apêndice C.2 contém um pseudo-código dos procedimentos realizados para a solução dos problemas de otimização. Os horizontes de predição e controle, assim como as matrizes de ponderação foram definidas de forma empírica. Essas e as demais variáveis empregadas na simulação estão descritas no Apêndice C.3.

No primeiro caso, foi considerado que as posições angulares das juntas são as saídas do sistema, de modo que a função custo atribuída ao problema de otimização da Equação (4.19) foi definida como:

$$\Gamma(\mathbf{q}, \Delta\mathbf{w}) = \|\mathbf{q}_r - \mathbf{q}\|_{\mathbf{Q}_q} + \|\Delta\mathbf{w}\|_{\mathbf{Q}_{\Delta w}}, \quad (5.2)$$

em que é realizado a ponderação do erro entre  $\mathbf{q}_r$  e  $\mathbf{q}$ , ou seja, é realizado o rastreamento da variável  $\mathbf{q}$ , de modo que ao final da tarefa, é desejado que as posições finais das juntas sejam iguais as referências de alto nível. Na função custo da Equação (5.2) também é ponderado os incrementos de  $\Delta\mathbf{w}$ .

Já para o segundo caso, foi inserido a ação integral na referência modulada  $\mathbf{w}$ , de modo que a função custo foi determinada como:

$$\Gamma(\mathbf{w}, \Delta\mathbf{w}) = \|\mathbf{q}_r - \mathbf{w}\|_{\mathbf{Q}_w} + \|\Delta\mathbf{w}\|_{\mathbf{Q}_{\Delta w}}, \quad (5.3)$$

neste caso, é realizada a ponderação entre o erro  $\mathbf{q}_r$  e  $\mathbf{w}$ , de modo que é desejado que ao final da tarefa, que a referência modulada atinja o valor da referência de alto nível. Da mesma forma que a função custo da Equação (5.2), na função custo da Equação (5.3), novamente é ponderado os incrementos de  $\Delta\mathbf{w}$ .

Por fim, a última função custo proposta para esta abordagem, incorpora as ponderações dos erros entre  $\mathbf{q}_r$  e  $\mathbf{q}$ ,  $\mathbf{q}_r$  e  $\mathbf{w}$ , assim, é desejado que tanto os valores de  $\mathbf{q}$  e  $\mathbf{w}$  tendam a referência de alto nível. De forma semelhante as duas funções custo anteriores, também é ponderado os incrementos  $\Delta\mathbf{w}$ :

$$\Gamma(\mathbf{q}, \mathbf{w}, \Delta\mathbf{w}) = \|\mathbf{q}_r - \mathbf{q}\|_{\mathbf{Q}_q} + \|\mathbf{q}_r - \mathbf{w}\|_{\mathbf{Q}_w} + \|\Delta\mathbf{w}\|_{\mathbf{Q}_{\Delta w}}. \quad (5.4)$$

Durante a análise dos resultados, observou-se que o par de pernas dianteiras possuem o mesmo comportamento, sendo que ocorre de forma igual para o par traseiro. Desta forma, para

uma melhor visualização dos resultados obtidos, neste capítulo foram vistos apenas os dados referentes à perna dianteira da esquerda, LF, e da perna traseira da direita, RH. Ademais, os movimentos realizados durante os experimentos podem ser observados em vídeo<sup>1</sup>.

Nas Figuras 14, 16 e 18 estão as trajetórias das juntas e do sinal modulado para o primeiro, segundo e terceiro caso, respectivamente. Já nas Figuras 15, 17 e 19 são vistos os torques nas juntas para os mesmos casos, em que  $\tau$  é o valor obtido por meio da leitura do toque aplicado nas juntas;  $\tau_p$  é o valor de torque obtido através do modelo de predição;  $\tau_{\max}$  e  $\tau_{\min}$  são o máximo e mínimo valor de torque admissível, respectivamente. As regiões sombreadas em cinza, representam o momento em que o *Reference Governor Control* age sobre o sistema.

Para uma melhor análise, a Figura 20 contém um comparativo entre cada uma das trajetórias das juntas obtidas de acordo com a função custo empregada e, de forma semelhante, na Figura 21 contém as referências moduladas  $w$  para as mesmas situações.

Usando a abordagem RGC-PD+P, e considerando a custo da Equação (5.2) que pondera o erro da posição das juntas, o sistema ainda exibe erro em regime permanente. Observa-se que, para tentar corrigir o erro de regime permanente, o RGC modula as referências de tal forma que as referências das juntas HAA são praticamente o módulo dos valores das referências de alto nível. Para obter erro zero em regime permanente, uma possível solução é a inclusão de um compensador de gravidade na estrutura. O compensador de gravidade é um estrutura que fornece diretamente o valor do torque necessário para manter o robô em uma dada postura, sendo que este é obtido através da dinâmica inversa da Equação (3.24) para uma situação em que o sistema está estático.

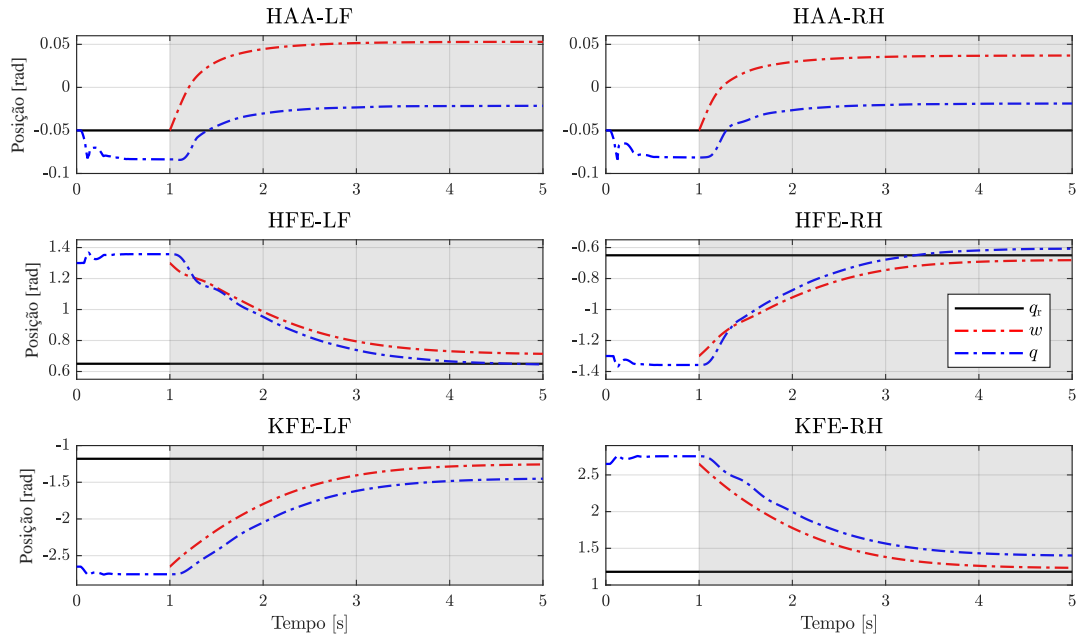
Em relação ao tempo de acomodação, a função custo da Equação (5.2) é a que exibe o maior *settling-time*. Tal desempenho influencia de forma direta nos torques aplicados às juntas, pois devido a menor taxa de variação da posição, mais suave é o comportamento do torque. Para esta condição, constata-se que o modelo de predição do torque previu de forma correta o comportamento dos atuadores.

Para o caso em que a Equação (5.3) é a função custo, sendo os valores  $\Delta w$  são integrados ao longo do tempo, é possível observar que as referências  $w$  tendem às referências  $q_r$ , obtendo erro nulo em regime, como é possível constatar na Figura 16. Identifica-se também, em detrimento do caso que utilizou a Equação (5.2) como função custo, uma melhora considerável na posição final das juntas. Entretanto, para esta situação, é verificado que apesar do modelo

---

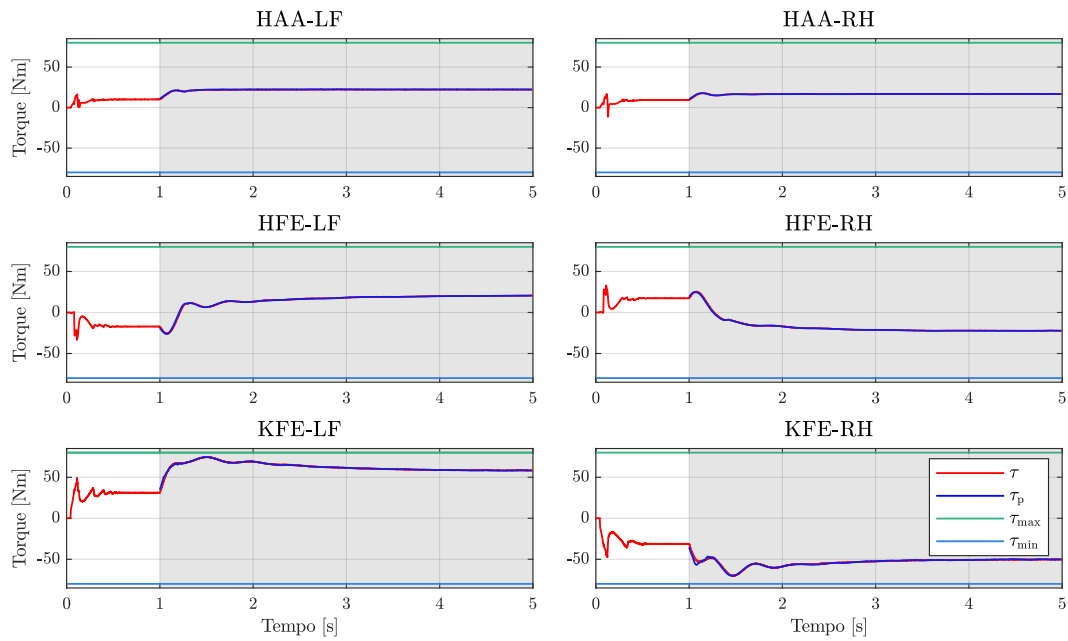
<sup>1</sup>Link do vídeo: <<https://www.youtube.com/watch?v=uDvJ7zRWO7o>>

Figura 14 – RGC-PD+P, função custo da Equação (5.2): trajetórias das juntas  $\mathbf{q}$  e das referências moduladas  $\mathbf{w}$ .



Fonte: próprio autor.

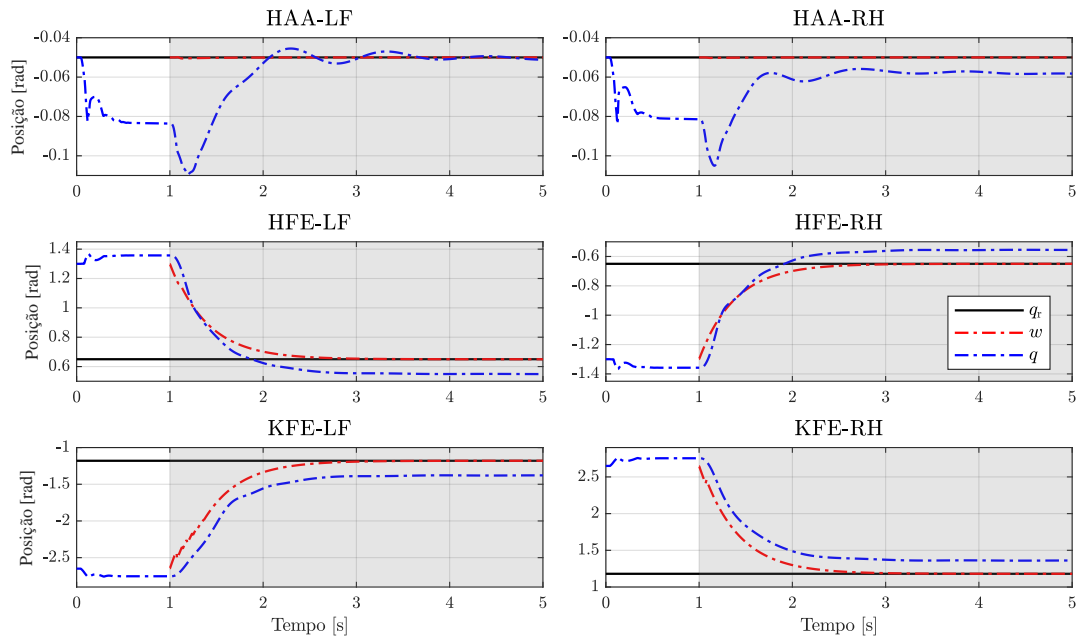
Figura 15 – RGC-PD+P, função custo da Equação (5.2): comportamento do torque.



Fonte: próprio autor.

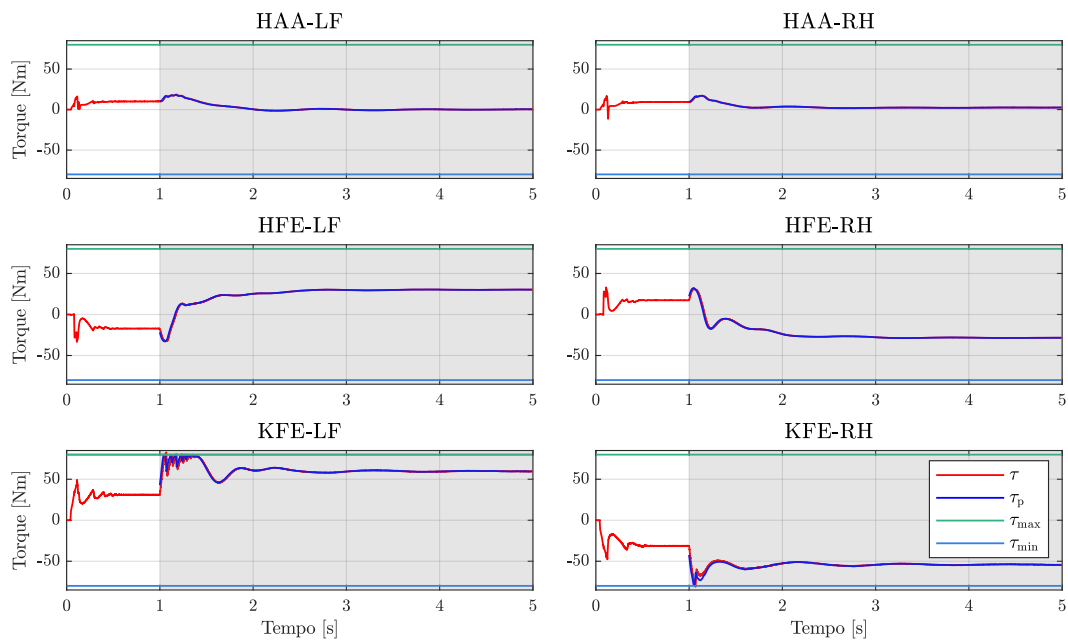


Figura 16 – RGC-PD+P, função custo da Equação (5.3): trajetórias das juntas  $\mathbf{q}$  e das referências moduladas  $\mathbf{w}$ .



