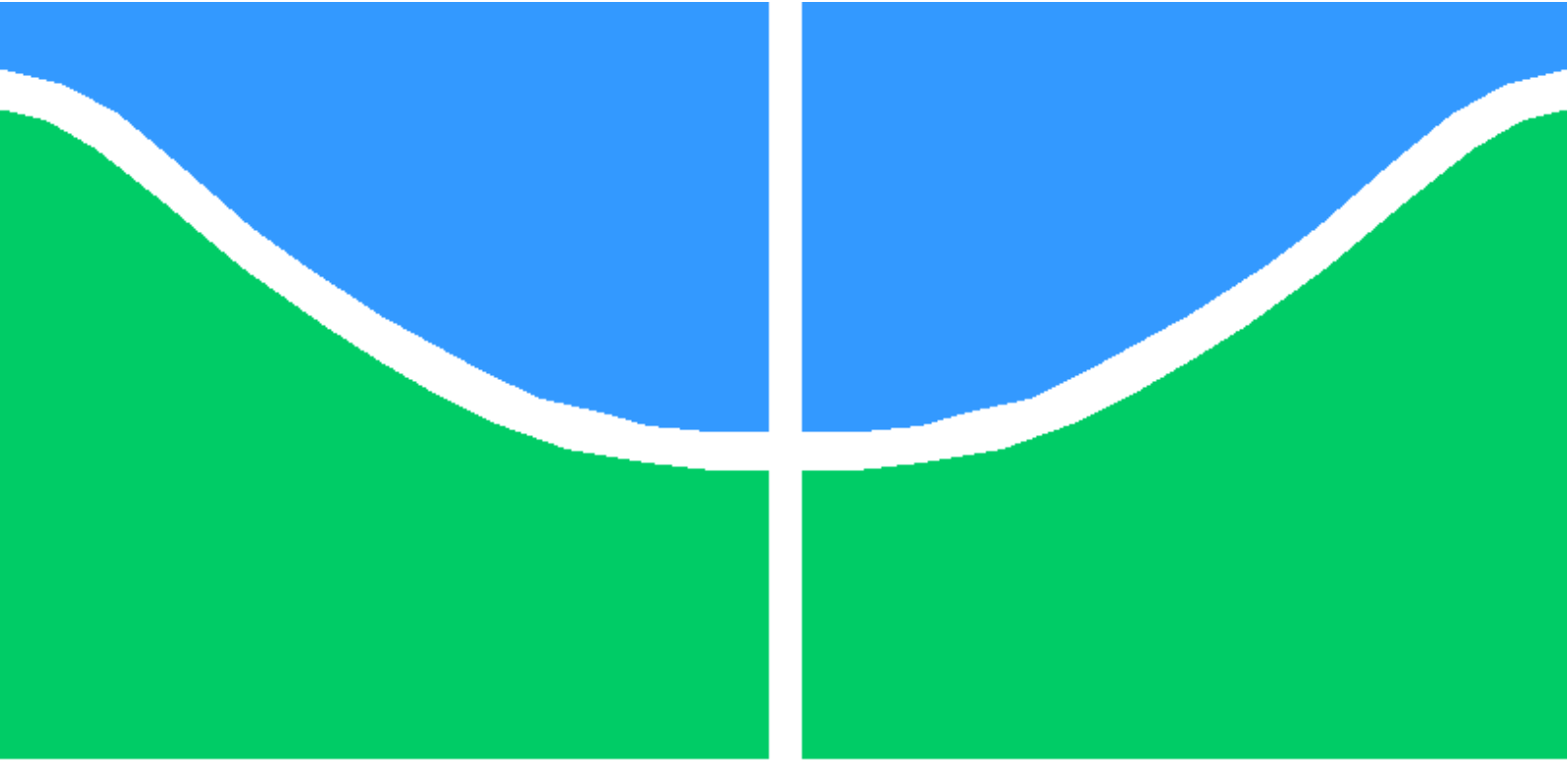


TRABALHO DE GRADUAÇÃO

CONTROLE DE ESTABILIDADE DE UM ROBÔ QUADRÚPEDE

Bruno Rodolfo de Oliveira Floriano

Brasília, agosto de 2017

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASILIA
Faculdade de Tecnologia

TRABALHO DE GRADUAÇÃO

CONTROLE DE ESTABILIDADE DE UM ROBÔ QUADRÚPEDE

Bruno Rodolfo de Oliveira Floriano

Relatório submetido ao Departamento de Engenharia Elétrica como requisito parcial para obtenção do grau de Engenheiro Eletricista

Banca Examinadora

Prof. Geovany Araújo Borges, ENE/UnB _____
Orientador

Prof. Alexandre Romariz, ENE/UnB _____
Co-orientador

Prof. , _____
Examinador externo

Prof. , _____
Examinador interno

Dedicatória

Dedicatória do autor 1

Bruno Rodolfo de Oliveira Floriano

Agradecimentos

A inclusão desta seção de agradecimentos é opcional e fica à critério do(s) autor(es), que caso deseje(em) inclui-la deverá(ao) utilizar este espaço, seguindo está formatação.

Bruno Rodolfo de Oliveira Floriano

RESUMO

O presente texto apresenta normas a serem seguidas por alunos do Curso de Engenharia Mecatrônica da Universidade de Brasília para redação de relatório na disciplina Projeto de Graduação 2. Tais normas foram aprovadas pela Comissão de Graduação do Curso de Engenharia Mecatrônica em julho/2005. São apresentadas instruções detalhadas para a formatação do trabalho em termos de suas partes principais.

ABSTRACT

The same as above, in english.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SÍMBOLOS

Símbolos Latinos

A	Área	[m^2]
Cp	Calor especifico a pressão constante	[kJ/kg.K]
h	Entalpia especifica	[kJ/kg]
$\dot{m}$	Vazão mássica	[kg/s]
T	Temperatura	[°C]
U	Coeficiente global de transferência de calor	[W/m^2.K]

Símbolos Gregos

α	Difusividade térmica	[m^2/s]
Δ	Variação entre duas grandezas similares	
ρ	Densidade	[m^3/kg]

Grupos Adimensionais

Nu	Número de Nusselt
Re	Número de Reynolds

Subscritos

amb	ambiente
ext	externo
in	entrada
ex	saída

Sobrescritos

$\cdot$	Variação temporal
$-$	Valor médio

Siglas

ABNT	Associação Brasileira de Normas Técnicas

Capítulo 1

Introdução

Este capítulo apresenta a principal motivação do trabalho de graduação. Os objetivos são claramente apresentados, visando assim satisfazer um conjunto de características prescritas para este trabalho. Por fim, o manuscrito é apresentado. (Este resumo é opcional)

1.1 Contextualização

Nas últimas décadas, o estudo e desenvolvimento de robôs com patas, especialmente de quadrúpedes, tem avançado consideravelmente. Devido às sua características bio-inspiradas, estes veículos apresentam alta mobilidade em terrenos irregulares, manejam obstáculos e podem se locomover em superfícies inclinadas tornando-os uma alternativa mais eficaz para a exploração e navegação de diversos terrenos quando comparados com veículos movidos a rodas, por exemplo [1, 2]. Isso faz com que estes robôs possam ser utilizados em diversas aplicações.

Neste sentido, os robôs quadrúpedes podem, por exemplo, ser amplamente utilizados no transporte de cargas, devido à possibilidade de distribuição de peso em suas pernas de suporte. Dentre outros, este é um dos objetivos do desenvolvimento de alguns destes robôs por parte da empresa norte-americana Boston Dynamics, desenvolvedora do *BigDog*, um robô quadrúpede criado para a locomoção em diversos terrenos, incluindo solos irregulares e superfícies inclinadas, capaz de suportar até 154 kg em terrenos horizontais [1]. Na Figura 1.1 podemos ver o *BigDog* se locomovendo sobre uma superfície com neve e inclinada.

Outra possível aplicação desta tecnologia é a substituição de seres humanos em atividades de risco tais como busca e resgate, construção, recuperação de ambientes em que houve desastre e exploração de ambientes inóspitos ou de difícil acesso. Estes são alguns objetivos traçados pelo Instituto Italiano de Tecnologia (Istituto Italiano di Tecnologia) no desenvolvimento do HyQ, um robô quadrúpede com atuadores hidráulicos e elétricos [3, 4]. A Figura 1.2 mostra tal robô construido.

Por fim, os quadrúpedes podem, ainda, ter como finalidade o entretenimento. Robôs como o AIBO, desenvolvido pela empresa japonesa Sony, são animais de estimação virtuais que visam o

Figura 1.1: Robô quadrúpede *BigDog* desenvolvido pela empresa Boston Dynamics

lazer [5]. Podemos observar uma imagem do AIBO na Figura 1.3.

Porém, para que um robô quadrúpede possa desenvolver as aplicações acima mencionadas, cumprindo os requisitos de suas funções, é necessário que, além de conseguir se movimentar de forma não cadenciada, isto é, garantindo um movimento contínuo, ele mantenha o equilíbrio de seu corpo, evitando o tombamento e garantindo a estabilidade. Dessa forma, manter a estabilidade permite que o sistema seja robusto contra distúrbios externos como empurrões ou ventos e contra irregulares no terreno como buracos e rugosidades.

Justamente pela importância da análise de estabilidade em robôs quadrúpedes que seu estudo começou logo em 1968 com os trabalhos de McGhee e Frank [6] sobre estabilidade estática e vêm se estendendo até os dias atuais com novos modelos e experimentos para análise de como robôs quadrúpedes podem se manter estáveis ainda que em alta velocidade ou em terrenos irregulares.

1.2 Definição do problema

Desde 2005, alunos e professores de Engenharia Elétrica e de Engenharia Mecatrônica da Universidade de Brasília (UnB) vêm desenvolvendo e aprimorando uma plataforma quadrúpede no Laboratório de Robótica e Automação (LARA). Desde então, várias abordagens já foram trabalhadas na plataforma, passando por diversas modificações em seus componentes [7]. Contudo, pouco foi trabalhada a questão referente ao controle de estabilidade do robô.

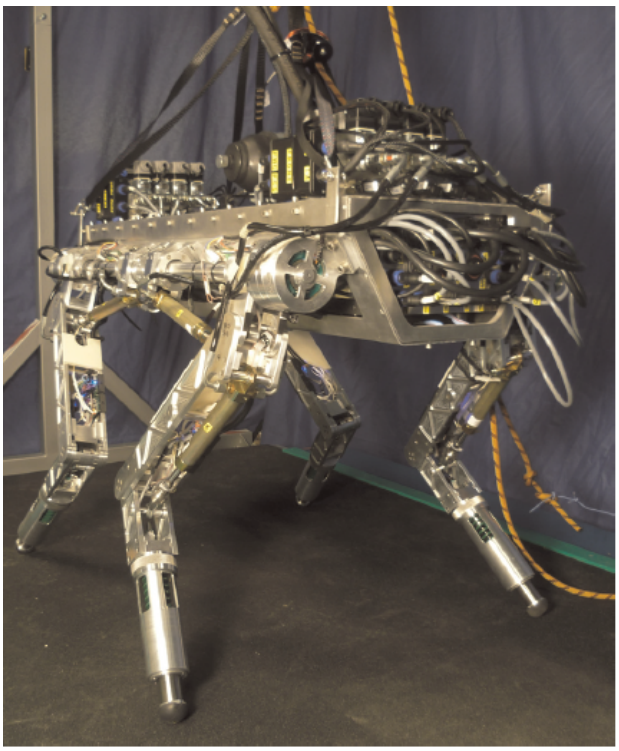


Figura 1.2: Robô quadrúpede HyQ

1.3 Objetivos do projeto

Atualmente a plataforma quadrúpede do LARA apresenta um movimento balístico satisfatório, desenvolvido por Porphirio e Santana [8], isto é, o robô apresenta capacidade de se locomover de forma não cadenciada, mas não possui um sistema de controle de equilíbrio que permita a este manter-se estável com a aplicação de distúrbios externos.

Dessa forma, os objetivos deste trabalho são desenvolver um sistema de equilíbrio para a plataforma quadrúpede baseada nas leituras do acelerômetro disponível no robô, implementá-lo em seu sistema embarcado e realizar experimentos para verificar a resposta ao distúrbio.

1.4 Apresentação do manuscrito

No capítulo 2 é feita uma revisão bibliográfica sobre o tema de estudo. Em seguida, o capítulo 3 descreve a metodologia empregada no desenvolvimento do projeto. Resultados experimentais são discutidos no capítulo 4, seguido das conclusões no capítulo 5. Os anexos contém material complementar.

Figura 1.3: Robô quadrúpede de entretenimento AIBO da Sony

Capítulo 2

Revisão Bibliográfica

Resumo opcional.

2.1 Introdução

Como mencionado no Capítulo 1, os estudos de estabilidade em robôs com pernas, em especial de robôs quadrúpedes, começaram em 1968 com os trabalhos de McGhee e Frank com a definição de estabilidade estática [6]. Com o passar do tempo, novas definições para a estabilidade estática foram sendo desenvolvidas, ao mesmo tempo que os estudos de estabilidade dinâmica começaram a ser feitos [9].

Desde então, diversos métodos de controle de estabilidade tem sido trabalhados na construção de robôs quadrúpedes, desde modelos cinemáticos, utilizando apenas as posições e velocidades do robô, até modelos dinâmicos mais complexos, que analisam as forças e momentos atuantes sobre o robô.

2.2 Estabilidade Estática

A estabilidade estática tem como principal característica a independência de fatores dinâmicos, isto é, ela não considera a atuação de forças externas, com exceção da gravidade, de momentos resultantes ou de componentes inerciais. Porém, conforme o robô aumenta sua velocidade os efeitos dinâmicos sobre o mesmo passam a ficar cada vez mais relevantes de forma que a estabilidade estática não garante o equilíbrio da máquina. Da mesma forma, o aumento de carga sobre o robô ou a presença de superfícies irregulares ou inclinadas tornam este tipo de controle mais ineficiente.

Contudo, estudar a estabilidade estática ainda é de suma importância no controle de equilíbrio de quadrúpedes, seja para melhor entendimento de todo o escopo deste estudo, seja pra aplicações restritas onde a aplicação da estabilidade estática seja suficiente para cumprir os requisitos propostos.

2.2.1 Métodos Geométricos

Os métodos geométricos utilizam a geometria proveniente da configuração das patas do robô a cada instante para definir a sua estabilidade e o quão perto dele se tornar instável (a margem de estabilidade).

Segundo McGhee e Frank [6] um robô movido a patas sobre uma superfície horizontal é estaticamente estável se e somente se a projeção vertical do centro de gravidade (COG) da máquina está dentro do polígono de suporte, onde este é definido como o polígono convexo formado pela conexão das patas em contato com o solo. Podemos observar na Figura 2.1 alguns possíveis polígonos de suporte formados por um robô quadrúpede durante sua caminhada. Na Figura 2.1(a) uma das patas não está em contato com o solo, de forma que o polígono resultante é um triângulo. Na Figura 2.1(b), todas as patas estão em contado com o solo, mas as patas 1 e 3 se encontram mais próximas, formando um trapézio. Por fim, na Figura 2.1(c), todas as patas estão em contato com o solo mas 1 e 3 estão mais afastadas, formando, dessa forma, um paralelogramo.

Em todos os casos da Figura 2.1 a projeção do centro de gravidade está dentro do polígono, indicando que o robô encontra-se estaticamente estável.

Além da definição de estabilidade, McGhee e Frank apresentam em [6] a definição de margem de estabilidade. Para este caso, ela é definida como a menor distância entre a projeção do COG e qualquer ponto da fronteira do polígono. Dessa forma, por exemplo, a margem de estabilidade da Figura 2.1(a) é menor que da Figura 2.1(c) pois no primeiro caso a projeção do COG se encontra bem mais próximo de uma das fronteiras do polígono.

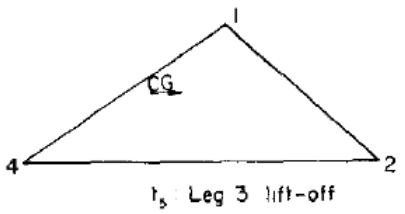
Posteriormente, novas difinições foram feitas para a margem de estabilidade estática baseada neste trabalho de McGhee e Frank [9]. A Margem de Estabilidade Longitudinal (S_{LSM}), por exemplo, é definida como a menor distância entre a projeção vertical do COG e a fronteira dianteira ou traseira do polígono de suporte, ou seja, a menor distância deve ser tomada apenas no eixo longitudinal do polígono. Isto é um método que facilita os cálculos da margem de estabilidade.

Um outro exemplo é a Margem de Estabilidade Longitudinal de Caranguejo (S_{CLSM}) onde, ao invés de se considerar o eixo longitudinal, deve-se considerar o eixo de direção do movimento do robô. Este método leva em conta a não-idealidade dos veículos, que não necessariamente se locomovem através de seu eixo longitudinal. A Figura 2.2 mostra uma representação da diferença das margens Longitudinal e de Caranguejo.

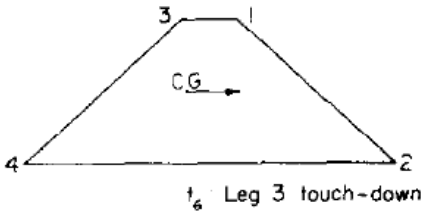
2.2.2 Métodos Energéticos

Os métodos energéticos calculam a energia necessária para tombar o robô para definir sua margem de estabilidade e, consequentemente, se ele se encontra instável ou não.

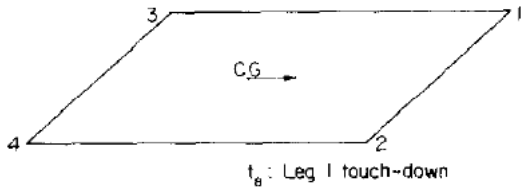
Proposto em 1985 por Messuri, a Margem de Estabilidade por Energia (S_{ESM}) é definida como a energia potencial mínima necessária para tombar o robô ao redor de uma das fronteiras do polígono de suporte [9]. Dessa forma, podemos escrever matematicamente essa relação como:



(a) Uma pata sem contato com o solo



(b) Todas as pernas no sobre o solo com patas próximas



(c) Todas as pernas no sobre o solo com patas afastadas

Figura 2.1: Polígonos de Suporte de um robô quadrúpede

$$S_{ESM} = \min_{i}^{n_s}(mgh_i), \tag{2.1}$$

em que m é a massa do robô, g é a gravidade, i denota qual segmento do polígono de suporte é considerado, n_s é o número de pernas de suporte e h_i é a variação da altura do COG durante o tombamento.

É interessante notar que, como este método ainda trata-se de estabilidade estática, apenas a gravidade é considerada e, por isso, para este caso, a energia potencial é a única componente energética que contribuiria para o tombamento do robô.

Em 1998, Hirose *et al.* definiram a Margem de Estabilidade por Energia Normalizada (S_{NESM}) como a S_{ESM} normalizada com o peso do robô, de forma que a unidade mantenha-se igual à proposta pelos métodos geométricos (unidade de comprimento). Por conseguinte, a margem normalizada pode escrita como:

$$S_{NESM} = \frac{S_{ESM}}{mg} = \min_{i}^{n_s}(h_i). \tag{2.2}$$

A S_{NESM} tem se mostrado como a mais eficiente medida de margem de estabilidade estática, mas ainda não comporta os efeitos dinâmicos que surgem em aplicações de maiores velocidades ou com distúrbios externos [9].

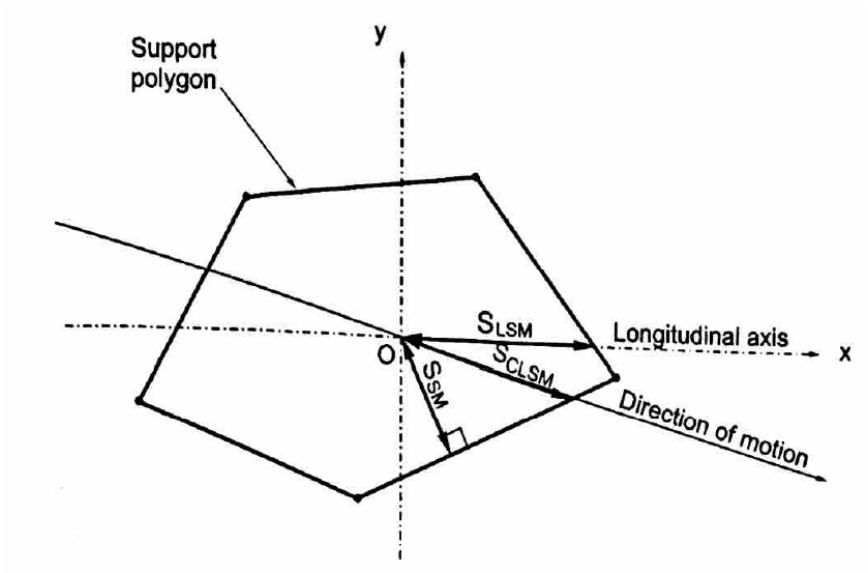


Figura 2.2: Margens de Estabilidade Longitudinal e de Caranguejo

2.3 Estabilidade Dinâmica

O estudo da estabilidade dinâmica tem como propósito a modelagem de situações mais concretas existentes em aplicações reais. Desta maneira, a inclusão de efeitos inerciais, forças e momentos externos ao estudo de estabilidade permite ao sistema de locomoção trabalhar com velocidades maiores, em terrenos mais irregulares e com condições mais adversas.