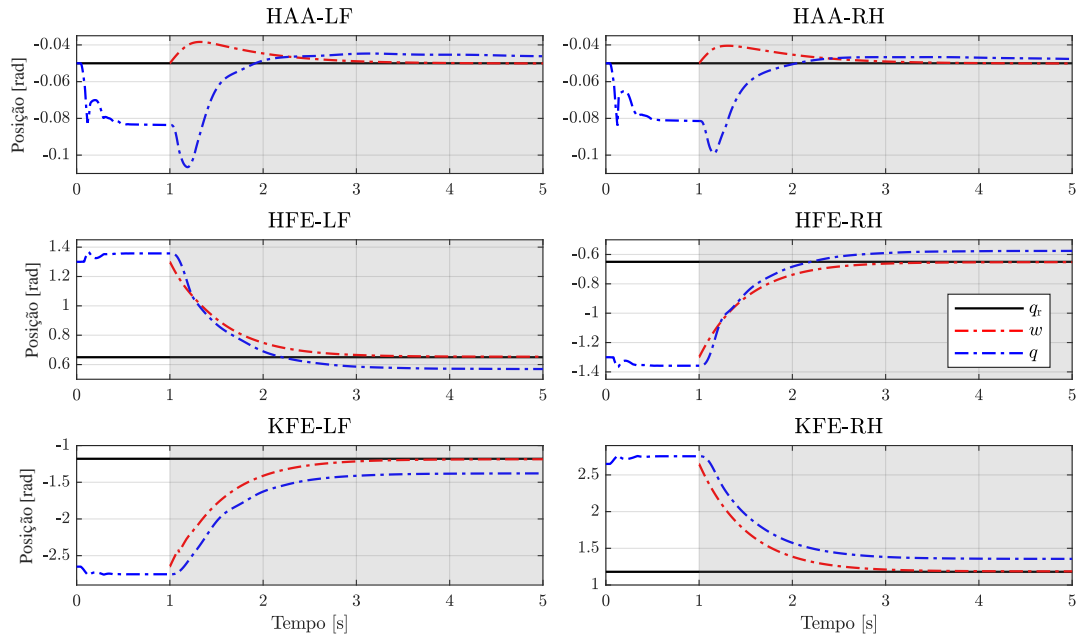
Fonte: próprio autor.

Figura 17 – RGC-PD+P, função custo da Equação (5.3): comportamento do torque.



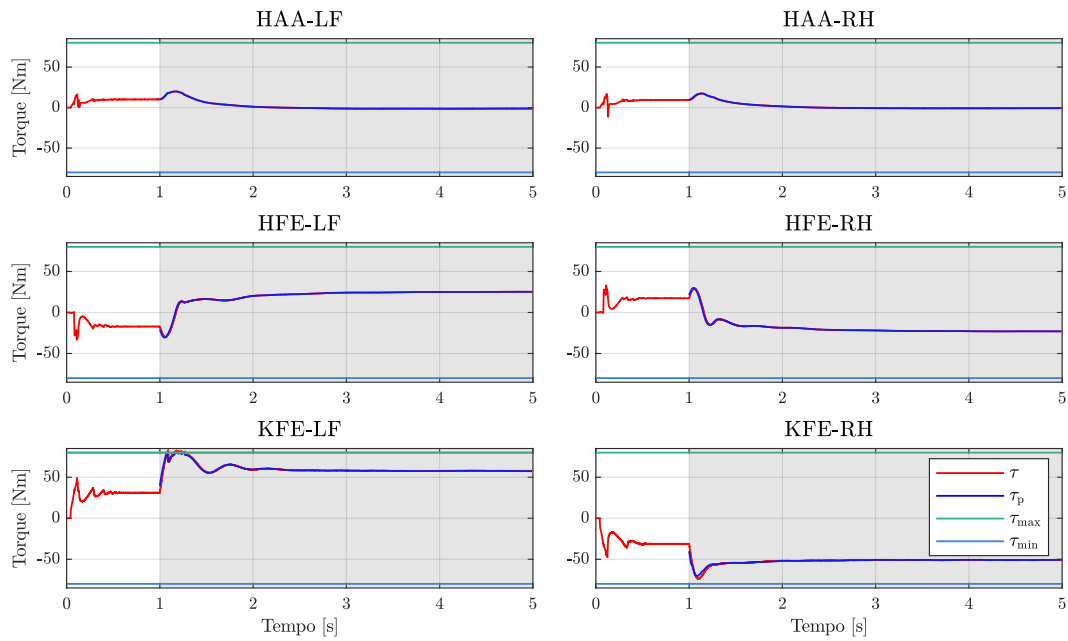
Fonte: próprio autor.

Figura 18 – RGC-PD+P, função custo da Equação (5.4): trajetórias das juntas  $\mathbf{q}$  e das referências moduladas  $\mathbf{w}$ .



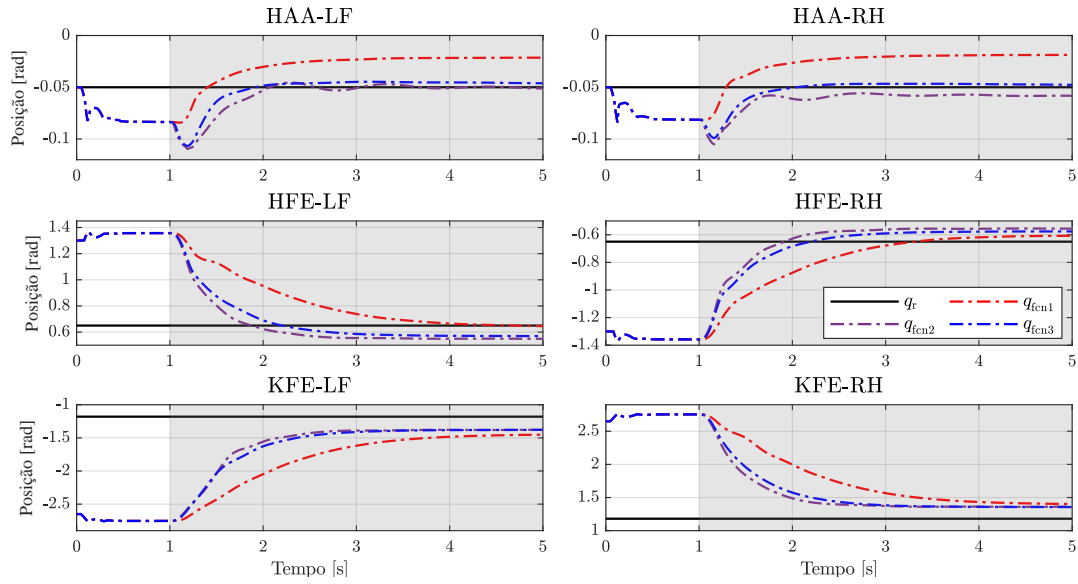
Fonte: próprio autor.

Figura 19 – RGC-PD+P, função custo da Equação (5.4): comportamento do torque.



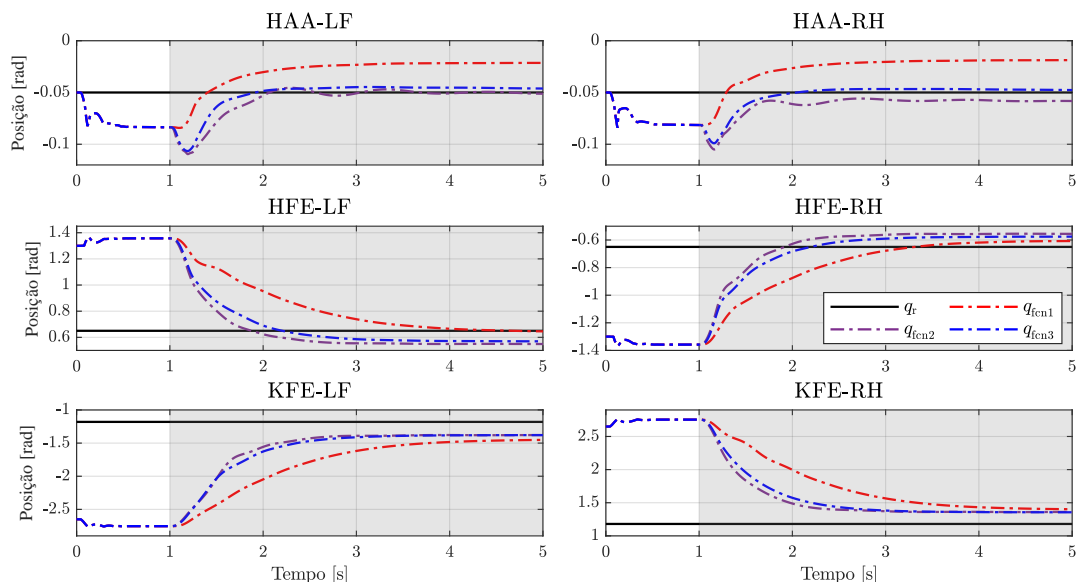
Fonte: próprio autor.

Figura 20 – RGC-PD+P: comparação das trajetórias das juntas  $q$  para as três funções custo.



Fonte: próprio autor.

Figura 21 – RGC-PD+P: comparação das referências moduladas  $w$  para as três funções custo.



Fonte: próprio autor.

de predição saturar em  $\bar{\tau}$ , Figura 17, o torque aplicado no joelho esquerdo, KFE-LF, excede o valor máximo em 2,38 Nm, o que equivale a 2,97 % do valor máximo.

Em relação ao uso da Equação (5.4) como função custo do problema de otimização, observa-se que esta condição possui desempenho semelhante ao uso da Equação da Equação (5.3), no que diz respeito ao rastreo de  $\mathbf{q}$  e  $\mathbf{w}$ . Esta afirmação pode ser comprovada ao comparar os Erros Médios Absolutos (EMA) discriminados na Tabela 6 para estas duas situações. Como é possível verificar na Figura 19, o torque aplicado no joelho dianteiro da esquerda excedeu em aproximadamente 1 Nm, 1,26%, apesar da condição estar sendo respeitada no modelo de predição.

Para as duas funções custo onde o torque excedeu ao valor limite, este ocorrido pode ser justificado devido a diferença do comportamento dinâmico do controlador PD e do modelo usado para a predição do torque, uma vez que é necessário adicionar um filtro passa baixas para a realização do modelo de predição do controlador PD.

Tabela 6 – RGC-PD+P: Erros médios absolutos

	F. custo Equação (5.2)			F. custo Equação (5.3)			F. custo Equação (5.4)		
	EMA <sub>1</sub>	EMA <sub>2</sub>	EMA <sub>3</sub>	EMA <sub>1</sub>	EMA <sub>2</sub>	EMA <sub>3</sub>	EMA <sub>1</sub>	EMA <sub>2</sub>	EMA <sub>3</sub>
$q_{HAA-LF}$	0,0230	0,0924	0,0735	0,0086	0,0000	0,0086	0,0089	0,0031	0,0105
$q_{HFE-LF}$	0,2005	0,2421	0,0508	0,1300	0,0641	0,0939	0,1260	0,0904	0,0793
$q_{KFE-LF}$	0,6703	0,4480	0,2224	0,4099	0,1786	0,2313	0,4442	0,2113	0,2329
$q_{HAA-RF}$	0,0221	0,0917	0,0738	0,0071	0,0000	0,0071	0,0089	0,0030	0,0104
$q_{HFE-RF}$	0,1997	0,2416	0,0513	0,1339	0,0641	0,0938	0,1261	0,0904	0,0788
$q_{KFE-RF}$	0,6681	0,4472	0,2209	0,4225	0,1859	0,2366	0,4451	0,2120	0,2331
$q_{HAA-LH}$	0,0267	0,0790	0,0556	0,0132	0,0000	0,0131	0,0066	0,0027	0,0081
$q_{HFE-LH}$	0,1671	0,1929	0,0594	0,1199	0,0612	0,0834	0,1193	0,0834	0,0719
$q_{KFE-LH}$	0,6229	0,4285	0,1944	0,3413	0,1421	0,1992	0,3986	0,1946	0,2040
$q_{HAA-RH}$	0,0258	0,0782	0,0556	0,0124	0,0000	0,0123	0,0068	0,0026	0,0084
$q_{HFE-RH}$	0,1671	0,1923	0,0597	0,1270	0,0617	0,0862	0,1200	0,0843	0,0727
$q_{KFE-RH}$	0,6207	0,4277	0,1930	0,3527	0,1475	0,2053	0,3990	0,1946	0,2044
EMA <sub>1</sub> = $\frac{\sum_{i=1}^k \mathbf{q}_{r,i} - \mathbf{q}_i}{k}$ ; EMA <sub>2</sub> = $\frac{\sum_{i=1}^k \mathbf{q}_{r,i} - \mathbf{w}_i}{k}$ ; EMA <sub>3</sub> = $\frac{\sum_{i=1}^k \mathbf{w}_i - \mathbf{q}_i}{k}$ .									