2.3.1 Método do Centro de Pressão

Analogamente ao método geométrico da projeção do centro de gravidade definido para a estabilidade estática, Orin definiu em 1976 o método do Centro de Pressão (COP) [9]. Neste método, define-se como dinamicamente estável o robô cuja projeção do centro de gravidade (COG) ao longo da direção da força resultante agindo sobre o COG se encontra dentro do polígono de suporte. Da mesma maneira que no caso estático, a margem de estabilidade será a menor distância desta projeção e qualquer uma das bordas do polígono.

Esta definição foi modificada posteriormente assim como a sua nomenclatura. Kang *et al.*, em 1997, renomeou o COP como Centro de Massa Efetiva (EMC) e o definiu como o ponto no plano em que o momento resultante é nulo. Esta definição é similar à definição de Ponto de Momento Zero (ZMP) utilizado em robôs bípedes [9].

2.3.2 Margem de Estabilidade Dinâmica

Quando o robô porventura sofre a ação de um distúrbio suficientemente forte para fazê-lo iniciar um processo de tombamento, algumas patas podem perder o contato com o chão e provocar uma rotação ao redor de um eixo, como pode ser observado na Figura 2.3. Quando isto acontece, o

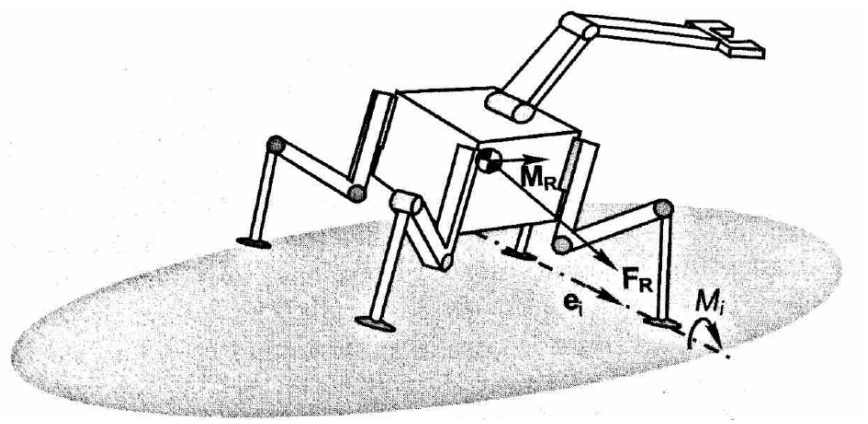


Figura 2.3: Robô prestes a tombar

sistema de controle deve gerar uma força resultante F_R e um momento resultante M_R para poder contrabalancear os distúrbios externos que provocaram o desequilíbrio inicial. Dessa forma, o momento total, M_i, gerado através de F_R e de M_R, deve ser suficiente para garantir essa compensação, caso contrário o sistema será considero instável.

Desse modo, Lin e Song, em 1993, definiram como Margem de Estabilidade Dinâmica (S_{DSM}) como o menor momento M_i dentre os eixos de rotações possíveis, normalizado com o peso do robô, onde i descreve cada eixo [9]. Portanto, podemos descrever essa relação como

$$S_{DSM} = \min\left(\frac{M_i}{mg}\right) = \min\left(\frac{e_i \cdot (F_R \times P_i + M_R)}{mg}\right), \tag{2.3}$$

na qual P_i é o vetor de posição do COG até a i-ésima pata e e_i é o vetor unitário que circunda o polígono de suporte na direção horária. Estes parâmetros podem ser observados com maior clareza na Figura 2.3.

2.3.3 Margem Normalizada de Estabilidade Dinâmica por Energia

Assim como foi feito no caso estático, podemos analisar a estabilidade dinâmica por uma perspectiva energética. Porém, para este modelo, considera-se, além da força gravitacional, forças e momentos aplicados externamente, além da energia cinética representando a inércia do sistema. Do mesmo modo, a Margem Normalizada de Estabilidade Dinâmica por Energia (S_{NDESM}) é definida como a menor energia necessária para se tombar o robô ao redor do polígono de suporte:

$$S_{NDESM} = \frac{\min(E_i)}{mg}, \tag{2.4}$$

em que E_i é a energia necessária para se tombar o robô através da i-ésima borda do polígono de suporte. Ela pode ser calculada através da seguinte equação:

$$E_i = mg|R|(\cos(\phi) - \cos(\varphi))\cos(\Psi) + (F_{RI} \cdot t)|R|\theta + (M_R \cdot e_i)\theta - \frac{1}{2} I_i \omega_i^2. \quad (2.5)$$

Nessa relação, R é o vetor ortogonal à i-ésima borda do polígono de suporte que aponta para o COG, F_{RI} é a componente não gravitacional das forças resultantes F_R, I_i é o momento de inércia do robô ao redor da i-ésima borda, ω_i é a velocidade angular do COG , Ψ é o ângulo de inclinação da i-ésima borda do polígono de suporte, φ é o ângulo de rotação necessário para posicionar o COG no plano vertical, ϕ é o ângulo entre o plano vertical e o plano crítico, θ é a soma de φ e ϕ, e t é o vetor unitário tangencial à trajetória do COG. A Figura 2.4 mostra o esquemático geométrico para a determinação de tais parâmetros.

As seções a seguir apresentam alguns outros métodos dinâmicos para o controle de estabilidade. Eles são considerados dinâmicos por considerarem a aplicação de forças externas e momentos resultantes, porém não utilizam, necessariamente, os métodos de análise de margem de estabilidade como o descrito nesta seção.

2.4 Algoritmo de Posicionamento de Pata

Em 1986, Raibert *et al.* propuseram uma abordagem diferente para o controle de estabilidade de robôs movidos a pernas [10, 11]. O Algoritmo de Posicionamento de Pata (*Foot Placement Algorithm - FPA*) utiliza os algoritmos de robôs com uma perna e os estende para múltiplas pernas através do conceito de perna virtual. Um exemplo de robô com uma perna pode ser visto na Figura 2.5. Dessa forma, o controle de equilíbrio pode ser dividido em três partes: altura de pulo, postura do corpo e velocidade direta de corrida. Até os dias atuais, esse conceito é utilizado em robôs como o *BigDog* [1].

2.4.1 Algoritmos de Uma Perna

Em robôs de uma perna como o da Figura 2.5, o sistema é dividido em corpo e perna, com um quadril tipo dobradiça separando-os. Um atuador gera um torque no quadril e outro proporciona movimento axial na perna. Uma mola é adicionada em série com o atuador axial de modo que o sistema de controle possa excitar o sistema massa-mola gerando o movimento desejado. Dessa forma, podemos dividir o controle de equilíbrio em três:

Altura de Pulo: O sistema de controle entrega um impulso vertical, através do atuador axial, regulando a altura que a máquina chega. Dessa forma, pode-se regular a amplitude do movimento e, consequentemente, manter o ciclo de oscilação governado pelo sistema massa-mola. Assim, parte da energia para cada salto é recuperada pela mola.

Postura do Corpo: O sistema de controle gera um torque sobre o quadril (entre o corpo e a perna) durante a fase de contato com o solo para manter o corpo numa posição ereta. O torque

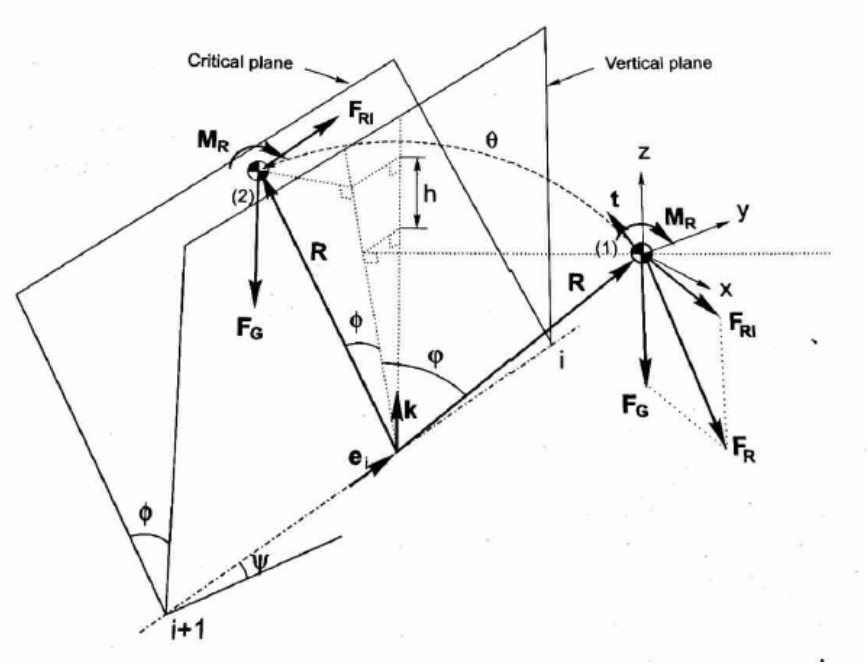


Figura 2.4: Esquemático dos parâmetros geométricos durante o tombo

aplicado será dado por

$$\tau = -k_p(\phi - \phi_d) - k_v(\dot{\phi}), \tag{2.6}$$

em que τ é o torque aplicado na dobradiça do quadril, k_p e k_v são ganhos, ϕ é a inclinação do corpo, ϕ_d é a inclinação desejada do corpo e $\dot{\phi}$ é a taxa de variação da inclinação do corpo.

Velocidade Direta de Corrida: Durante a fase de voo, o sistema de controle manipula a posição que a pata irá atingir o solo possibilitando, dessa forma, o controle da velocidade de corrida. Definindo *CG Print* como o conjunto de pontos no chão por quais o centro de massa do corpo irá passar durante o contato com o solo podemos identificar que, se a pata pousar no centro do *CG Print* (o chamado ponto neutro), não haverá aceleração. Do contrário, um posicionamento da pata após o ponto neutro irá causar uma desaceleração e, com a pata posicionada antes do ponto neutro, o robô sofrerá uma aceleração. Podemos observar uma imagem do *CG Print* na Figura 2.6. Esta relação pode ser descrita pela seguinte equação:

$$x_f = \frac{\dot{x}T_S}{2} + k_{\dot{x}}(\dot{x} - \dot{x_d}), \tag{2.7}$$

de modo que x_f é o posicionamento do pé na direção do movimento, com relação à projeção do centro de gravidade, $\dot{x}$ é a velocidade atual do movimento, $\dot{x_d}$ é a velocidade desejada, T_S é a duração do período de suporte (enquanto a pata ainda está em contato com o solo) e $k_{\dot{x}}$ é um ganho de velocidade.