Fonte: próprio autor

## 5.2 ABORDAGEM RGC-MDLSIMP

Para esta abordagem, de forma semelhante ao exposto na Seção 5.1, foram observados o desempenho do *Reference Governor Control* para três casos de funções custo distintas. Para o primeiro e segundo caso, considera-se como função custo as Equações (5.3) e (5.4), respecti-

vamente. Já para o terceiro, foi definida uma nova  $\Gamma$ , que além de ponderar o erro entre  $\mathbf{q}_r$  e  $\mathbf{w}$  e os ternos incrementais  $\Delta\mathbf{w}$ , ainda considera a ponderação da diferença entre  $\mathbf{w}$  e  $\mathbf{q}$ , de modo que é escrita da seguinte forma:

$$\Gamma(\mathbf{q}, \mathbf{w}, \Delta\mathbf{w}) = \|\mathbf{q}_r - \mathbf{w}\|_{\mathbf{Q}_w} + \|\mathbf{w} - \mathbf{q}\|_{\mathbf{Q}_e} + \|\Delta\mathbf{w}\|_{\mathbf{Q}_{\Delta w}}. \quad (5.5)$$

O pseudo-código desta abordagem é mostrado no Apêndice C.4. Os valores das matrizes de ponderação utilizados, horizontes de predição e demais informações pertinentes à simulação encontram-se no Apêndice C.5.

Considerando apenas restrições nos torques, as Figuras 22, 24 e 26 contém as trajetórias das juntas  $\mathbf{q}$  e o sinal modulado  $\mathbf{w}$  para cada uma das funções custo. Já os torques podem ser visualizados nas Figuras 23, 25 e 27. A comparação entre as posições das juntas  $\mathbf{q}$ , para cada uma das funções custo, podem ser observadas na Figura 28, ao passo que na Figura 29, a mesma análise é realizada para as referências moduladas  $\mathbf{w}$ .

É possível observar que todas as referências moduladas  $\mathbf{w}$  tendem à referência  $\mathbf{q}_r$ , uma vez que foi adicionada a ação integral para este estado. Já, tratando-se das posições das juntas  $\mathbf{q}$ , constata-se que possuem erro em regime permanente nas três situações. Como já mencionado na Seção 5.1, a inclusão do compensador de gravidade poderia solucionar este problema, entretanto não foi implementado pois não era o foco deste trabalho.

Os torques aplicados às juntas respeitaram os limites estabelecido em todas as situações, constantando que a solução proposta pode proporcionar um funcionamento mais seguro para os componentes mecânicos que compõe o robô.

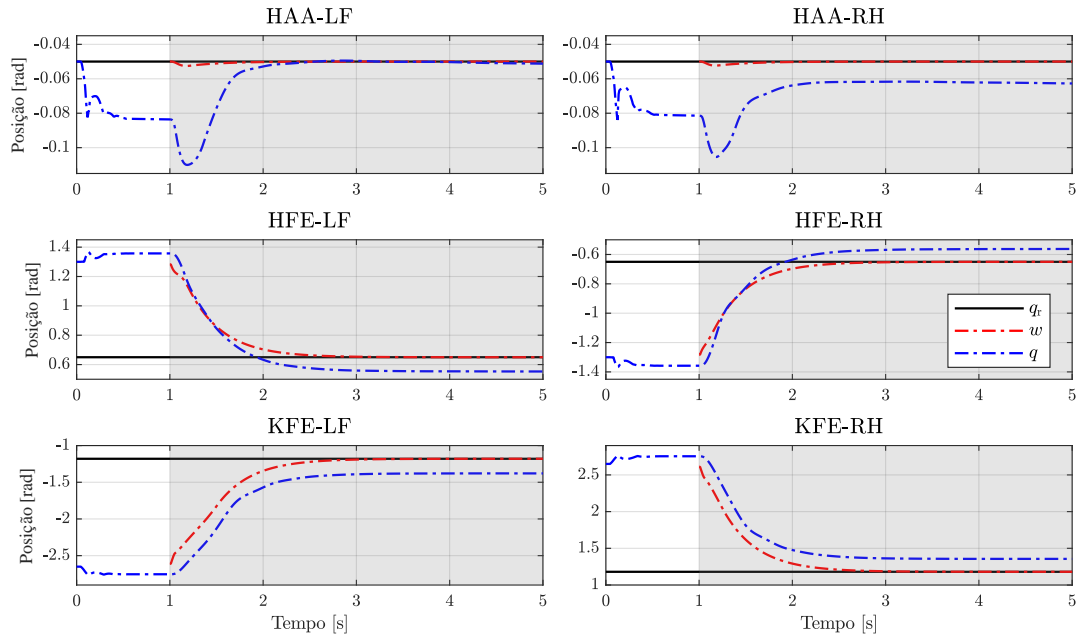
Da mesma maneira que foi realizado para a abordagem RGC-PD+P, os erros médios absolutos para a implementação RGC-MdlSimp, podem ser verificados na Tabela 7.

Constata-se que as três funções custo têm um desempenho próximo, uma vez que, na média, os valores dos erros são bem próximos.

Até aqui, as soluções descritas são funcionais, entretanto, ao observar o comportamento durante a execução das trajetórias obtidas, constatou-se que para todos os casos os pés deslizam sobre a superfície. Tal condição é inaceitável, uma vez que dependendo do movimento, o robô pode sofrer uma nova queda. Desta forma, a inclusão da restrição de forças de contato se faz necessário.

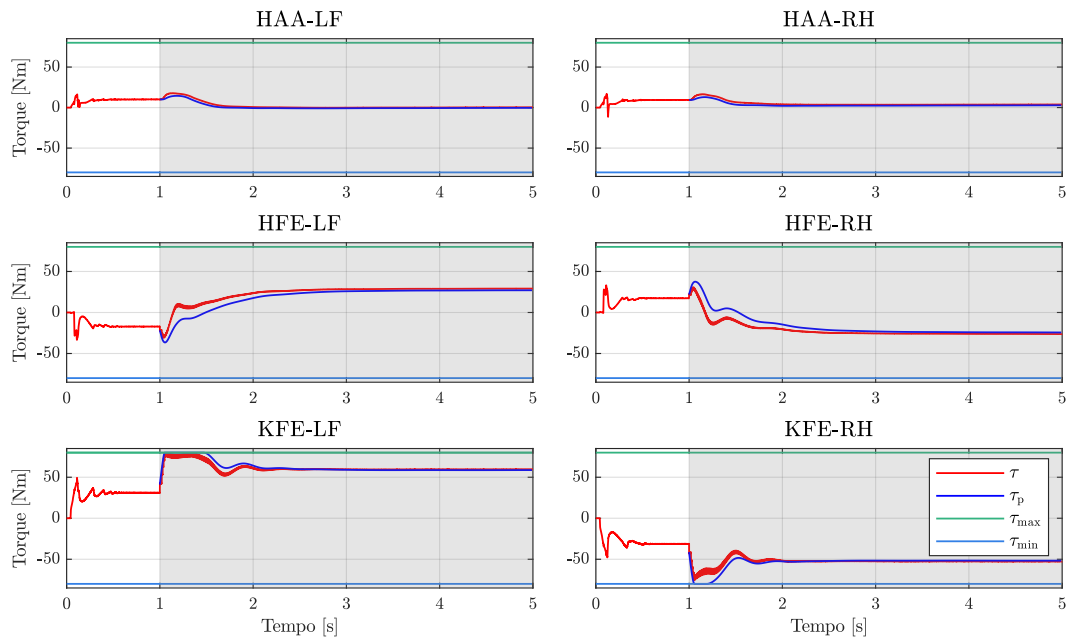
O problema de otimização da Equação (4.29), que utiliza a Equação (5.3) como função custo, foi modificado para que além da restrição de torque, considerasse as restrições das GRFs.

Figura 22 – RGC-MdlSimp, função custo da Equação (5.3): trajetórias das juntas  $\mathbf{q}$  e das referências moduladas  $\mathbf{w}$ .



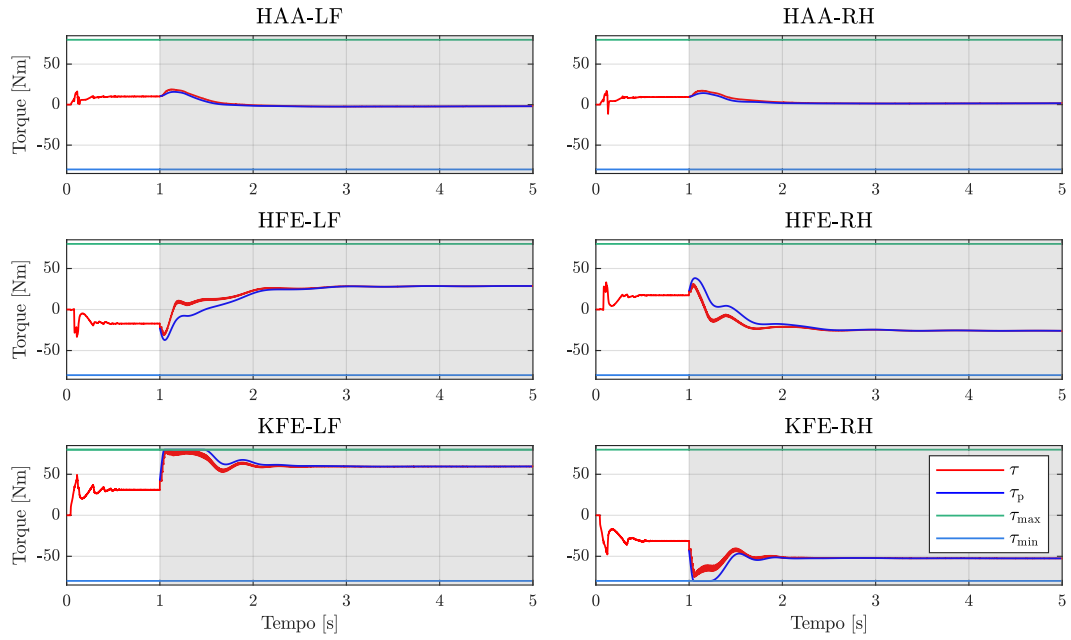
Fonte: próprio autor.

Figura 23 – RGC-MdlSimp, função custo da Equação (5.3): comportamento dos torques.



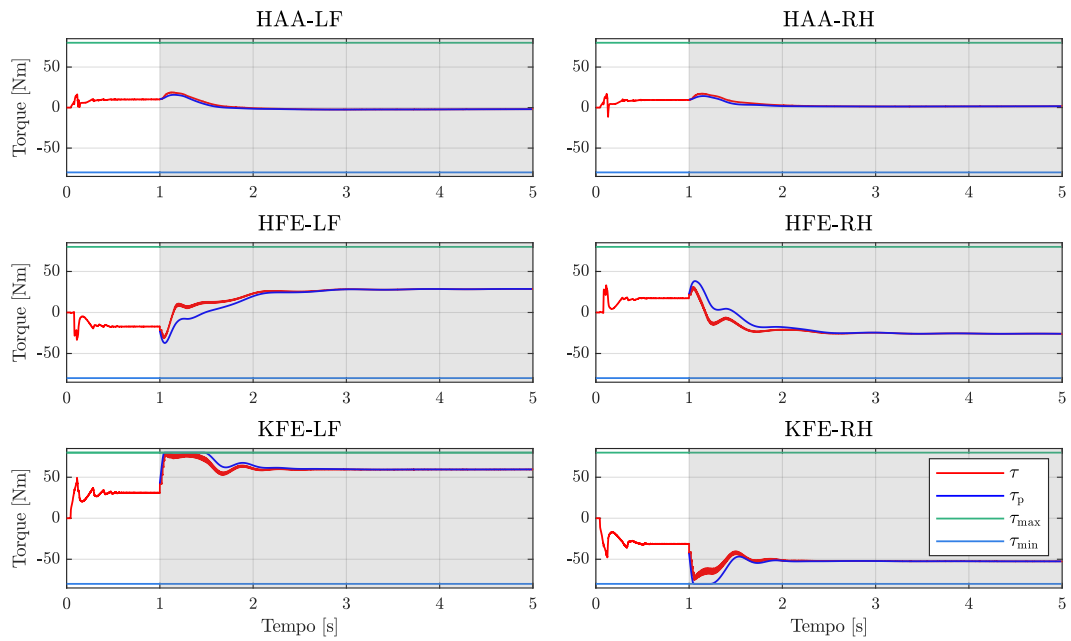
Fonte: próprio autor.

Figura 24 – RGC-MdlSimp, função custo da Equação (5.4): trajetórias das juntas  $\mathbf{q}$  e das referências moduladas  $\mathbf{w}$ .



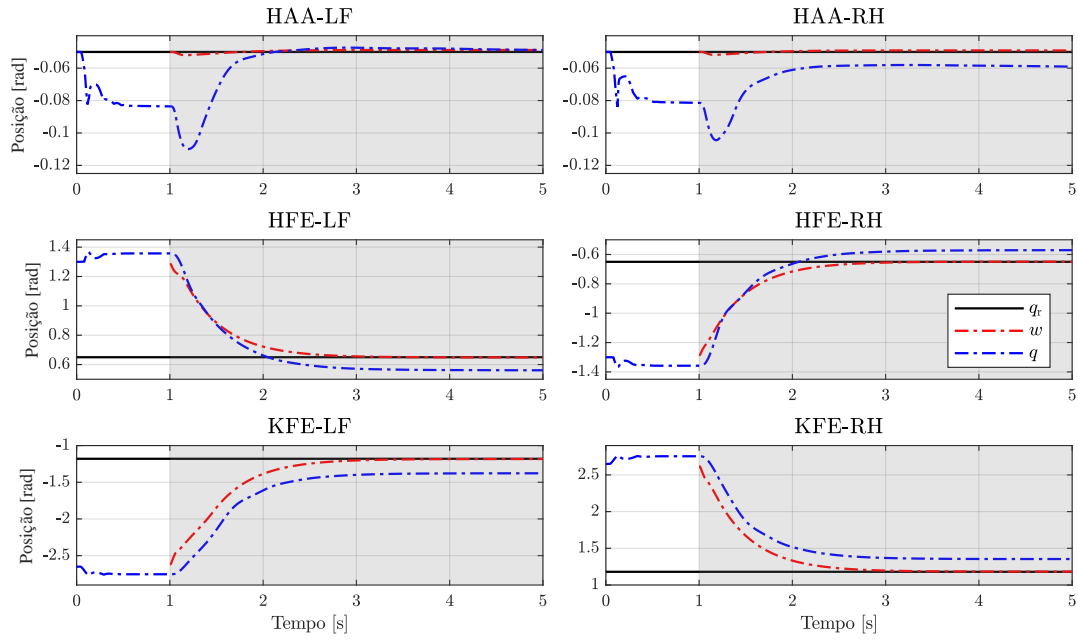
Fonte: próprio autor.