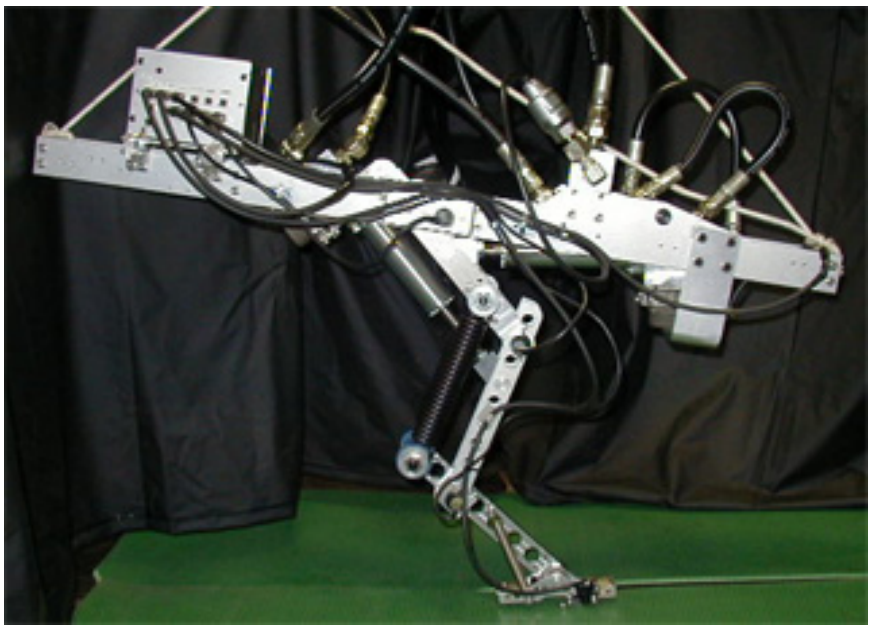


Figura 2.5: Robô de uma perna

Obtido x_f, uma transformação cinemática calcula o ângulo que a junta deve fornecer para atingir tal posição.

2.4.2 Marcha de uma pata

Considerando um robô com mais de uma pata, podemos definir como Marcha de uma Pata (*One foot gait*) como uma marcha em que apenas uma perna oferece suporte por vez e que as fases de suporte e de voo ocorrem de forma alternada. Para o caso de quadrúpedes, por exemplo, é necessário que o sistema de controle alterne as patas a serem utilizadas de forma que a perna atualmente em contato com o solo proporcione o impulso vertical para manter o movimento enquanto sua respectiva dobradiça providencia o torque para correção da postura e a próxima pata

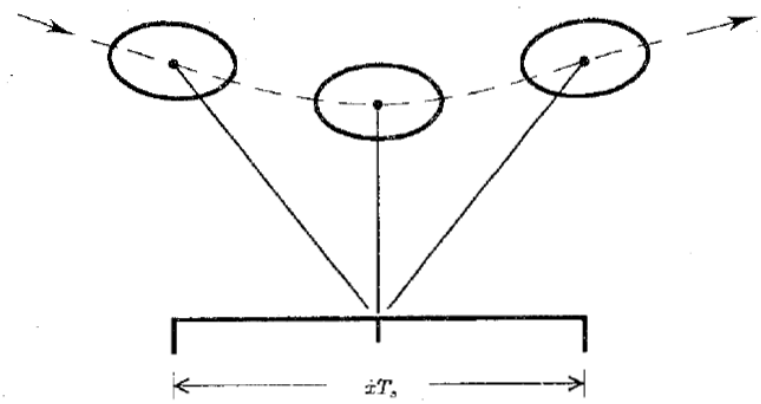


Figura 2.6: *CG Print* de um robô de uma perna

se prepara para aterrissar no local adequado à velocidade desejada. Dessa forma, o sistema estaria provendo as três parte do controle.

O problema que surge com esta configuração é que as pernas devem estar muito próximas do ponto neutro, o que, além de dificultar o design do robô, pode causar interferências de uma perna com outra. A solução adotada por Raibert *et al.*, foi de utilizar o conceito de pernas virtuais.

2.4.3 Pernas Virtuais

O conceito de pernas virtuais trata-se da coordenação de pares de pernas funcionando ao mesmo tempo de forma que pode-se representar tal par como uma perna equivalente. O par de pernas e a perna virtual exercem a mesma força e momento sobre o corpo, portanto, ambas as configurações geram os mesmos efeitos sobre este. A Figura 2.7 demonstra a representação de um par de pernas em uma perna virtual.

Nos quadrúpedes, a coordenação de pares de pernas podem ser feitas em três diferentes maneiras: pernas diagonais, gerando o movimento conhecido como trote, pernas laterais, gerando o passo e pernas traseiras e dianteiras, gerando o salto. Podemos ver na Figura 2.8 as pernas virtuais equivalentes para cada tipo de marcha.

Por conseguinte, utilizando o conceito de pernas virtuais, podemos estender a equação 2.7 para o movimento de quadrupedes que, por sua vez, possuem duas pernas virtuais e podem se mover em duas dimensões, obtendo, dessa forma:

$$x_{f,d} = \frac{\dot{x}T_S}{2} + k_{\dot{x}}(\dot{x} - \dot{x_d}), \tag{2.8}$$

$$y_{f,d} = \frac{\dot{y}T_S}{2} + k_{\dot{y}}(\dot{y} - \dot{y_d}), \tag{2.9}$$

em que $x_{f,d}$ e $y_{f,d}$ são os posicionamentos das patas virtuais na direção de cada dimensão, com relação à projeção do centro de gravidade, $\dot{x}$ e $\dot{y}$ são as velocidades atuais do movimento, $\dot{x_d}$ e $\dot{y_d}$ são as velocidades desejadas, T_S é a duração do período de suporte (enquanto a pata ainda está em contato com o solo) e $k_{\dot{x}}$ e $k_{\dot{y}}$ são ganhos de velocidade.

Para controlar a postura do corpo, o sistema atua nos ângulos de rolagem e arfagem aplicando torques sobre os quadris virtuais durante o período de suporte usando servos lineares, através das

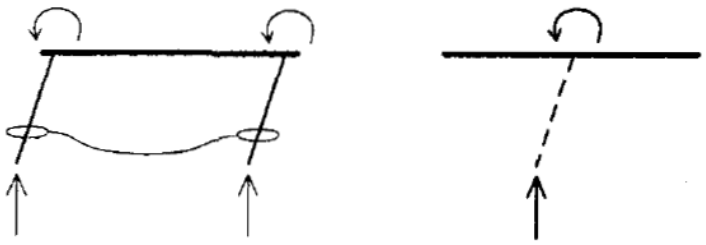


Figura 2.7: Perna Virtual

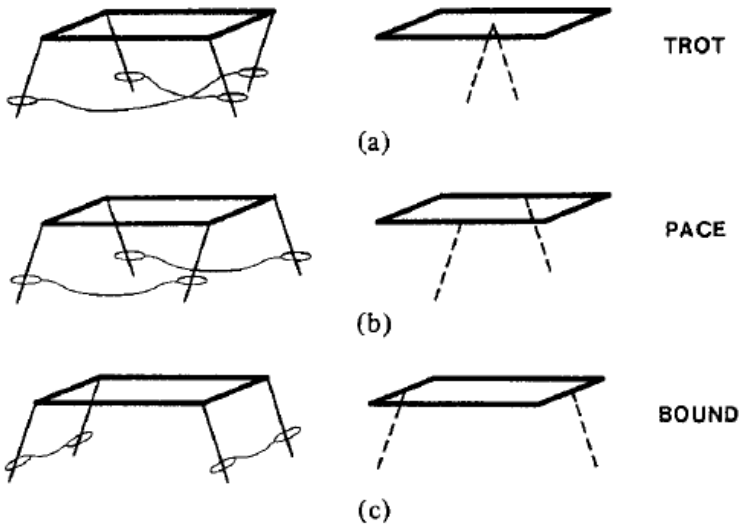


Figura 2.8: Pernas virtuais para cada marcha

seguintes equações:

$$u_x = -k_{p,x}(\phi_P - \phi_{P,d}) - k_{v,x}(\dot{\phi_P}) - k_{f,x}(f_x), \tag{2.10}$$

$$u_y = -k_{p,y}(\phi_R - \phi_{R,d}) - k_{v,y}(\dot{\phi_R}) - k_{f,y}(f_y), \tag{2.11}$$

de modo que u_x e u_y são os sinais de saída dos atuadores, ϕ_P e ϕ_R são os ângulos de arfagem e rolagem, respectivamente, $\phi_{P,d}$ e $\phi_{R,d}$ são os ângulos desejados de arfagem e rolagem, respectivamente, f_x e f_y são as forças entregues pelos atuadores do quadril e k_p, k_v e k_f são ganhos.

2.4.4 Exemplo de Aplicação

Este modelo já foi utilizado na modelagem de robôs quadrúpedes de diversas maneiras [11] e um exemplo pode ser encontrado nos trabalhos de Li *et al.* [2]. Nestes, os modelos de Raibert são utilizados em conjunto com o modelo do Pêndulo Invertido com Massa-Mola (*Spring Loaded Inverted Pendulum*) para poder obter mais detalhes dinâmicos, como por exemplo, para robôs com 6 graus de liberdade. Ao fazer isso, pode-se aumentar a robustez do trote e suavizar e estabilizar a corrida do quadrúpede. A Figura 2.9 mostra diversas vistas do modelo deste robô.

Assim como em [10] o modelo proposto em [2] utiliza o conceito de pernas virtuais para implementar uma marcha de trote (pernas diagonais se movimentando em pares) atuando, virtualmente, como um bípede. Neste caso, o modelo dinâmico utilizado, baseado no modelo SLIP, pode ser descrito através das equações:

$$M\ddot{r} + K(r - r_0) - Mr\dot{\theta}^2 = -Mg\cos\theta, \tag{2.12}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(Mr^2\dot{\theta}\right) = Mgr\sin\theta, \tag{2.13}$$

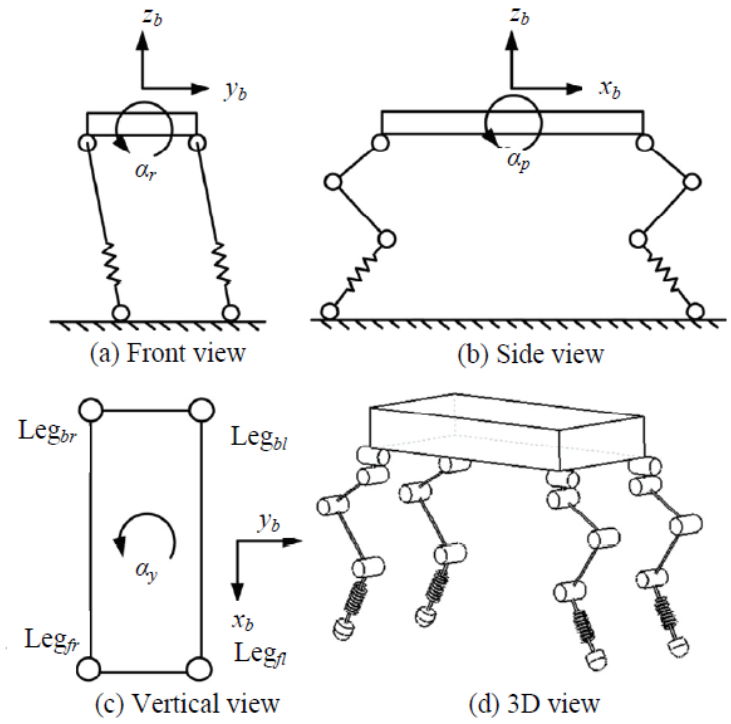


Figura 2.9: Vistas do quadrúpede com 6 graus de liberdade

em que M é a massa do corpo, K é o coeficiente elástico da mola, r é o comprimento da perna, r_0 é o comprimento inicial da perna, θ é o ângulo em relação ao plano vertical, g é a gravidade e $\dot{\theta}$ é a velocidade angular da perna. O modelo utilizando para determinação de tais parâmetros pode ser visto na Figura 2.10.

Por sua vez, o algoritmo de controle utilizado para regular a corrida e o salto é uma variação do modelo de Raibert e é dado da seguinte forma:

$$\theta_s = \arcsin(\frac{v_x T_s}{2l_0}) + k_p(v_x - v_{xt}) + k_i \sum_{step}(v_x - v_{xt}), \tag{2.14}$$

$$l_u = l_{u0} + C(v_z - v_{zt}), \tag{2.15}$$

em que θ_s é o ângulo de toque (complemento de θ, da Figura 2.10), l_u é o comprimento do atuador, T_s é o período da fase de suporte, l_0 é o comprimento inicial da perna, v_x e v_z são as velocidades na direção do movimento e na direção vertical, respectivamente, v_{xt} e v_{zt} são as velocidades desejadas, l_{u0} é o comprimento inicial do atuador e k_p, k_i e C são parâmetros constantes.

Dessa forma, o algoritmo de controle utiliza a equação 2.14 para determinar os ângulos que devem ser aplicados em cada junta para proporcionar o movimento lateral e direto desejado. A equação 2.15, por sus vez, determina a ação dos atuadores axiais.

Já o controle dos ângulos de rolagem e arfagem utiliza um modelo simplificado como o da Figura 2.11. Este modelo pode ser descrito por

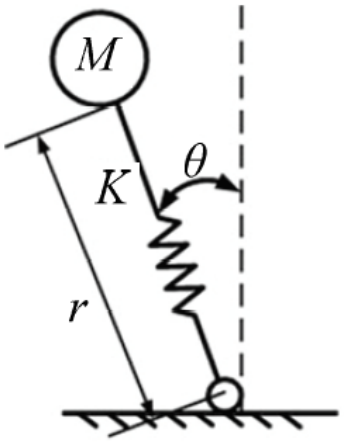


Figura 2.10: Modelo SLIP

$$J\ddot{\beta} = \tau_b + \tau_f + (F_2 - F_1) \cdot l_1/2, \tag{2.16}$$

em que J é o momento de inércia do torso do robô, β é o ângulo de postura, $\ddot{\beta}$ é a aceleração angular de β, F_1 e F_2 são as forças de contato das patas, l_1 é a distância entre as duas juntas e τ_b e τ_f são os torque de saída das juntas.

Pode-se considerar o corpo como um sistema amortecido com mola de forma que o algoritmo de controle pode ser descrito como

$$J\ddot{\beta} = -k_{p\beta}(\beta - \beta_d) - k_{d\beta}\dot{\beta}, \tag{2.17}$$

de modo que $k_{p\beta}$ é a rigidez da mola equivalente, $k_{d\beta}$ é o amortecimento e β_d é o ângulo desejado. Este configura um sistema de segunda ordem e é dado pela seguinte função de transferência:

$$\frac{\beta(s)}{\beta_d(s)} = \frac{\omega_n}{s^2 + 2\xi\omega_n s + \omega_n^2} = \frac{2k_{p\beta}/J}{s^2 + 2k_{d\beta}s/J + 2k_{p\beta}/J}, \tag{2.18}$$

sendo que ξ é a razão de amortecimento e ω_n é a frequência natural do sistema de segunda ordem.

2.5 Controle de Balanço Dinâmico

Em 2015, Meng *et al.* realizaram um estudo para analisar o balanceamento de um robô quadrúpede quando este se encontra apoiado em duas pernas diagonais [11]. Analisando robôs previamente construídos os autores deste estudo compreenderam que, para que estes veículos permanecessem

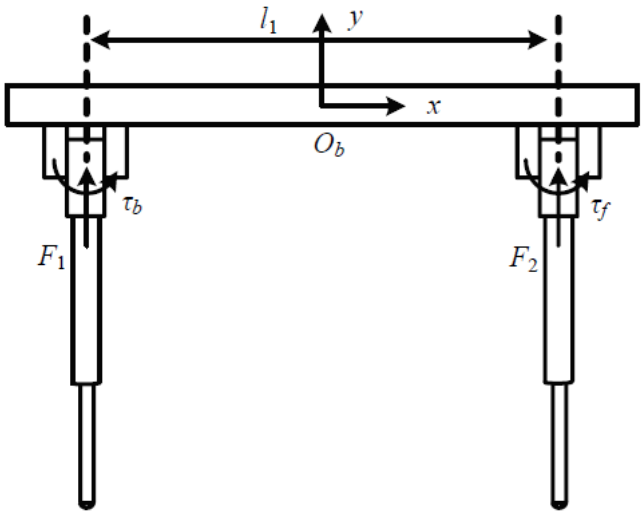


Figura 2.11: Modelo para controle de arfagem e rolagem

estáveis, eles tinham que manter o movimento (como no desenvolvimento de Raibert) ou estaticamente sobre as quatro patas. Com isso em mente, foi proposto um modelo de controle dinâmico para este tipo de equilíbrio.

Em um quadrúpede apoiado em suas pernas diagonais podemos assumir que as pernas sem contato com o solo fazem parte do corpo, facilitando a compreensão e desenvolvimento das equações. Dessa forma, o modelo utilizado tem o formato presente da Figura 2.12

Se θ_{corpo} é o ângulo de arfagem, θ_{FS} e θ_{HS} são, respectivamente, os ângulos da perna dianteira e traseira e todos podem ser medidos através de sensores, podemos definir uma relação da seguinte forma:

$$\theta_F = \theta_{FS} - \theta_{corpo}, \tag{2.19}$$

$$\theta_H = \theta_{HS} - \theta_{corpo}, \tag{2.20}$$

em que θ_F é o ângulo entre a perna dianteira e a direção da gravidade e θ_H é o ângulo entre a perna traseira e a direção da gravidade. Esta relação pode ser exemplificada pela Figura 2.13.

O modelo dinâmico utilizado é baseado na equação de Lagrange, uma alternativa à Lei de Newton particularmente conveniente para sistemas com vários graus de liberdade ou com sistema de coordenadas muito complexo. Ela é dada por

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q} = S\tau, \tag{2.21}$$

sendo que, para este caso, $q = \begin{bmatrix}\theta_F & \theta_H & \theta_{corpo}\end{bmatrix}$, $S = \begin{bmatrix}1 & 0 & -1\\ 0 & 1 & -1\end{bmatrix}^T$ e $\tau = \begin{bmatrix}\tau_F\\ \tau_H\end{bmatrix}$ sendo que τ_F e τ_H são os torques exercidos pelo atuador dianteiro e traseiro, respectivamente.

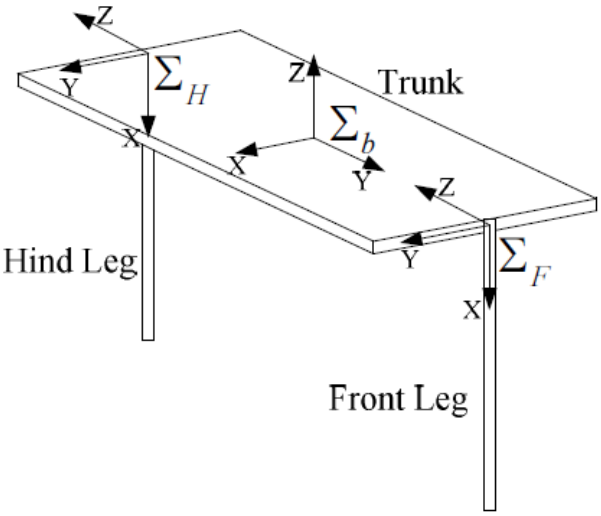


Figura 2.12: Modelo de quadrúpede apoiado em pernas diagonais

Dessa forma, o controle de balanço para este modelo será realizado com base na seguinte definição de variáveis de estado: $X = \begin{bmatrix} \theta_F & \dot{\theta}_F & \theta_H & \dot{\theta}_H & \theta_{corpo} & \dot{\theta}_{corpo} \end{bmatrix}^T$. Assim, a equação dinâmica linearizada pode ser dada por $\dot{X} = AX + B\tau$ e o estado de realimentação, baseado em Regulador Quadrático Linear (RQL) pode ser obtido por $\tau = -KX$. Dessa forma, a estrutura de controle pode ser arranjada conforme a Figura 2.14.

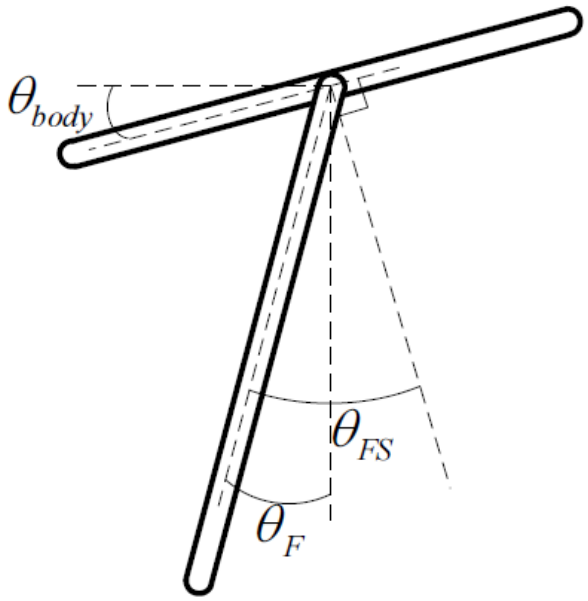


Figura 2.13: Modelo das relações de ângulo entre perna e corpo

2.6 Razão de Aterrissagem em Acordo

Uma forma alternativa de mensurar a estabilidade dinâmica em robôs quadrúpedes foi proposta em 2009 por Won *et al.*, denominada Razão de Aterrissagem em Acordo (*Landing Accordance Ration - LAR*) [12].

Em uma marcha quadrúpede do tipo trote os membros diagonalmente dispostos se movem em pares, idealmente em sincronia, como pode ser exemplificado graficamente pela Figura 2.15. Onde LF e RF referem-se, respectivamente, às pernas esquerda e direita da frente enquanto que LH e RH referem-se, respectivamente, às pernas esquerda e direita da parte traseira do robô. O gráfico mostra quais pernas estão em contato com o solo em função do tempo.