Figura 25 – RGC-MdlSimp, função custo da Equação (5.4): comportamento do torque.



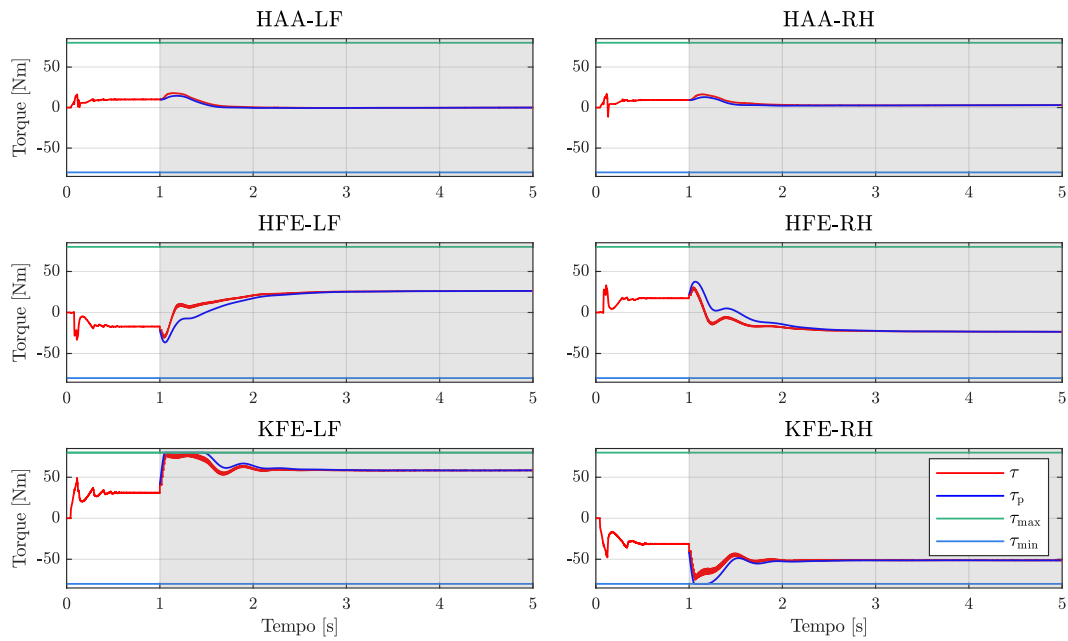
Fonte: próprio autor.

Figura 26 – RGC-MdlSimp, função custo da Equação (5.5): trajetórias das juntas  $\mathbf{q}$  e das referências moduladas  $\mathbf{w}$ .



Fonte: próprio autor.

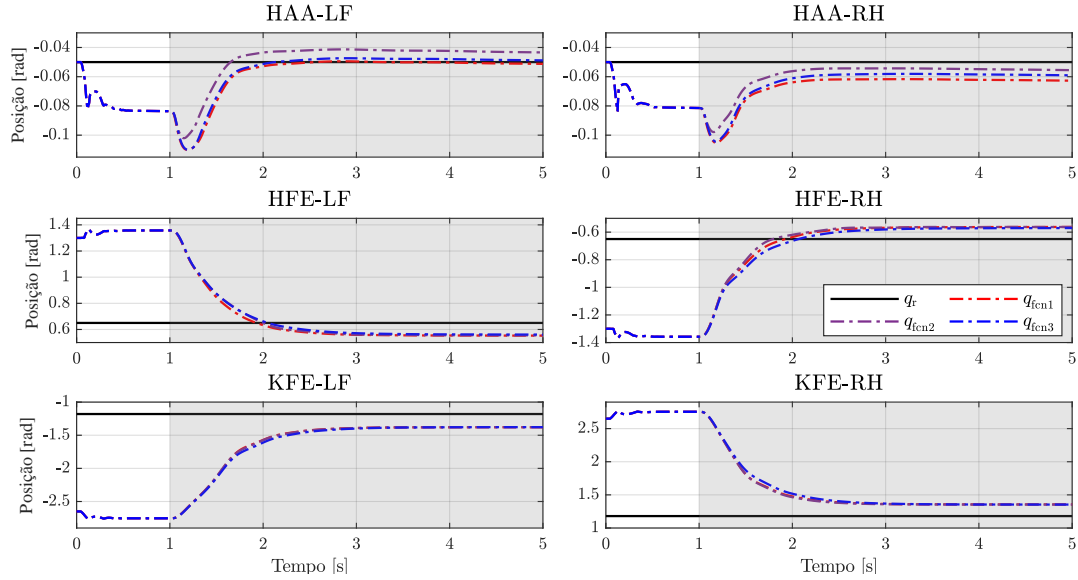
Figura 27 – RGC-MdlSimp, função custo da Equação (5.5): comportamento do torque.



Fonte: próprio autor.

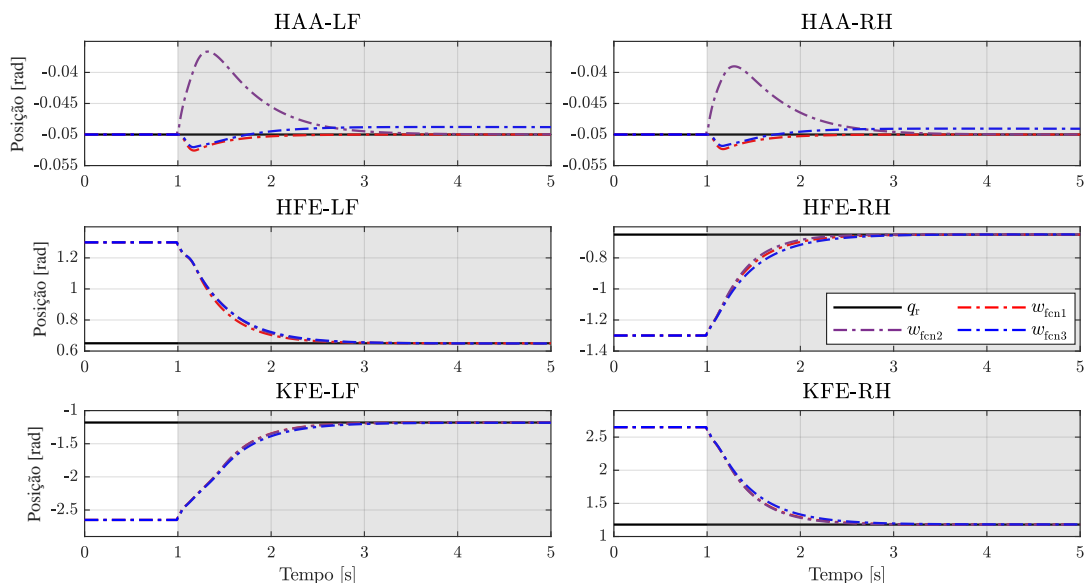


Figura 28 – RGC-MdlSimp: comparação das trajetórias das juntas  $\mathbf{q}$  para as três funções custo.



Fonte: próprio autor.

Figura 29 – RGC-MdlSimp: comparação das referências moduladas  $\mathbf{w}$  para as três funções custo.



Fonte: próprio autor.

Tabela 7 – RGC-MdlSimp: Erros médios absolutos

	F. custo Equação (5.3)			F. custo Equação (5.4)			F. custo Equação (5.5)		
	EMA <sub>1</sub>	EMA <sub>2</sub>	EMA <sub>3</sub>	EMA <sub>1</sub>	EMA <sub>2</sub>	EMA <sub>3</sub>	EMA <sub>1</sub>	EMA <sub>2</sub>	EMA <sub>3</sub>
$q_{\text{HAA-LF}}$	0,0040	0,0001	0,0038	0,0082	0,0021	0,0089	0,0049	0,0008	0,0041
$q_{\text{HFE-LF}}$	0,0930	0,0406	0,0682	0,1003	0,0534	0,0644	0,0909	0,0483	0,0608
$q_{\text{KFE-LF}}$	0,2792	0,1090	0,1702	0,3072	0,1331	0,1742	0,2956	0,1269	0,1687
$q_{\text{HAA-RF}}$	0,0040	0,0001	0,0039	0,0082	0,0019	0,0091	0,0050	0,0008	0,0043
$q_{\text{HFE-RF}}$	0,0928	0,0406	0,0682	0,1002	0,0533	0,0646	0,0908	0,0483	0,0609
$q_{\text{KFE-RF}}$	0,2783	0,1086	0,1696	0,3060	0,1325	0,1736	0,2946	0,1265	0,1682
$q_{\text{HAA-LH}}$	0,0122	0,0001	0,0121	0,0069	0,0018	0,0086	0,0099	0,0006	0,0104
$q_{\text{HFE-LH}}$	0,0850	0,0374	0,0621	0,0897	0,0414	0,0611	0,0833	0,0451	0,0552
$q_{\text{KFE-LH}}$	0,2355	0,0870	0,1485	0,2527	0,1021	0,1510	0,2519	0,1048	0,1471
$q_{\text{HAA-RH}}$	0,0120	0,0001	0,0119	0,0068	0,0016	0,0085	0,0097	0,0006	0,0102
$q_{\text{HFE-RH}}$	0,0848	0,0374	0,0622	0,0896	0,0413	0,0611	0,0831	0,0450	0,0553
$q_{\text{KFE-RH}}$	0,2346	0,0869	0,1477	0,2516	0,1019	0,1502	0,2510	0,1047	0,1463
EMA <sub>1</sub> = $\frac{\sum_{i=1}^k \mathbf{q}_{r,i} - \mathbf{q}_i}{k}$ ; EMA <sub>2</sub> = $\frac{\sum_{i=1}^k \mathbf{q}_{r,i} - \mathbf{w}_i}{k}$ ; EMA <sub>3</sub> = $\frac{\sum_{i=1}^k \mathbf{w}_i - \mathbf{q}_i}{k}$ .									

Fonte: próprio autor

Para a implementação, os valores de torque máximo e mínimo foram aumentados em 25%.

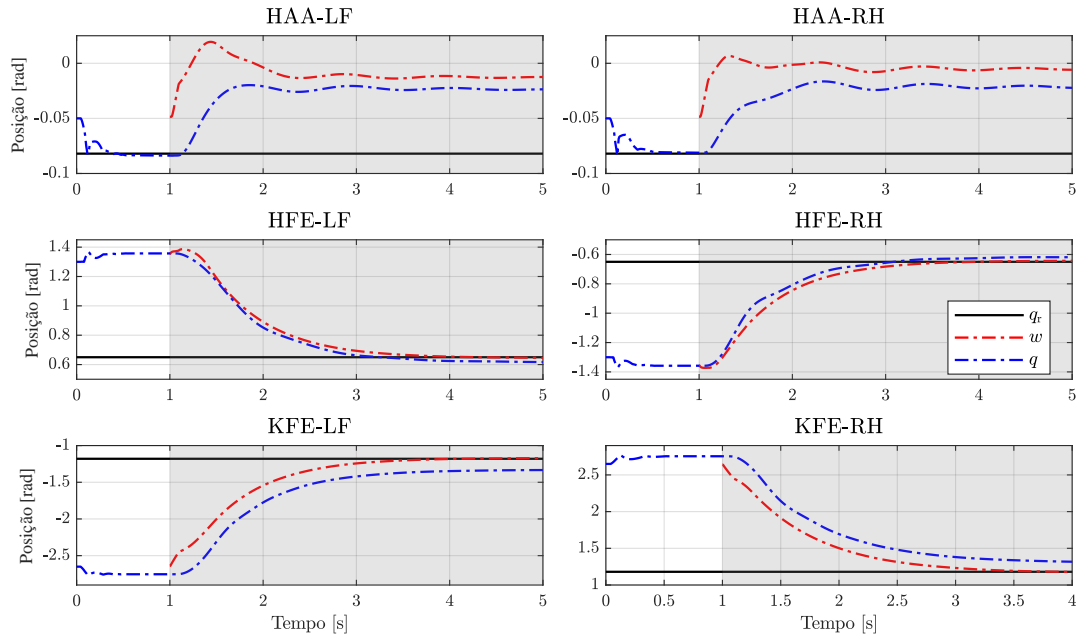
Inicialmente, a frequência de MPC foi mantida em 100 Hz, entretanto, constatou-se que com esta amostragem e considerando as restrições das GRFs, o controlador lavava o robô a situações de instabilidade. Uma possível justificativa para este problema é o fato que é assumido que os Jacobianos são contantes ao longo dos horizontes de predição e controle, desta forma, ao considerar esta aproximação no cálculo das restrições, o modelo perdia a representatividade. Assim, a frequência foi aumentada para 250 Hz, de modo que foi possível realizar o experimento. Os valores das matrizes de ponderação, horizontes de predição e demais constantes utilizadas na simulação estão discriminadas no Apêndice C.6.

Nestas condições, no instante de 1 s, o RGC passa a modificar as referências  $\mathbf{q}_r$  de modo que as trajetórias das juntas  $\mathbf{q}$  e das referências moduladas  $\mathbf{w}$  foram obtidas, e podem ser visualizadas na Figura 30.

Já os comportamentos dos torques podem ser verificados na Figura 31. É possível observar que devido as restrições das GRFs serem mais conservadoras, os torques preditos pelo modelo não chegaram a saturar, como nas situações anteriores.

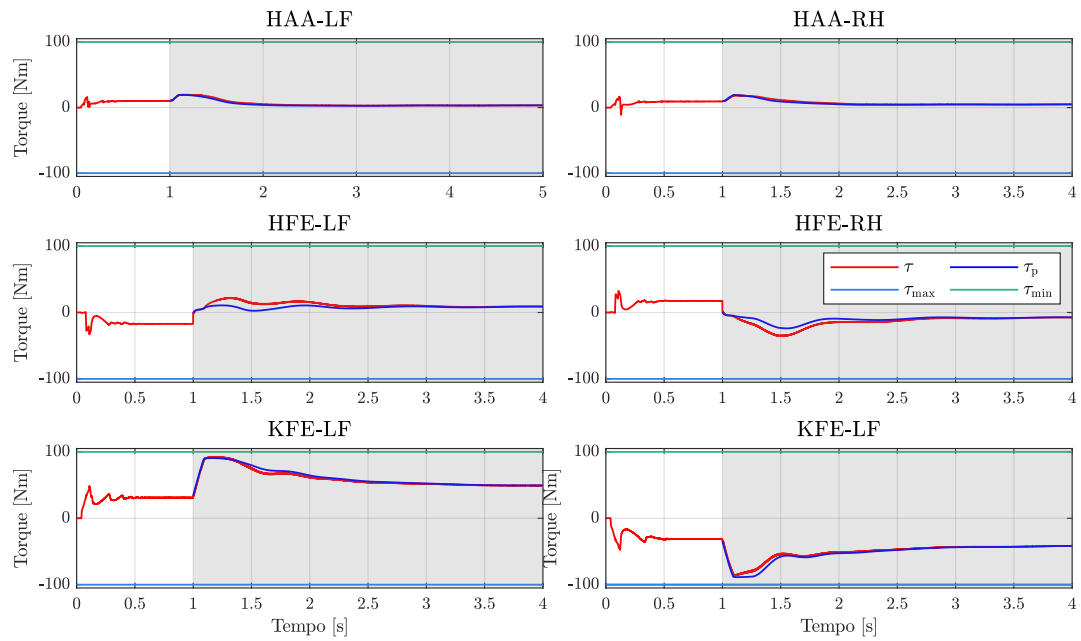
As forças de reação entre os pés e o chão são vistas na Figura 32. Constata-se que os pés respeitaram a restrição imposta pela Equação (4.33), garantido o contato em todo o movimento de *stand up*.

Figura 30 – Trajetórias das juntas  $\mathbf{q}$  e das referências moduladas  $\mathbf{w}$  considerando restrições de torque e força de contato.



Fonte: próprio autor.

Figura 31 – Comportamento dos torques considerando restrições de torque e força.



Fonte: próprio autor.

As trajetórias dos pés obtidos através do problema de otimização que considera restrições de torque e força, e utiliza a Equação (5.3) como função custo, foram comparadas com as mesmas grandezas obtidas através de um problema de otimização usando a mesma função custo e apenas restrições nos torques. Para uma melhor comparação, considerou-se os mesmos valores de ponderação e de limites de torques. A Figura 33 contém as trajetórias obtidas para os quatro pés.

Observa-se na Figura 33 que a posição dos pés muda nos dois problemas de otimização. As distâncias  $d_i$  percorridas pelos pés estão na Tabela 8.

Tabela 8 – Distância percorrida pelos pés.

	$d_{LF}[m]$	$d_{RF}[m]$	$d_{LH}[m]$	$d_{RH}[m]$
s. rest.	0,046	0,046	0,057	0,057
c. rest.	0,014	0,013	0,013	0,013

Fonte: próprio autor

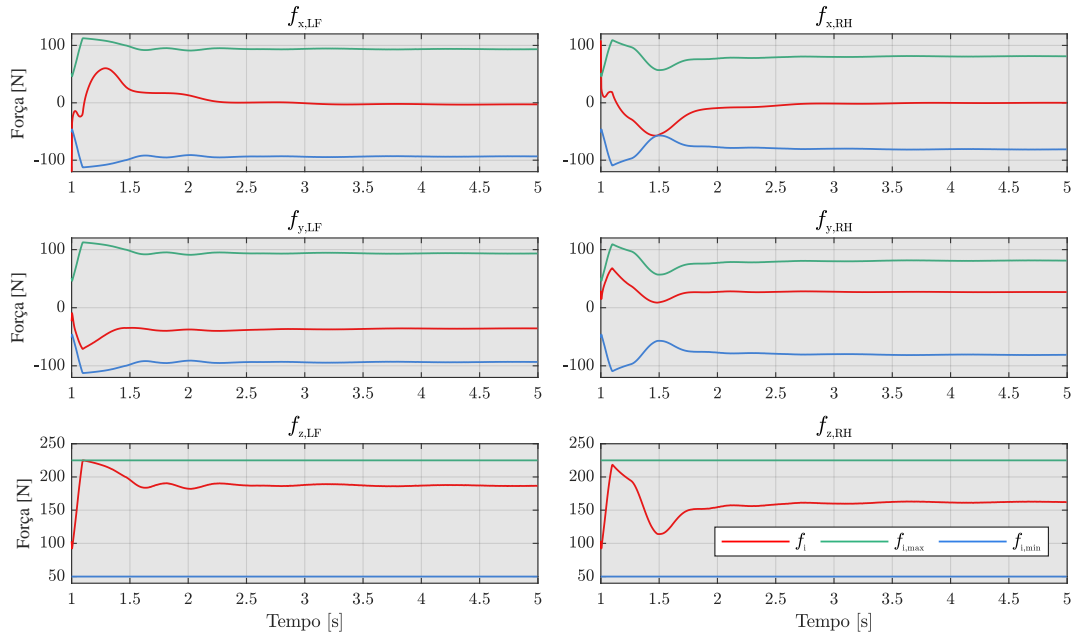
Observa-se que ao não considerar as restrições de forças de contato, os deslocamentos realizados pelos pés são em torno de 3,0 a 4,5 vezes maior, se comparados com a situação que utiliza restrições nas GRFs. Nota-se que mesmo considerando restrições nas GRFs há deslocamentos nos pés, o que é justificado pelo fato da relação entre torques e forças ser descrita por um Jacobiano de contato, que é uma aproximação e, portanto, sujeito a erros.

Uma última consideração foi realizada. Como mencionado na Seção 2, ao empregar o RGC como solução do problema de *self-righting*, é a possível considerar restrições causadas devido ao ambiente.

Para realizar uma demonstração, o novo estado  ${}^w r_z$ , referente à altura do CoM no sistema de coordenadas fixa, foi adicionado ao modelo da Equação (4.22), de modo que é possível inserir restrição da máxima altura que o CoM poderia alcançar.

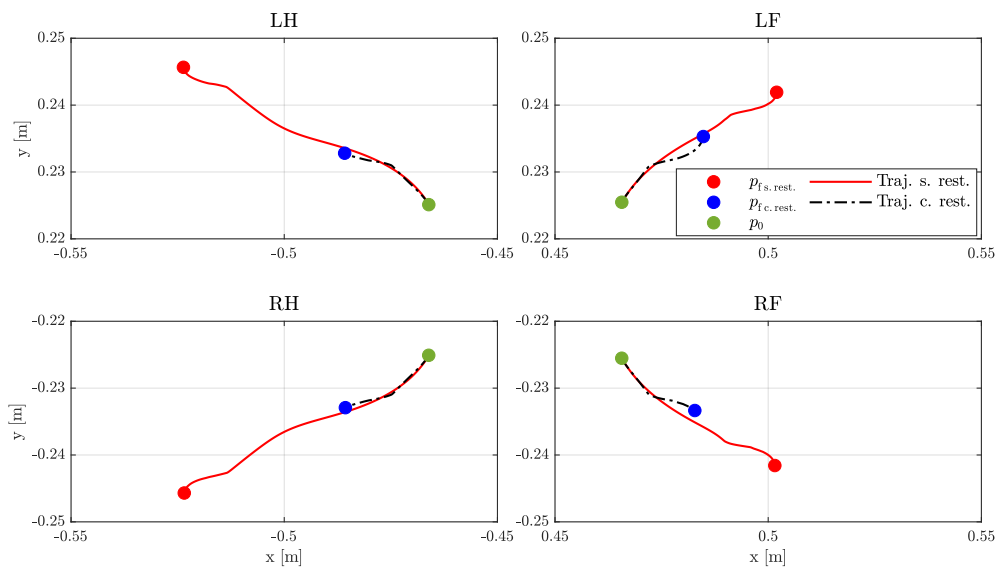
Desta forma, utilizando a Equação (5.3) como função custo, e considerando a restrições das GRFs, e ainda a restrição da posição da posição do CoM em  $\hat{z}$ , um novo ensaio de simulação foi realizado, de modo que é possível visualizar na Figura 34 o deslocamento vertical do centro de massa.

Figura 32 – Comportamento das forças de reação entre os pés e o solo das pernas LF e RF.



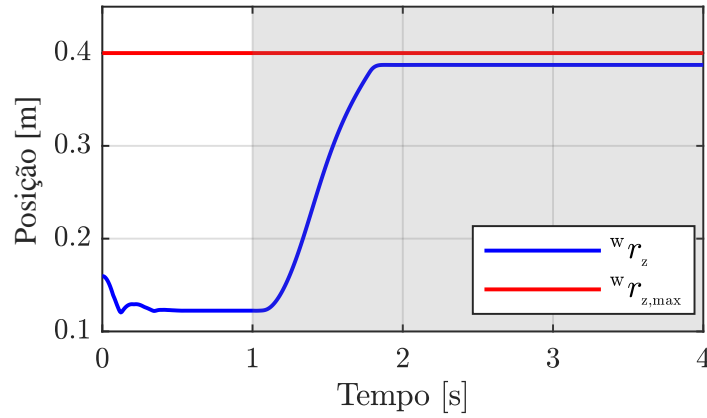
Fonte: próprio autor.

Figura 33 – Comparação das trajetórias dos pés para a situação sem e com restrição de forças de contato.



Fonte: próprio autor.

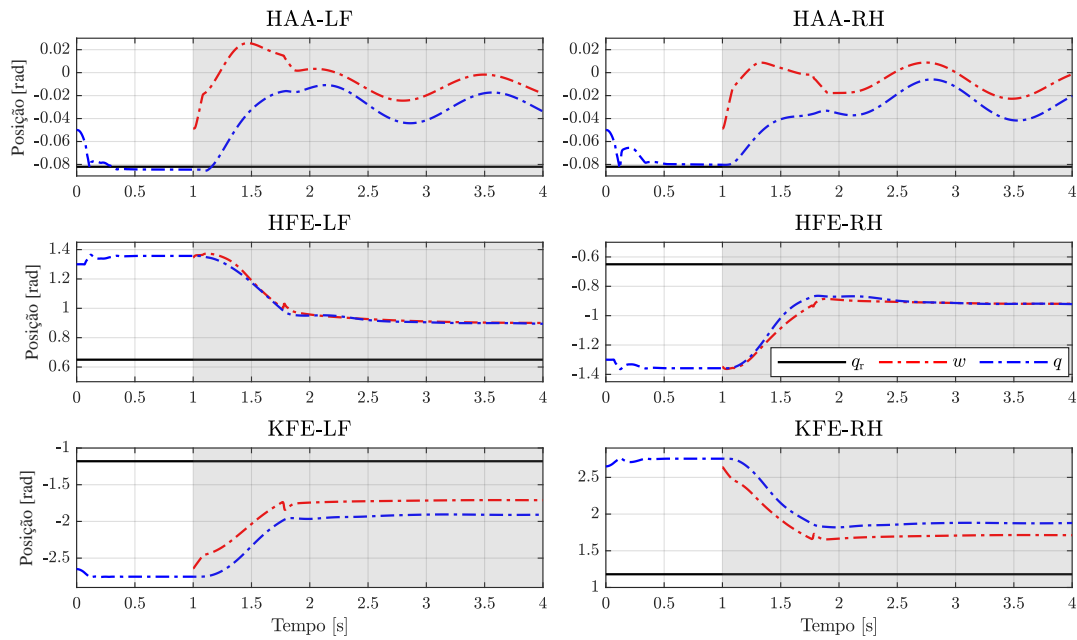
Figura 34 – Trajetória dos centro de massa ao considerar restrição de  ${}^w r_z$ .



Fonte: próprio autor.

Na Figura 35 se vê as referências moduladas  $\mathbf{w}$  para a situação de restrição na posição do CoM. É possível verificar que o RGC modifica os valores das referências  $\mathbf{q}_r$  de maneira a garantir que a restrição de  ${}^w r_z$  seja respeitada, mesmo que  $\mathbf{w}$  exibindo erro em regime permanente.

Figura 35 – Trajetórias das juntas  $\mathbf{q}$  e das referências moduladas  $\mathbf{w}$  considerando restrições na componente  ${}^w r_z$ .



Fonte: próprio autor.

O valor de  ${}^w r_{z,\max}$  poderia ser provido, por exemplo, por algum algoritmo de alto nível que seja capaz de verificar as condições do ambiente, e de modo *online*, alterar o valor da

restrição no problema de otimização.

Esta situação deixa claro que, ao usar o RGC, condições externas ao robô podem ser levadas em consideração para obter as referências das juntas factíveis.

### 5.3 SÍNTESE DO CAPÍTULO

Neste Capítulo, os resultados obtidos através das simulações das duas propostas de implementação do *Reference Governor Control*, implementado via MPC foram descritos. Nas duas situações, as arquiteturas propostas foram viáveis de implementação e garantiram uma performance segura de *stand up*.