No entanto, durante a ação real do robô, pode haver um desacordo entre os pares de pernas de modo que as patas não se encontrem com o solo ao mesmo tempo como pode ser exemplificado pela Figura 2.16. Onde t_{td} é o tempo que um par de pernas fica dessincronizado para tocar o solo e t_{lo} é o tempo de desacordo para um par de pernas sair do chão.

Dessa forma, sabendo que em marchas tipo trote estáveis a aterrissagem deve ser bem sincronizada definiu-se a Razão de Aterrissagem em Acordo (LAR) como

$$\lambda = \frac{t - t_{td}}{t}, \tag{2.22}$$

em que λ representa o LAR e t é o período de suporte, ou seja, o tempo que aquele par de pernas fica em contato com o solo.

2.6.1 Controle e Planejamento de Marcha

O controle é feito baseado nas forças de reação em cada pata. Cada força, por sua vez, é dividida em uma componente para a geração da trajetória de caminhada (f_{vsd}) e outra para o equilíbrio dinâmico (f_{bal}). Dessa forma o torque que deve ser exercido é dado por

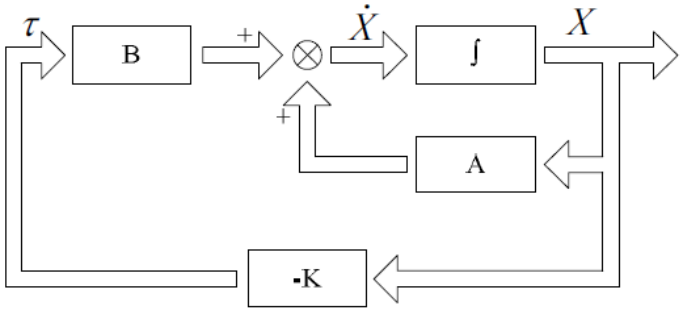


Figura 2.14: Estrutura do sistema de controle

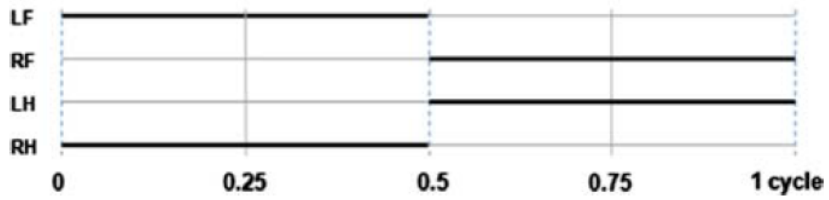


Figura 2.15: Sincronia em marcha quadrúpede

$$\tau = J^T(f_{vsd} + f_{bal}\hat{k}), \tag{2.23}$$

em que J^T é a transposta da matriz Jacobiana dos pontos finais de cada pata em relação aos ângulos das juntas. O cálculo dessa matriz será melhor abordada nas seções seguintes. Já $\hat{k}$ refere-se ao vetor unitário na direção vertical.

O sistema de controle, dessa forma, pode ser implementado conforme o diagrama de blocos da Figura 2.17.

Adotando o sistema de coordenada e ângulos conforme o mostrado na Figura 2.18(a) podemos estabelecer as relações necessárias para os cálculos da dinâmica a ser controlada. A ideia do controle é redistribuir os torques aplicados de modo que o robô possa recuperar a postura corporal quando sofre um momento de instabilidade. Conforme ilustrado pela Figura 2.18(b) as forças de reação exercídas sobre as patas do quadrúpede geram um momento diferente de zero levando o robô à instabilidade. Este momento irá causar erros nos ângulos de rolagem e arfagem assim como em suas velocidades. Dessa forma, o sistema de controle pode ser modelado como um sistema de mola amortecido da seguinte forma:

$$\tau_\phi = k_{p,x}\Delta\phi + k_{d,x}\Delta\dot{\phi}, \tag{2.24}$$

$$\tau_\theta = -(k_{p,y}\Delta\theta + k_{d,y}\Delta\dot{\theta}), \tag{2.25}$$

de modo que τ_ϕ e τ_θ são os torques que devem ser aplicados para corrigir, respectivamente, os ângulos de rolagem e arfagem, $\Delta\phi$ e $\Delta\theta$ são os erros associados ao movimento de rolagem e arfagem, respectivamente, ou seja, o quanto eles diferem da situação de estabilidade. Por fim, $k_{p,x}$, $k_{p,y}$, $k_{d,x}$ e $k_{d,y}$ são ganhos.

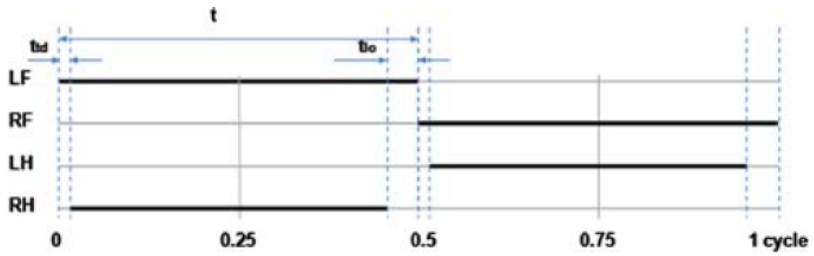


Figura 2.16: Representação da Dessincronização das Pernas

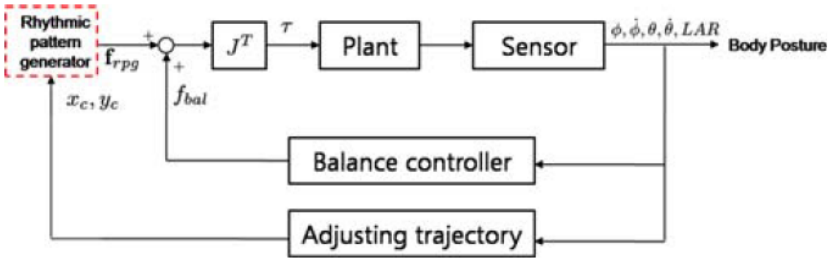


Figura 2.17: Diagrama de Blocos do Sistema de Controle

Portanto a componente de força associada ao controle de equilíbrio do quadrúpede será dado pela soma das forças que gerarão o torque acima descrito. Dessa forma

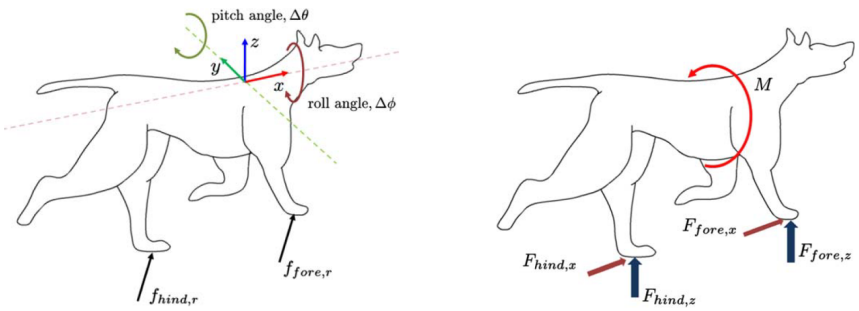
$$f_{bal} = f_{\phi,l} + f_{\theta,l} = \frac{k_{p,x}\Delta\phi + k_{d,x}\Delta\dot{\phi}}{r_{y,l}} - \frac{k_{p,y}\Delta\theta + k_{d,y}\Delta\dot{\theta}}{r_{x,l}}, \tag{2.26}$$

em que $f_{\phi,l}$ e $f_{\theta,l}$ são as forças de reação para compensar os ângulos de rolagem e arfagem, respectivamente, $r_{x,l}$ e $r_{y,l}$ são as componentes x e y, respectivamente, do vetor que vai do centro de massa do robô até a pata de índice l.

Por fim, um algoritmo é criado utilizando a medição do LAR para determinar a trajetória de caminhada que compensa a instabilidade do robô. Tal algoritmo tem a seguinte forma: caso o valor do LAR diminua procura-se a pata que está em desacordo, o controle de postura é acionado para corrigir a instabilidade, caso o LAR não tenha aumentado, retorna-se para a busca pela pata em desacordo, caso o LAR tenha aumentado o algoritmo termina. Este processo pode ser descrito pelo diagrama mostrado na Figura 2.19.

2.7 Modelo Cinemático e Cinemático Inverso

O modelo cinemático e cinemático inverso são formas de relacionar as velocidades das juntas de cada perna com a posição da pata, o chamado efetuador final (*end effector*) [13]. Nestes modelos



(a) Sistemas de coordenadas e ângulos adotados

(b) Forças e momentos atuantes no quadrúpede

Figura 2.18: Parâmetros de um quadrúpede em instabilidade

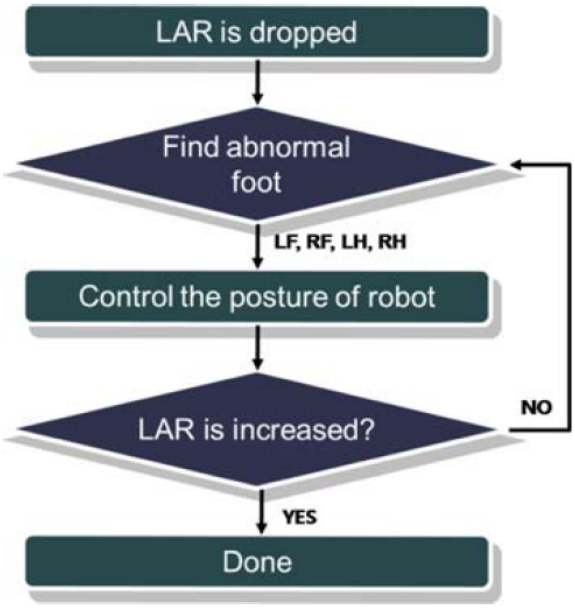


Figura 2.19: Algoritmo para controle de estabilidade utilizando o LAR

não são considerados forças nem momentos e são bastante úteis para o controle de quadrúpedes, uma vez que, podendo relacionar juntas com efetuadores finais, podemos controlar de forma precisa a posição dos atuadores rotacionais para gerar a posição de pata desejada. No entanto, para compreender estes modelos, deve-se, primeiramente analisar o modelo geométrico.