Apesar de serem funcionais, as duas propostas devem ser melhoradas. Os horizontes de predição utilizados são pequenos em relação ao objetivo da tarefa, sendo necessário aumentá-los, entretanto, o problema de otimização é bastante sensível à mudança dessas variáveis. Com horizontes pequenos, o modelo de predição é suficientemente representativos, de modo que o controlador apresenta um bom desempenho. Todavia, ao aumentar o horizonte de predição, as diferenças entre o modelo proposto e a dinâmica do sistema se tornam mais evidentes, de modo que as ações de controle geradas pelo MPC não são capazes de controlar a tarefa. Assim, ao aprimorar o modelo, tornando-o mais representativo, será possível melhorar ainda mais a performance do controlador da tarefa de *self-righting*.

As questões envolvendo tempo de processamento não eram objetivos desta dissertação, mesmo assim, os tempos utilizados pelo *solver* no cálculo das ações de controle foram obtido e são sumarizados nas Tabelas 9 e 10:

Tabela 9 – Abordagem RGC-PD+P: Tempo de cálculo da solução do problema de otimização

	F. Custo Equação (5.2)	F. Custo Equação (5.3)	F. Custo Equação (5.4)
$\bar{t}$ [s]	0,0635	0,0488	0,0061
$t_{\max}$ [s]	0,1174	0,1142	0,1312
$\sigma$ [s]	0,0105	0,0090	0,0110

Fonte: próprio autor

Observa-se que estes valores estão muito acima do tempo de amostragem  $t_s$ , 10 ms. Em simulações, esta questão do tempo de processamento não afeta na realização da tarefa, uma vez que o software utilizado consegue lidar internamente com este problema, de modo a corrigi-lo. Entretanto, no presente estado, a aplicação das propostas em sistemas reais é inviável. Duas

Tabela 10 – Abordagem RGC-MdlSimp: Tempo de execução da solução do problema de otimização

	F. Custo Equação (5.3)	F. Custo Equação (5.4)	F. Custo Equação (5.5)
$\bar{t}$ [s]	0,0564	0,0534	0,0566
$t_{\max}$ [s]	0,1079	0,1150	0,1097
$\sigma$ [s]	0,0084	0,0081	0,0085

Fonte: próprio autor

possíveis soluções podem ser empregadas na para este problema: optar por um *solver* que lide melhor com este tipo de problema, como o *Operator Splitting Quadratic Program* (OSQP) (STELLATO et al., 2017); ou aumentar o tempo de amostragem. Entretanto, ao optar por este último, o modelo de predição pode ser deteriorado, uma vez que é assumido que os Jacobiano são constantes ao longo dos horizontes, necessitando então de uma análise mais aprofundada.

Realizando uma comparação direta entre a abordagem RGC-PD+P com RGC-MdlSimp é possível elencar algumas vantagens e desvantagens de cada uma das arquiteturas propostas. Em relação aos estados, a abordagem RGC:PD+P apresenta maior número do que a abordagem RGC-MdlSimp. Isso se deve ao fato de que para representar a estrutura da dinâmica em malha fechada do sistema, é necessário incorporar os estados internos do controlador. Ainda em relação aos estados internos do PD, é necessário a inclusão de um estimado para encontra-los, de modo que é necessário a inclusão de mais uma estrutura na malha de controle.

Já considerando as restrições do controlador interno, a abordagem RGC-PD+P lida de forma direta com elas, possibilitando utilizar formulações já estabelecidas na literatura. Já na abordagem RGC-MdlSimp, todas as restrições foram obtidas utilizando manipulações algébricas, consequentemente necessitando de um conhecimento mais aprofundando do sistema, ou seja, de que maneira uma grandeza afeta nas demais.

Uma das vantagens das propostas realizadas neste trabalho é a modularidade, uma vez que não é preciso mudar nenhuma característica interna da planta para sua utilização.

Dito isso, no Capítulo 6, as considerações finais deste trabalho são expostas, bem como as propostas de trabalhos futuros.



## 6 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho teve como principal objetivo estabelecer uma forma capaz de fechar a malha e controlar a tarefa de *self-righting* de robôs quadrúpedes. Assim, foi estabelecida uma estrutura utilizando a arquitetura de *Reference Governor Control*, implementada via Controlador Preditivo Baseado em Modelo, como solução.

No decorrer do trabalho, foram propostos dois modelos de predição distintos para serem usados para a solução do problema. No primeiro, os modelos do controlador interno e da planta foram obtidos de forma separadas e, posteriormente, encontrada a dinâmica de malha fechada. Ou seja, são dois modelos de predição, um para a dinâmica imposta pelo controlador do *loop* interno e outro para a planta. Nesta abordagem, a tratativa da restrição da ação de controle é realizada de forma direta a partir do seu modelo.

Já o segundo modelo, foi obtido através de manipulações algébricas da dinâmica linear do centroide e do modelo de impedância das juntas. A partir desta abordagem, se comparado à proposta anterior, é obtido um modelo reduzido, porém as restrições não são descritas de forma direta.

As arquiteturas propostas foram usadas para realizar a sub tarefa de *stand up* do robô quadrúpede HyQ2Max, se mostrando efetivas quando consideradas as restrições de torque nas juntas. Ao comparar as estratégias, o modelo reduzido realiza a tarefa na metade do tempo, mesmo considerando restrições conservativas para os torques. Questões de tempo real não eram requisitos neste trabalho, entretanto, para estas duas situações, os tempos de cálculo foram avaliados, onde foi verificado que é necessário diminuir o tempo despendido pelo *solver*. Como já mencionado no Capítulo 5, este problema pode ser sanado ao utilizar um *solver* mais eficiente, ficando como sugestão de trabalho futuro.

Também foi possível inserir restrições das forças de reação entre o solo e o pé do robô, que é de suma importância para evitar escorregamentos e quedas. Nesta situação o modelo obtido se mostrou sensível ao período de amostragem, de modo que foi necessário aumentar a frequência do controlador e aplicar solução da proposta. Ao realizá-lo, o RGC teve sucesso, garantindo que os pés do robô permanecessem estáticos. Uma comparação foi realizada entre um problema de otimização com restrição das GRF's e um problema com restrição apenas de torques, verificando a eficácia da utilização das restrições de forças.

Restrições espaciais também foram implementadas mostrando que, caso o ambiente imponha algum tipo de restrição espacial, a solução proposta é capaz de lidar com elas e reerguer o robô até uma altura segura.

Como já mencionado, o problema de otimização é sensível ao aumento dos horizontes de predição e controle, tornando a solução do problema infactível em alguns casos. Uma solução para este caso, que não era foco desta dissertação, é utilizar modelos aperfeiçoados que melhorem as predições e, conseqüentemente, obtenham uma performe ainda melhor da tarefa. A inclusão de alguma estratégia que considere questões de custo garantido pode também ser tema de algum trabalho futuro, pois insere robustez ao controle.

Ainda em relação à trabalhos futuros, é válida o estudo da inclusão das dinâmicas angulares ao modelo. Ao inserir tal comportamento, as pequenas assimetrias encontradas nas trajetórias das juntas dianteiras e traseiras poderiam ser anuladas. Também, esta dinâmica é necessária para a realização das demais etapas da tarefa de *self-righting*, principalmente o rolamento.

Durante o desenvolvimento deste trabalho, constatou-se que existe a necessidade da obtenção de alguma estrutura que seja capaz de detectar os contatos do robô com o solo. Na literatura, este problema é majoritariamente estudado para observar os contatos do pés com o solo durante as caminhada, entretanto, durante a sub tarefa de rolamento, outras partes do corpo tocam o chão, de modo que é necessário estimar estes pontos. Assim, encontra-se um problema que deve ser tratado em trabalhos futuros.

Destaca-se ainda que a arquitetura proposta fornece uma ampla customização no que diz respeito à função custo, uma vez que utilizando os estados considerados nos modelos, seria possível ponderar velocidade, aceleração, momento linear, energia e etc. ficando a critério do projetista determinar qual é a melhor para o seu objetivo.

Vê-se, por todos estes pontos acima e pelos resultados apresentados, que a proposta de arquitetura de controle de *self-righting* contida neste trabalho cumpriu os requisitos definidos nos objetivos. Identifica-se como uma solução que utiliza conceitos já consolidados na teoria, mas que sua estruturação é inédita e demonstra perspectivas de expansão tanto no sentido teórico como de aplicação prática.

## REFERÊNCIAS

- ACKERMAN, E. **iRobot Completely Redesigns Its Floor Care Robots With New m6 and s9**. 2019. Acesso em: 08 jul. 2020. Disponível em: <<https://spectrum.ieee.org/automaton/robotics/home-robots/irobot-completely-redesigns-its-floor-care-robots-with-new-m6-and-s9>>.
- BACHELIER, M.; CHEMORI, A.; KRUT, S. A control law for human like walking biped robot sherpa based on a control and a ballistic phase-application on the cart-table model. In: IEEE. **Humanoids 2008-8th IEEE-RAS International Conference on Humanoid Robots**. [S.l.], 2008. p. 132–137.
- BEMPORAD, A. Reference governor for constrained nonlinear systems. **IEEE Transactions on Automatic Control**, IEEE, v. 43, n. 3, p. 415–419, 1998.
- BEMPORAD, A. et al. The explicit linear quadratic regulator for constrained systems. **Automatica**, Elsevier, v. 38, n. 1, p. 3–20, 2002.
- BEMPORAD, A.; MOSCA, E. Constraint fulfilment in feedback control via predictive reference management. In: IEEE. **Proc. 3rd IEEE Conf. on Control Applications**. [S.l.], 1994. v. 3, p. 1909–1914.
- BENCATEL, R. et al. Reference governor strategies for vehicle rollover avoidance. **IEEE Transactions on Control Systems Technology**, IEEE, v. 26, n. 6, p. 1954–1969, 2017.
- BLET, N. et al. Nonlinear mpc versus mpc using on-line linearisation—a comparative study. **IFAC Proceedings Volumes**, Elsevier, v. 35, n. 1, p. 147–152, 2002.
- CAMACHO, E. F.; BORDONS, C. **Model predictive control**. [S.l.]: Springer, London, 2007.
- CASAREZ, C. S.; FEARING, R. S. Dynamic terrestrial self-righting with a minimal tail. In: IEEE. **2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**. [S.l.], 2017. p. 314–321.
- CASAVOLA, A.; MOSCA, E.; PAPINI, M. Control under constraints: an application of the command governor approach to an inverted pendulum. **IEEE Transactions on control systems technology**, IEEE, v. 12, n. 1, p. 193–204, 2004.
- DINI, N.; MAJD, V. J. Sliding-mode tracking control of a walking quadruped robot with a push recovery algorithm using a nonlinear disturbance observer as a virtual force sensor. **Iranian Journal of Science and Technology, Transactions of Electrical Engineering**, Springer, p. 1–25, 2019.
- DO, H. M.; CHOI, J. Y. Design of reference governor for linear systems with input constraints. **IFAC Proceedings Volumes**, Elsevier, v. 38, n. 1, p. 928–933, 2005.
- DYNAMICS, B. **LS3-Legged Squad Support System**. 2012. Acesso em: 08 jul. 2020. Disponível em: <<https://youtu.be/R7ezXBEBE6U>>.
- DYNAMICS, B. **LEGACY ROBOTS**. 2020. Acesso em: 26 jun. 2020. Disponível em: <<https://www.bostondynamics.com/legacy>>.
- ELLERY, A. **Planetary rovers: robotic exploration of the solar system**. [S.l.]: Springer, 2015.

- ETH. **Robot Dynamics Lecture Notes**. 2017. Acesso em: 22 jun. 2019. Disponível em: <<https://www.ethz.ch/content/dam/ethz/special-interest/mavt/robotics-n-intelligent-systems/rsl-dam/documents/RobotDynamics2016/RD2016script.pdf>>.
- FEATHERSTONE, R.; ORIN, D. E. Dynamics. In: **Springer Handbook of Robotics**. [S.l.]: Springer, 2016. p. 37–66.
- GARONE, E.; CAIRANO, S. D.; KOLMANOVSKY, I. Reference and command governors for systems with constraints: A survey on theory and applications. **Automatica**, Elsevier, v. 75, p. 306–328, 2017.
- GILBERT, E. G.; KOLMANOVSKY, I.; TAN, K. T. Discrete-time reference governors and the nonlinear control of systems with state and control constraints. **International Journal of robust and nonlinear control**, Wiley Online Library, v. 5, n. 5, p. 487–504, 1995.
- GRIMMINGER, F. et al. An open torque-controlled modular robot architecture for legged locomotion research. **IEEE Robotics and Automation Letters**, IEEE, v. 5, n. 2, p. 3650–3657, 2020.
- GUIZZO, E. **Fukushima Robot Operator Writes Tell-All Blog**. 2011. Acesso em: 08 jul. 2020. Disponível em: <<https://spectrum.ieee.org/automaton/robotics/industrial-robots/fukushima-robot-operator-diaries>>.
- IIT-DLSLAB. **hyq2max-description**. 2015. Acesso em: 08 jul. 2020. Disponível em: <<https://github.com/iit-DLSLab/hyq2max-description>>.
- JOHNSON, A. M. et al. Tail assisted dynamic self righting. In: **Adaptive Mobile Robotics**. [S.l.]: World Scientific, 2012. p. 611–620.
- KALABIĆ, U. et al. Constrained spacecraft attitude control on so (3) using reference governors and nonlinear model predictive control. In: IEEE. **2014 American Control Conference**. [S.l.], 2014. p. 5586–5593.
- KHORRAM, M.; MOOSAVIAN, S. A. A. Push recovery of a quadruped robot on challenging terrains. **Robotica**, Cambridge University Press, v. 35, n. 8, p. 1670–1689, 2017.
- KIEBOOM, J. v. d. **The Jacobin: Don't Panic**. BIOROB: EPFL Biorobotics Laboratory, 2019. Acesso em: 22 jun. 2019. Disponível em: <<https://ponyo.epfl.ch/gitlab/jesse.vandenkieboom/jacobian/raw/e57d7f79248d231803373fbf11969bd5db522330/jacobian.pdf>>.
- KLAUČO, M.; KALÚZ, M.; KVASNICA, M. Real-time implementation of an explicit mpc-based reference governor for control of a magnetic levitation system. **Control Engineering Practice**, Elsevier, v. 60, p. 99–105, 2017.
- KLAUČO, M.; KVASNICA, M. Control of a boiler-turbine unit using mpc-based reference governors. **Applied Thermal Engineering**, Elsevier, v. 110, p. 1437–1447, 2017.
- KLAUČO, M.; KVASNICA, M. **MPC-Based Reference Governors**. [S.l.]: Springer, 2019.
- KOTHARE, M. V.; BALAKRISHNAN, V.; MORARI, M. Robust constrained model predictive control using linear matrix inequalities. **Automatica**, Elsevier, v. 32, n. 10, p. 1361–1379, 1996.
- KOVAČ, M. et al. Steerable miniature jumping robot. **Autonomous Robots**, Springer, v. 28, n. 3, p. 295–306, 2010.
- KWON, S.; OH, Y. Real-time estimation algorithm for the center of mass of a bipedal robot with flexible inverted pendulum model. In: IEEE. **2009 IEEE/RSJ International Conference on Intelligent Robots and Systems**. [S.l.], 2009. p. 5463–5468.

- LEE, J.; HWANGBO, J.; HUTTER, M. Robust recovery controller for a quadrupedal robot using deep reinforcement learning. **arXiv preprint arXiv:1901.07517**, 2019.
- LEE, J. H. Model predictive control: Review of the three decades of development. **International Journal of Control, Automation and Systems**, Springer, v. 9, n. 3, p. 415, 2011.
- LI, C. et al. Mechanical principles of dynamic terrestrial self-righting using wings. **Advanced Robotics**, Taylor & Francis, v. 31, n. 17, p. 881–900, 2017.
- LI, C. et al. Cockroach-inspired winged robot reveals principles of ground-based dynamic self-righting. In: IEEE. **2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**. [S.l.], 2016. p. 2128–2134.
- LI, Y. et al. Research of mammal bionic quadruped robots: A review. In: IEEE. **2011 IEEE 5th International Conference on Robotics, Automation and Mechatronics (RAM)**. [S.l.], 2011. p. 166–171.
- LI, Z.; YU, S.; LIU, S. Dynamical balance optimization and control of quadruped robots under perturbing external force. **IFAC Proceedings Volumes**, Elsevier, v. 46, n. 20, p. 199–205, 2013.
- LIU, M. et al. Quadrupedal robots whole-body motion control based on centroidal momentum dynamics. **Applied Sciences**, Multidisciplinary Digital Publishing Institute, v. 9, n. 7, p. 1335, 2019.
- MACIEJOWSKI, J. M. **Predictive control: with constraints**. [S.l.]: Pearson education, 2002.
- MAGNI, L.; RAIMONDO, D. M.; ALLGÖWER, F. Nonlinear model predictive control. **Lecture Notes in Control and Information Sciences**, Springer, n. 384, 2009.
- MURPHY, R. R. A decade of rescue robots. In: IEEE. **2012 IEEE/RSJ International Conference on Intelligent Robots and Systems**. [S.l.], 2012. p. 5448–5449.
- ORIN, D. E.; GOSWAMI, A. Centroidal momentum matrix of a humanoid robot: Structure and properties. In: IEEE. **2008 IEEE/RSJ International Conference on Intelligent Robots and Systems**. [S.l.], 2008. p. 653–659.
- ORIN, D. E.; GOSWAMI, A.; LEE, S.-H. Centroidal dynamics of a humanoid robot. **Autonomous Robots**, Springer, v. 35, n. 2-3, p. 161–176, 2013.
- PARKER, S. P. **McGraw-Hill Dictionary of Scientific and Technical Terms (6th Ed.)**. USA: McGraw-Hill, Inc., 2003. ISBN 007042313X.
- PENG, S. et al. Motion planning and implementation for the self-recovery of an overturned multi-legged robot. **Robotica**, Cambridge University Press, v. 35, n. 5, p. 1107–1120, 2017.
- PERKINS, A. D.; MALCHANO, M.; TALEBINEJAD, S. **Systems and methods for robotic self-right**. [S.l.]: Google Patents, 2016. US Patent 9,308,648.
- PRETT, D. M.; RAMAKER, B. L.; CUTLER, C. R. **Dynamic matrix control method**. [S.l.]: Google Patents, 1982. US Patent 4,349,869.
- QIN, S. J.; BADGWELL, T. A. A survey of industrial model predictive control technology. **Control engineering practice**, Elsevier, v. 11, n. 7, p. 733–764, 2003.
- REMY, C. D. **Optimal exploitation of natural dynamics in legged locomotion**. Tese (Doutorado) — ETH Zurich, 2011.

- RICHALET, J. et al. Model predictive heuristic control. **Automatica (Journal of IFAC)**, Pergamon Press, Inc., v. 14, n. 5, p. 413–428, 1978.
- ROSSITER, J. A. **Model-based predictive control: a practical approach**. [S.l.]: CRC press, 2017.
- SARANLI, U.; RIZZI, A. A.; KODITSCHKE, D. E. Model-based dynamic self-righting maneuvers for a hexapedal robot. **The International Journal of Robotics Research**, SAGE Publications, v. 23, n. 9, p. 903–918, 2004.
- SEMINI, C. **HyQ – Design and Development of a Hydraulically Actuated Quadruped Robot**. Tese (Doutorado) — Istituto Italiano di Tecnologia (IIT) and University of Genova, 2010.
- SEMINI, C. et al. Design of the hydraulically actuated, torque-controlled quadruped robot hyq2max. **IEEE/ASME Transactions on Mechatronics**, IEEE, v. 22, n. 2, p. 635–646, 2016.
- SEMINI, C. et al. Design overview of the hydraulic quadruped robots. In: **The Fourteenth Scandinavian International Conference on Fluid Power**. [S.l.: s.n.], 2015. p. 20–22.
- SICILIANO, B. et al. **Robotics: modelling, planning and control**. [S.l.]: Springer Science & Business Media, 2010.
- SIEGWART, R.; NOURBAKHSI, I. R.; SCARAMUZZA, D. **Introduction to autonomous mobile robots**. [S.l.]: MIT press, 2011.
- SIMON, D.; HÄRKEGÅRD, O.; LÖFBERG, J. Angle of attack and load factor limiting in fighter aircraft using command governors. In: **AIAA Guidance, Navigation, and Control Conference**. [S.l.: s.n.], 2017. p. 1257.
- STELLATO, B. et al. OSQP: An operator splitting solver for quadratic programs. **ArXiv e-prints**, nov. 2017.
- STOETER, S. A.; PAPANIKOLOPOULOS, N. Kinematic motion model for jumping scout robots. **IEEE transactions on robotics**, IEEE, v. 22, n. 2, p. 397–402, 2006.
- TSUKAGOSHI, H. et al. Development of jumping & rolling inspector to improve the debris-traverse ability. **Journal of Robotics and Mechatronics**, v. 15, n. 5, p. 482–490, 2003.
- TZAFESTAS, S. G. **Introduction to mobile robot control**. [S.l.]: Elsevier, 2013.
- VILLARREAL, O. **Bridging Vision and Dynamic Legged Locomotion**. Tese (Doutorado) — Istituto Italiano di Tecnologia (IIT) and University of Genova, 2020.
- WALDRON, K. J.; SCHMIEDELER, J. Kinematics. In: **Springer Handbook of Robotics**. [S.l.]: Springer, 2016.
- WAN, Z.; KOTHARE, M. V. An efficient off-line formulation of robust model predictive control using linear matrix inequalities. **Automatica**, Elsevier, v. 39, n. 5, p. 837–846, 2003.
- WANG, L. **Model predictive control system design and implementation using MATLAB®**. [S.l.]: Springer Science & Business Media, 2009.
- WENSING, P. M.; PALMER, L. R.; ORIN, D. E. Efficient recursive dynamics algorithms for operational-space control with application to legged locomotion. **Autonomous Robots**, Springer, v. 38, n. 4, p. 363–381, 2015.
- WIEBER, P.-B.; TEDRAKE, R.; KUINDERSMA, S. Modeling and control of legged robots. In: **Springer handbook of robotics**. [S.l.]: Springer, 2016. p. 1203–1234.

WINKLER, A. W. **Optimization-based motion planning for legged robots**. Tese (Doutorado) — ETH Zurich, 2018.

WIT, C. C. de; SICILIANO, B.; BASTIN, G. **Theory of robot control**. [S.l.]: Springer Science & Business Media, 2012.

ZHANG, J. et al. Self-righting, steering and takeoff angle adjusting for a jumping robot. In: IEEE. **2012 IEEE/RSJ International Conference on Intelligent Robots and Systems**. [S.l.], 2012. p. 2089–2094.

## APÊNDICE A – *REFERENCE GOVERNOR CONTROL* COM LOOP DE CONTROLE INTERNO.

Para plantas que tem o número de controladores internos igual ao número de saídas e de ações de controle, o primeiro passo é descrever tais sistemas no formato de espaço de estados (KLAUČO; KVASNICA, 2019), como:

$$\begin{aligned}\dot{\mathbf{x}}_{r,i}(t) &= \mathbf{A}_{r,i}\mathbf{x}_{r,i}(t) + \mathbf{B}_{r,i}e_{r,i}(t); \\ u_{r,i} &= \mathbf{C}_{r,i}\mathbf{x}_{r,i}(t) + \mathbf{D}_{r,i}e_{r,i}(t).\end{aligned}\tag{A.1}$$

As matrizes  $\mathbf{A}_{r,i}$ ,  $\mathbf{B}_{r,i}$ ,  $\mathbf{C}_{r,i}$  e  $\mathbf{D}_{r,i}$  têm suas dimensões e seu valores determinadas pela estrutura de controle interno. A variável  $e_{r,i}$ , correspondente ao erro do  $i$ -ésimo controlador, e é determinada como sendo  $e(t) = w(t) - y(t)$ .  $\mathbf{x}_{r,i}$  são os estados internos, enquanto  $u_{r,i}$  correspondente a ação de controle que é efetivamente aplicada à planta. O próximo passo é realizar a discretização do espaço de estados da Equação (A.1) através de alguma técnica conhecida.

Para um sistema com  $p$  compensadores internos, é possível reorganizar os termos de modo a obter um novo espaço de estados, como:

$$\begin{aligned}\mathbf{x}_r[k+1] &= \mathbf{A}_r\mathbf{x}_r[k] + \mathbf{B}_r\mathbf{e}_r[k]; \\ \mathbf{u}[k] &= \mathbf{C}_r\mathbf{x}_r[k] + \mathbf{D}_r\mathbf{e}_r[k],\end{aligned}\tag{A.2}$$

em que

$$\mathbf{A}_r = \begin{bmatrix} \mathbf{A}_{r,1} & \dots & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{A}_{r,p} \end{bmatrix},\tag{A.3}$$

sendo equivalente para as demais matrizes da Equação (A.2).

Com o objetivo de determinar a dinâmica do sistema em malha fechada é definido um espaço de estados aumentado  $\tilde{\mathbf{x}}$  que incorpora tanto os estados internos dos controladores, quanto os da planta em si. A partir dos resultados obtidos em (KLAUČO; KVASNICA, 2019):



$$\tilde{\mathbf{x}}[k+1] = \mathbf{A}_{OL}\tilde{\mathbf{x}}[k] + \mathbf{B}_{OL}\mathbf{e}[k]; \quad (\text{A.4a})$$

$$\mathbf{u}[k] = \mathbf{C}_{OL,u}\tilde{\mathbf{x}}[k] + \mathbf{D}_{OL,u}\mathbf{e}[k]; \quad (\text{A.4b})$$

$$\mathbf{y}[k] = \mathbf{C}_{OL,y}\tilde{\mathbf{x}}[k] + \mathbf{D}_{OL,y}\mathbf{e}[k], \quad (\text{A.4c})$$

cujos termos são descritos abaixo:

$$\tilde{\mathbf{x}} = \begin{bmatrix} \mathbf{x}_r \\ \mathbf{x} \end{bmatrix}, \quad (\text{A.5a})$$

$$\mathbf{A}_{OL} = \begin{bmatrix} \mathbf{A}_r & \mathbf{0} \\ \mathbf{B}\mathbf{C}_r & \mathbf{A} \end{bmatrix}, \quad (\text{A.5b})$$

$$\mathbf{B}_{OL} = \begin{bmatrix} \mathbf{B}_r \\ \mathbf{B}\mathbf{D}_r \end{bmatrix}, \quad (\text{A.5c})$$

$$\mathbf{C}_{OL,u} = \begin{bmatrix} \mathbf{C}_r & \mathbf{0} \end{bmatrix}, \quad (\text{A.5d})$$

$$\mathbf{D}_{OL,u} = \mathbf{D}_r, \quad (\text{A.5e})$$

$$\mathbf{C}_{OL,y} = \begin{bmatrix} \mathbf{D}\mathbf{C}_r & \mathbf{C} \end{bmatrix}, \quad (\text{A.5f})$$

$$\mathbf{D}_{OL,y} = \mathbf{D}\mathbf{D}_r, \quad (\text{A.5g})$$

em que o termo  $\mathbf{x}$  são os estados da planta, com  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{C}$  e  $\mathbf{D}$  são os coeficientes do modelo dinâmico dela.