2.7.1 Modelo Geométrico

Considere dois sistemas de coordenadas, $X_1 \times Y_1 \times Z_1$ e $X_0 \times Y_0 \times Z_0$ ambos com a origem no mesmo ponto mas rotacionados entre si. Se p_1 é a representação de um determinado ponto em $X_1 \times Y_1 \times Z_1$, podemos representá-lo em $X_0 \times Y_0 \times Z_0$ atráves da seguinte matrix de rotação:

$$R_1^0 = \begin{bmatrix} x_1 \cdot x_0 & y_1 \cdot x_0 & z_1 \cdot x_0 \\ x_1 \cdot y_0 & y_1 \cdot y_0 & z_1 \cdot y_0 \\ x_1 \cdot z_0 & y_1 \cdot z_0 & z_1 \cdot z_0 \end{bmatrix}. \tag{2.27}$$

Nesta relação, x_0, y_0 e z_0 são vetores unitário referentes ao sistema de coordenadas $X_0 \times Y_0 \times Z_0$ e x_1, y_1 e z_1 são vetores unitários referentes ao sistema de coordenadas $X_1 \times Y_1 \times Z_1$. Dessa forma, a representação rotacionada, p_0, será dada por

$$p_0 = R_1^0 \cdot p_1. \tag{2.28}$$

Agora, considere que os dois sistemas de coordenadas referidos acima se encontram com a mesma orientação, ou seja $R_1^0 = I$, onde I é a matriz identidade, mas transladados entre si. Podemos representar o ponto p_1 no sistema de coordenadas $X_0 \times Y_0 \times Z_0$ como

$$p_0 = p_1 + d_1^0, \tag{2.29}$$

em que d_1^0 é o vetor que vai da origem O_0 até a origem O_1 de cada sistema de coordenadas.

2.7.1.1 Tranformação Homogênia

Um ponto $p = \begin{bmatrix} p_x & p_y & p_z \end{bmatrix}^T$ pode ser descrito em coordenadas homogênas da seguinte forma:

$$\tilde{p} = \begin{bmatrix} \tilde{p_x} \\ \tilde{p_y} \\ \tilde{p_z} \\ \omega \end{bmatrix}, \tag{2.30}$$

sendo que $p_x = \frac{\tilde{p}_x}{\omega}$, $p_y = \frac{\tilde{p}_y}{\omega}$ e $p_z = \frac{\tilde{p}_z}{\omega}$. A variável ω é um fator escalar que, para este caso, será considerado como unitário.

Isto posto, podemos unificar as relações de rotação e translação em uma única transformação homogênea, representada pela seguinte matriz:

$$H = \begin{bmatrix} R & d \\ 0_{1\times 3} & 1 \end{bmatrix}, \tag{2.31}$$

em que R é a matriz de rotação, d é o vetor de translação e $0_{1\times 3}$ é o vetor linha $\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$. Portanto, uma transformação homogênea, composta por rotação e translação, de um sistema de coordenadas S_1 para um sistema de coordenadas S_0 é dada por

$$\tilde{p}_0 = H_1^0 \tilde{p}_1. \tag{2.32}$$

Podemos utilizar, ainda, uma notação simplificada H_i ($i \in \mathbb{Z}$) para representar uma transformação homogênea do sistema de coordenadas S_i para o sistema de coordenadas S_{i-1}.

2.7.1.2 Modelo Geométrico em Robôs Manipuladores

O modelo geométrico consiste na aquisição das posições dos efetuadores finais (neste caso, das patas do robô) em função das variáveis das juntas.

Considere que a matriz $H_m^0(q, \lambda)$ representa uma transformação homogênea que possibilita a aquisição das posições e orientações de um efetuador com m graus de liberdade em função do vetor (q), das varáveis de juntas e do vetor λ composto pelas dimensões físicas do robô. Dessa forma,

desconsiderando a orientação do efetuador (assume-se que esta é irrelevante para este estudo), podemos obter sua posição ξ, em coordenadas homogêneas, através da seguinte equação:

$$\xi = g(q, \lambda) = H_m^0 \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2.33}$$

sendo que g representa uma função do vetor q e dos parâmetros geométricos incluídos em λ.

2.7.2 Modelo Cinemático

O modelo cinemático relaciona as velocidades dos efetuadores como uma função das variáveis das juntas. Se derivarmos a equação (2.33) com respeito ao tempo, obteremos, pela regra da cadeia, que

$$\dot{\xi} = \frac{\partial g}{\partial q} \frac{\mathrm{d}q}{\mathrm{d}t}. \tag{2.34}$$

Se definirmos $\frac{\partial g}{\partial q}$ como a matriz Jacobiana J, obteremos o modelo cinemático:

$$\dot{\xi} = J\dot{q}, \tag{2.35}$$

de forma que a matriz Jacobiana será dada por:

$$J = \frac{\partial g(q, \lambda)}{\partial q} = \begin{bmatrix} \frac{\partial x}{\partial q_1} & \frac{\partial x}{\partial q_2} & \cdots & \frac{\partial x}{\partial q_m} \\ \frac{\partial y}{\partial q_1} & \frac{\partial y}{\partial q_2} & \cdots & \frac{\partial y}{\partial q_m} \\ \frac{\partial z}{\partial q_1} & \frac{\partial z}{\partial q_2} & \cdots & \frac{\partial z}{\partial q_m} \end{bmatrix}. \tag{2.36}$$

2.7.3 Modelo Cinemático Inverso

O modelo cinemático inverso busca descrever a velocidade das juntas como uma função da velocidade dos efetuadores. Para isso, temos que modificar a equação 2.35 de modo a isolar $\dot{q}$. Como J não é, necessariamente, uma matriz quadrada, a aplicação de sua inversa não será possível. Para poder fazer isto utilizaremos a matriz pseudo-inversa de J da seguinte forma:

$$\begin{aligned} J\dot{q} &= \dot{\xi}, \\ J^T J\dot{q} &= J^T\dot{\xi}, \\ \dot{q} &= (J^T J)^{-}1 J^T\dot{\xi}, \\ \dot{q} &= J^{\dagger}\dot{\xi}. \end{aligned} \tag{2.37}$$

Nestas relações, $J^{\dagger} = (J^T J)^{-}1J^T$ é a matriz pseudo-inversa de J.

A utilização da matriz pseudo-inversa, no entanto, pode apresentar graves problemas numéricos, especialmente caso haja singularidades em J. Isto decorre do fato que este sistema pode apresentar mais de uma solução, especialmente considerando casos com redundância nos graus de liberdade de um segmento (por exemplo vários atuadores em uma perna). Nestes casos, mais de uma configuração dos efetuadores acarretaria na mesma posição do efetuador final [13, 14]. Na seção seguinte será apresentado, além de alguns exemplos de utilização dos modelos cinemático e cinemático inverso, algumas possíveis soluções para este problema.

2.7.4 Aplicações

Este modelo cinemático e cinemático inverso já foi utilizado na modelagem de robôs e esta seção destina-se a exemplificar alguns casos encontrados na literatura.

Em 2016, Featherstone, desenvolveu um método de equilíbrio de robôs baseado em análises de ganho e utilizou o modelo cinemático para calcular o modelo proposto [15]. Segundo o autor, uma vez que o controle de equilíbrio tem como objetivo principal controlar o centro de massa mas tem como controle direto apenas os atuadores das juntas, pode-se dizer que, da perspectiva do controlador, a planta tem como entrada o movimento dos atuadores enquanto a saída é modelada como o movimento do centro de massa. Portanto, a performance deste controle pode ser mensurada através do ganho que carateriza essa relação de entrada e saída.

O exemplo utilizado por Featherstone tem como base o modelo da Figura 2.20, que consiste de duas juntas, uma junto ao chão e uma mais acima.

Dessa forma, c representa o vetor que sai do ponto de contato com o solo até o centro de massa, ϕ representa a direção de c em relação ao solo, q_1 e q_2 são as variáveis das juntas de baixo e de cima, respectivamente, e b é o vetor unitário perpendicular a c na direção do aumento do ângulo ϕ.

O sistema de controle deve, para estes fins, levar c_x (a componente de c na direção x) para zero ou fazer com que o ângulo ϕ seja 90^o. Dessa forma, podemos pensar na entrada do sistema como a junta q_2 e a saída podendo ser c_x ou ϕ. Portanto, o ganho associado a este sistema seria dado por

$$G_v = \frac{\Delta \dot{c}_x}{\Delta \dot{q}_2}, \tag{2.38}$$

$$G_\omega = \frac{\Delta \dot{\phi}}{\Delta \dot{q}_2}, \tag{2.39}$$

em que G_v é denominado de ganho de velocidade linear e G_ω é denominado de ganho de velocidade angular. O símbolo Δ indica uma variação tipo degrau da subsequente variável.

O autor aponta três métodos distintos para poder calcular a relação entre q_2 e c_x, possibilitando, assim, a determinação do ganho G_v. Para os propósitos deste estudo daremos foco apenas em um

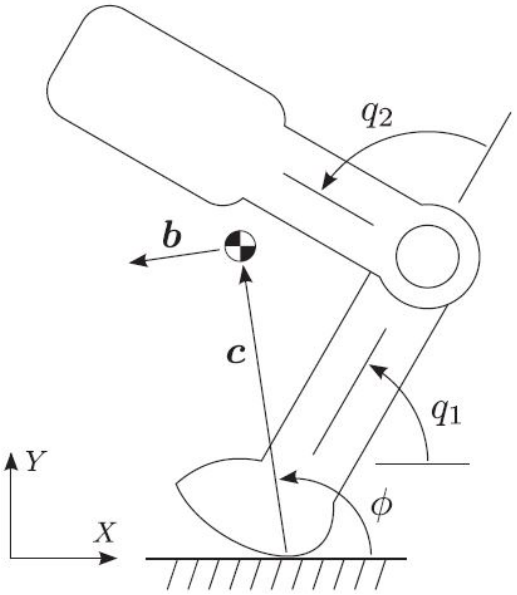


Figura 2.20: Modelo utilizado por Featherstone

destes métodos, o Jacobiano do centro de massa.

Utilizando a equação 2.35 podemos mapear diretamente a velocidade do centro de massa com a velocidade das juntas se substituirmos ξ por c e utilizarmos uma matriz J adequada para tal mapeamento. Dessa forma, temos que

$$\Delta\dot{c} = J\Delta\dot{q} = J \begin{bmatrix} \Delta\dot{q}_1 \\ \Delta\dot{q}_2 \end{bmatrix}. \tag{2.40}$$

Se considerarmos uma variação unitária em $\dot{q}_2$ e soubermos o valor de $\Delta\dot{q}_1$, podemos calcular o vetor $\dot{c}$ utilizando esta equação. O valor do ganho de velocidade linear será dado, dessa forma, pelo valor da variação da componente x de c, c_x.

Um exemplo da utilização do modelo cinemático inverso está nos trabalhos de RunBin *et al.* de 2013 [14]. Estes, especificamente, buscam modelar um sistema de controle para um quadrúpede com graus de liberdade redundantes (4 juntas em cada perna) baseado na cinemática inversa. O modelo do robô desenvolvido pode ser observado na Figura 2.21.