Considerando ainda que o erro é dado por:

$$\mathbf{e}[k] = \mathbf{w}[k] - \mathbf{y}[k]. \quad (\text{A.6})$$

Substituindo  $\mathbf{e}[k]$  da Equação (A.4c) pela igualdade da Equação (A.6) e realizando as devidas manipulações,  $\mathbf{y}[k]$  passa a ser definido como:

$$\mathbf{y}[k] = (\mathbf{I} + \mathbf{D}_{OL,y})^{-1}\mathbf{C}_{OL,y}\tilde{\mathbf{x}}[k] + (\mathbf{I} + \mathbf{D}_{OL,y})^{-1}\mathbf{D}_{OL,y}\mathbf{w}[k]. \quad (\text{A.7})$$

Após substituir a Equação (A.7) na Equação (A.6) o erro  $\mathbf{e}[k]$  passa a depender apenas dos valores dos estados  $\tilde{\mathbf{x}}[k]$  e da referência  $\mathbf{w}[k]$ . Logo, substituindo a expressão de  $\mathbf{e}[k]$  obtida

nas Equações (A.4a) e (A.4b), o espaço de estados em malha fechada é obtido como:

$$\mathbf{A}_{CL} = \mathbf{A}_{OL} - \mathbf{B}_{OL}(\mathbf{I} + \mathbf{D}_{OL,y})^{-1}\mathbf{C}_{OL,y}, \quad (\text{A.8a})$$

$$\mathbf{B}_{CL} = \mathbf{B}_{OL} - \mathbf{B}_{OL}(\mathbf{I} + \mathbf{D}_{OL,y})^{-1}\mathbf{D}_{OL,y}, \quad (\text{A.8b})$$

$$\mathbf{C}_{CL,u} = \mathbf{C}_{OL,u} - \mathbf{D}_{OL,u}(\mathbf{I} + \mathbf{D}_{OL,y})^{-1}\mathbf{C}_{OL,y}, \quad (\text{A.8c})$$

$$\mathbf{D}_{CL,u} = \mathbf{D}_{OL,u} - \mathbf{D}_{OL,u}(\mathbf{I} + \mathbf{D}_{OL,y})^{-1}\mathbf{D}_{OL,y}, \quad (\text{A.8d})$$

$$\mathbf{C}_{CL,y} = (\mathbf{I} + \mathbf{D}_{OL,y})^{-1}\mathbf{C}_{OL,y}, \quad (\text{A.8e})$$

$$\mathbf{D}_{CL,y} = (\mathbf{I} + \mathbf{D}_{OL,y})^{-1}\mathbf{D}_{OL,y}. \quad (\text{A.8f})$$

Portanto, a representação em espaços de estados, do sistema em malha fechada passa a ser descrito por:

$$\tilde{\mathbf{x}}[k+1] = \mathbf{A}_{CL}\tilde{\mathbf{x}}[k] + \mathbf{B}_{CL}\mathbf{w}[k]; \quad (\text{A.9a})$$

$$\mathbf{u}[k] = \mathbf{C}_{CL,u}\tilde{\mathbf{x}}[k] + \mathbf{D}_{CL,u}\mathbf{w}[k]; \quad (\text{A.9b})$$

$$\mathbf{y}[k] = \mathbf{C}_{CL,y}\tilde{\mathbf{x}}[k] + \mathbf{D}_{CL,y}\mathbf{w}[k]. \quad (\text{A.9c})$$

## APÊNDICE B – CONSIDERAÇÕES DE MODELAGEM

### B.1 MATRIZES DE ROTAÇÃO

As matrizes de rotação obtidas de (WALDRON; SCHMIEDELER, 2016) são:

$$\mathbf{R}_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix}, \quad (\text{B.1})$$

$$\mathbf{R}_y(\theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}, \quad (\text{B.2})$$

$$\mathbf{R}_z(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (\text{B.3})$$

### B.2 CONVENÇÃO DE DENAVI-HARTEMBERG

Como já mencionado na Seção 3.3.1, a convenção de Denavit-Hartenberg utilizada ao longo neste trabalho foi obtida de Wit, Siciliano e Bastin (2012), onde as seguintes considerações são feitas

- O mecanismo tem  $\mathcal{N}_B$  partes móveis, numeradas de 1 até  $\mathcal{N}_B$ . A base fixa é numerada como sendo 0;
- As juntas do mecanismo são numeradas de 1 até  $\mathcal{N}_B$ , com a junta  $i$  localizadas entre os componentes  $i - 1$  e  $i$ .

A base de referência de cada componente do mecanismo segue a seguinte convenção:

- O eixo  $\hat{z}_i$  está localizado ao longo do eixo de movimento da junta  $i$ ;
- O eixo  $\hat{x}_i$  está localizado ao longo da comum normal entre os eixos  $\hat{z}_i$  e  $\hat{z}_{i+1}$ .
- O eixo  $\hat{y}_i$  é inserido de forma a completar o sistema cartesiano respeitando a regra da mão direita.

Utilizando esta base, os quatro parâmetros são obtidos:

- $a_i$ : distância entre  $\hat{z}_{i-1}$  e  $\hat{z}_i$  ao longo de  $\hat{x}_{i-1}$ ;
- $\alpha_i$ : ângulo entre  $\hat{z}_{i-1}$  e  $\hat{z}_i$  ao longo de  $\hat{x}_{i-1}$ ;
- $d_i$ : distância entre  $\hat{x}_{i-1}$  e  $\hat{x}_i$  ao longo de  $\hat{z}_i$ ;
- $\theta_i$ : ângulo entre  $\hat{x}_{i-1}$  e  $\hat{x}_i$  ao longo de  $\hat{z}_i$ .

Substituindo as variáveis acima na sequência de translações de rotações dadas por:

$${}^{i-1}\mathbf{T}_i = Rot(\hat{x}_{i-1}, \alpha_i) Trans(\hat{x}_{i-1}, a_i) Rot(\hat{z}_i, \theta_i) Trans(\hat{z}_i, d_i); \quad (\text{B.4})$$

é obtida a matriz de transformada homogênea entre os *links*  $i - 1$  e  $i$ :

$${}^{i-1}\mathbf{T}_i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & a_i \\ \sin \theta_i \cos \alpha_i & \cos \theta_i \cos \alpha_i & -\sin \alpha_i & -d_i \sin \alpha_i \\ \sin \theta_i \sin \alpha_i & \cos \theta_i \sin \alpha_i & \cos \alpha_i & -d_i \cos \alpha_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (\text{B.5})$$

## APÊNDICE C – CONSIDERAÇÕES SOBRE AS SIMULAÇÕES

### C.1 HARDWARE E SOFTWARE.

As simulações foram realizadas usando o Matlab R2020a, usando o ambiente de simulação *Simscape Multibody*. O computador utilizado para realizar as simulações possui as seguintes configurações:

- Processado: Intel(R) Core(TM) i5-8300h CPU @ 2.30Ghz;
- Memória: 2X 8GB - DUAL CHANNEL MEMÓRIA DDR4 (2666 MHZ);
- Placa de video: NVIDIA® GEFORCE® GTX 1060 GPU (6GB GDDR5);
- Armazenamento: SSD M.2 NVME SAMSUNG 970 EVO - 500GB -(3.400 MB/S);
- Sistema operacional: Windows 10;

### C.2 ABORDAGEM RGC-PD+P: PSEUDO-CÓDIGO.

---

**Algoritmo 1** Pseudo-código da abordagem RGC-PD+P.

---

**Entrada:**  $\mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\Theta}, \mathbf{x}_r, t$

**Saída:**  $\mathbf{u}[k]$

**se**  $t = t_s$  **então**

Atualiza  $\mathbf{J}_{c,i}(\mathbf{q})$

Atualiza  $\mathbf{J}_{CoM,i}(\mathbf{q})$

Estima  ${}^w\dot{\mathbf{r}}$  usando a Equação (4.6)

Obtêm as matrizes da dinâmica da planta da Equação (4.13)

Obtêm as matrizes do espaço de estado da Equação (A.5)

Obtêm as matrizes do espaço de estado da Equação (A.8)

Atualiza  $\mathbf{G}$  e  $\boldsymbol{\Phi}$

Calcula o problema de otimização da Equação (4.19)

$\mathbf{u}[k] = \mathbf{u}[k-1] + \Delta\mathbf{u}[k]$

$t = 0$

**fim**

---

### C.3 ABORDAGEM RGC-PD+P: PARÂMETROS DE SIMULAÇÃO.

Tabela 11 – Parâmetros do MPC: abordagem RGC-PD+P.

Parâmetro	Valor	Descrição
$t_s$	0,01 s	Tempo de amostragem
N	12	Horizonte de predição
M	6	Horizonte de controle
$\mathbf{Q}_q$	$0,04 \times \mathbf{I}_{12}$	Matriz de ponderação $\mathbf{q}_r - \mathbf{q}$
$\mathbf{Q}_q^*$	$0,01 \times \mathbf{I}_{12}$	-
$\mathbf{Q}_w$	$0,03 \times \mathbf{I}_{12}$	Matriz de ponderação $\mathbf{q}_r - \mathbf{w}$
$\mathbf{Q}_w^*$	$0,02 \times \mathbf{I}_{12}$	-
$\mathbf{Q}_{\Delta w}$	$140 \times \mathbf{I}_{12}$	Matriz de ponderação dos incrementos $\Delta \mathbf{w}$
$\bar{\tau}$	80 Nm	Torque máximo
$\underline{\tau}$	-80 Nm	Torque mínimo
$\mu$	0,5	Coefficiente de atrito estático
$\mathbf{g}$	$[0 \ 0 \ -9,81]^T \text{ m/s}^2$	Vetor gravitacional

\* função custo Eq. (5.4).

Fonte: próprio autor

Tabela 12 – Parâmetros Controlador PD.

Parâmetro	Valor	Descrição
$K_p$	300	Ganho proporcional
$K_d$	10	Ganho Derivativo
$N_f$	20	Coefficiente do filtro passa baixa

Fonte: próprio autor

Tabela 13 – Parâmetros Filtro de Kalman.

Parâmetro	Valor	Descrição
$t_o$	0,001 s	Tempo de amostragem do observador
$\mathbf{QN}$	$\mathbf{I}_{12}$	Matriz de ponderação
$\mathbf{RN}$	$\mathbf{I}_{12}$	Matriz de ponderação

Fonte: próprio autor

## C.4 ABORDAGEM RGC-MDLSIMP: PSEUDO-CÓDIGO.

---

**Algoritmo 2** Pseudo-código da abordagem RGC-MddlSimp.

---

**Entrada:**  $\mathbf{q}, \dot{\mathbf{q}}, \Theta, \mathbf{x}_r, t$ **Saída:**  $\mathbf{u}[k]$ **se**  $t = t_s$  **então**    Atualiza  $\mathbf{J}_{c,i}(\mathbf{q})$     Atualiza  $\mathbf{J}_{\text{CoM},i}(\mathbf{q})$     Estima  ${}^w\dot{\mathbf{r}}$  usando a Equação (4.6)

Obtêm as matrizes da dinâmica da planta da Equação (4.22)

    Atualiza  $\mathbf{G}$  e  $\Phi$ 

Calcula o problema de otimização da Equação (4.29)

 $\mathbf{u}[k] = \mathbf{u}[k-1] + \Delta\mathbf{u}[k]$      $t = 0$ **fim**


---

## C.5 PARÂMETROS DE SIMULAÇÃO DA ABORDAGEM RGC-MDLSIMP COM RESTRIÇÃO EM TORQUE

Tabela 14 – Abordagem RGC-MdlSimp: parâmetros do MPC.

Parâmetro	Valor	Descrição
$t_s$	0,01 s	Tempo de amostragem
$N$	12	Horizonte de predição
$M$	6	Horizonte de controle
$\mathbf{K}_p$	$300 \times \mathbf{I}_{12}$	Matriz de ganhos proporcionais
$\mathbf{K}_d$	$10 \times \mathbf{I}_{12}$	Matriz de ganhos derivativos
$\mathbf{Q}_q$	$0,025 \times \mathbf{I}_{12}$	Matriz de ponderação $\mathbf{q}_r - \mathbf{q}$
$\mathbf{Q}_w$	$0,03 \times \mathbf{I}_{12}$	Matriz de ponderação $\mathbf{q}_r - \mathbf{w}$
$\mathbf{Q}_e$	$0,01 \times \mathbf{I}_{12}$	Matriz de ponderação $\mathbf{w} - \mathbf{q}$
$\mathbf{Q}_{\Delta w}$	$140 \times \mathbf{I}_{12}$	Matriz de ponderação dos incrementos $\Delta\mathbf{w}$
$\bar{\tau}$	80 Nm	Torque máximo
$\underline{\tau}$	-80 Nm	Torque mínimo
$\mu$	0,5	Coefficiente de atrito estático
$\mathbf{g}$	$[0 \ 0 \ -9,81]^T \text{ m/s}^2$	Vetor gravitacional

Fonte: próprio autor

## C.6 PARÂMETROS DE SIMULAÇÃO DA ABORDAGEM RGC-MDLSIMP COM RESTRIÇÃO EM TORQUE E FORÇAS DE REAÇÃO DO SOLO

Tabela 15 – Abordagem RGC-MdlSimp com restrição de forças: parâmetros do MPC.

Parâmetro	Valor	Descrição
$t_s$	0,004 s	Tempo de amostragem
N	10	Horizonte de predição
M	3	Horizonte de controle
$\mathbf{K}_p$	$300 \times \mathbf{I}_{12}$	Matriz de ganhos proporcionais
$\mathbf{K}_d$	$10 \times \mathbf{I}_{12}$	Matriz de ganhos derivativos
$\mathbf{Q}_q$	$0,07 \times \mathbf{I}_{12}$	Matriz de ponderação $\mathbf{q}_r - \mathbf{q}$
$\mathbf{Q}_w$	$0,09 \times \mathbf{I}_{12}$	Matriz de ponderação $\mathbf{q}_r - \mathbf{w}$
$\mathbf{Q}_{\Delta w}$	$140 \times \mathbf{I}_{12}$	Matriz de ponderação dos incrementos $\Delta \mathbf{w}$
$\bar{\tau}$	100 Nm	Torque máximo
$\underline{\tau}$	-100 Nm	Torque mínimo
$\bar{f}_z$	225 N	Força normal máxima
$\underline{f}_z$	50 N	Força normal mínima
$\mu$	0,5	Coefficiente de atrito estático
$\mathbf{g}$	$[0 \ 0 \ -9,81]^T \text{ m/s}^2$	Vetor gravitacional

Fonte: próprio autor