O controle do quadrúpede, de modo geral, pode ser dividido em duas partes: o controle de marcha e o controle de postura. O primeiro tem como objetivo principal o controle da velocidade direta do robô, de modo que suas saídas são as posições das pernas e do centro de massa. Já os objetivos do controle de postura tem como finalidade principal a resolução do modelo cinemático inverso e tem como saídas os ângulos das juntas ou seus torques. Os dois são combinados utilizando um sistema de realimentação conforme pode ser observado na Figura 2.22. Para o trabalho realizado em [14] especificamente, utilizou-se os ângulos das juntas (ao invés dos torques) como saída do sistema de controle de postura.

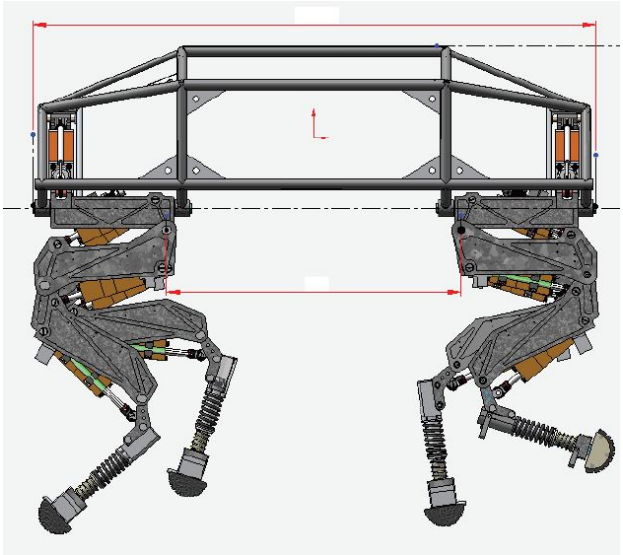


Figura 2.21: Quadrúpede com total de 16 graus de liberdade

Para este trabalho, analisaremos apenas o controle de postura, em que é aplicado diretamente o modelo cinemático inverso. Para modelá-lo é preciso, primeiramente, analisar a geometria de cada perna, exposta na Figura 2.23.

Esta configuração nos dá:

$$\begin{bmatrix} x \\ z \end{bmatrix} = \begin{bmatrix} r_1 \cos\theta_1 + r_2 \cos(\theta_1 + \theta_2) + r_3 \cos(\theta_1 + \theta_2 + \theta_3) \\ r_1 \sin\theta_1 + r_2 \sin(\theta_1 + \theta_2) + r_3 \sin(\theta_1 + \theta_2 + \theta_3) \end{bmatrix}, \tag{2.41}$$

em que x e z são as componentes das coordenadas do efetuador final, r_1, r_2 e r_3 são segmentos da perna que vão de uma junta até outra e θ_1, θ_2 e θ_3 são os ângulos das juntas. Deste modo, a matrix Jacobiana pode ser calculada como

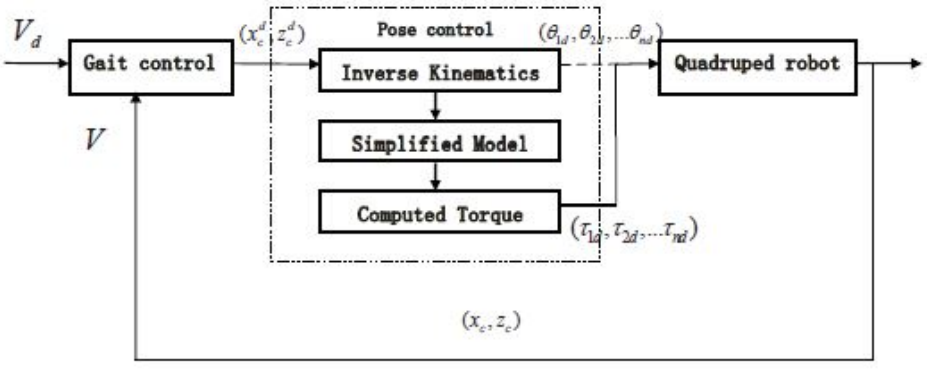


Figura 2.22: Sistema de realimentação para controle de marcha e de postura

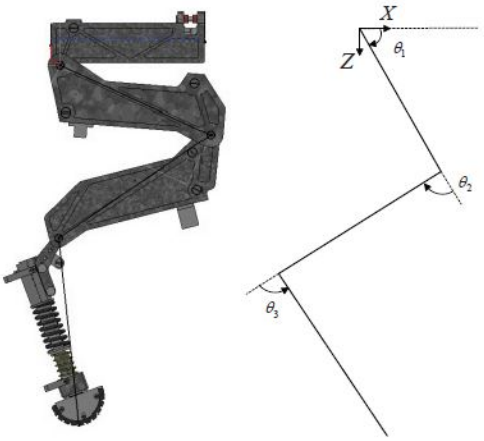


Figura 2.23: Modelo geométrico de uma perna com 4 graus de liberdade

$$J = \begin{bmatrix} J_{11} & J_{12} & J_{13} \\ J_{21} & J_{22} & J_{23} \end{bmatrix}. \tag{2.42}$$

Nesta relação, os parâmetros da primeira linha serão iguais à derivada de x com relação a θ_1, θ_2 e θ_3, nessa ordem, e da segunda linha, iguais à derivada de z com relação a θ_1, θ_2 e θ_3.

Uma vez obtido o Jacobiano, os autores propõe três métodos distintos para calcular os ângulos das juntas. O primeiro é o cálculo da matriz pseudo-inversa, conforme foi abordado anteriormente, culminando na equação 2.37. A diferença é que do lado direito da equação, os autores adicionam o termo $k(I - J^{\dagger}J * g)$, onde I é a matriz identidade, e g é uma velocidade de auto-locomoção. Este novo termo se trata da solução da equação linear homogênea enquanto que o primeiro termo é a solução normal mínima.

O segundo método é o da Menor Norma Ponderada (*Weighted least-norm - WLN*). Neste método, utilizado para minimizar a velocidade das juntas, defini-se um novo vetor de velocidade das juntas ($\dot{\theta}_W$), normalizado por uma matriz simétrica e positiva W. Então, realiza-se algumas transformações de modo a obter uma nova matriz Jacobiana para o novo sistema. Com isso, o resultado final será dado por

$$\dot{\theta}_W = W^{-1}J^T[JW^{-1}J^T]^{-1}\xi. \tag{2.43}$$

Por fim, o terceiro método é proposto e utilizado de modo a estender a matrix Jacobiana atráves de uma função g que minimiza a velocidade das juntas e, dessa forma, tornar J uma matriz quadrada que, então, será invertível.

2.8 Controle de Estabilidade Empírico

Muitos dos trabalhos anteriores utilizam sistemas de realimentação para fazer com que o sistema de controle seja mais robusto e possa convergir mais rapidamente para o valor desejado. Dessa forma, o valor atual da variável (em sua maioria os ângulos das juntas dos atuadores), é utilizado para fornecer o erro associado com o valor desejado para aquela variável. Em geral, um ganho é associado ao erro correspondente para que a saída possa convergir rapidamente.

A ideia desta seção é discutir, com maior foco, sobre este controle, baseado em dados empíricos coletados.

Em 2010, Sousa *et al.*, desenvolveram um sistema de controle de postura baseado na leitura de diversos sensores. Com uma topologia inspirada nas respostas biológicas de animais, os autores deste estudo propuseram um sistema de controle de postura independente do sistema de locomoção, havendo interação apenas quando necessário. O controle de postura seria construído com base em cada estímulo medido e uma resposta adequada seria produzida para corrigir aquela variável. O modelo de controle é integrado com um *Central Pattern Generators (CPG)* para gerar a resposta desejada.

A plataforma quadrúpede AIBO, da Sony (que pode ser vista na Figura 1.3) é utilizada para o estudo, uma vez que esta possui um acelerômetro de três eixos e um sensor de força em cada pata.

O controle de postura proposto se baseia na integração de diversos aspectos sensoriais e na produção de uma resposta apropriada para cada medição. Pode-se afirmar que cada dado sensorial é a entrada de um sistema específico e que sua saída é uma resposta de modo a corrigir a postura do robô. Ao final, todas as respostas produzidas são integradas para obter o movimento total desejado para a postura. Podemos ver, na Figura 2.24, as medições empíricas e suas respectivas respostas.

Desse modo, se $y_{i,p}$ é a resposta de correção total da junta p, da perna i, então este será a soma de todas as respostas:

$$y_{i,p} = f_{roll,i,p} + f_{pitch,i,p} + f_{COM,i,p} + f_{force,i,p} + f_{touch,i,p} + f_{disperser,i,p} + f_{reset,i,p}, \qquad (2.44)$$

Postural response	Sensory input
Roll compensation	Body roll angle
Pitch compensation	Body pitch angle
Center of Mass adjustment	Encoders and body angle
Load distribution	Joints load
Touch control	Foot touch
Leg disperser	Leg encoders

Figura 2.24: Medições Empíricas e suas Respectivas Respostas

em que f é a resposta proveniente de cada dado sensorial apresentado na Figura 2.24.

Para este estudo, as respostas mais relevantes são as dos ângulos de rolagem e arfagem e são dadas por

$$f_{roll,i,p} = k_{roll} f_i(\phi_{roll}), \tag{2.45}$$

$$f_{pitch,i,p} = k_{pitch} f_i(\phi_{pitch}), \tag{2.46}$$

de modo que k_{roll} e k_{pitch} são ganhos estáticos que definem a velocidade de convergência da resposta à situação de equilíbrio, ϕ_{roll} e ϕ_{pitch} são os ângulos medidos de rolagem e arfagem, respectivamente, e f_i é uma função linear utilizada para eliminar o ruído do sensor e pode ser positiva ou negativa, dependendo da contribuição da junta para aquela variável.

Capítulo 3

Desenvolvimento

Resumo opcional.

3.1 Introdução

Os trabalhos desenvolvidos anteriormente na plataforma quadrúpede do LARA não apenas atualizaram-na com componentes adequados para o projeto, como motores mais robustos, *Raspberry Pi*, Arduino, sensores de força e acelerômetro, com também implementaram um movimento balístico que permite ao robô andar com marchas específicas.

O presente trabalho, por sua vez, consiste na implementação de um controle de estabilidade na plataforma que a permita responder a distúrbios externos, tais como empurrões ou irregularidade no terreno, de modo a manter-se na posição desejada, seja parado ou durante o movimento, sem tombar.

Para tal, o distúrbio é, primeiramente, detectado pelo acelerômetro da plataforma. Um controlador, posteriormente, utiliza deste sinal para determinar a resposta desejada de cada pata para a correção do distúrbio e, por fim, esta resposta é acoplada à posição desejada em regime permanente para determinar a posição de cada motor a cada período de amostragem.

3.2 Arquitetura do Robô

O robô quadrúpede possui um total de 12 motores, sendo 3 em cada pata, como pode ser observado na Figura 3.1. O motor representado mais acima, em cada pata, tem o objetivo de movimentá-la no sentido transversal, isto é, para a esquerda ou direita. Estes motores serão os responsáveis pela correção do distúrbio de rolagem. Os outros dois motores se movimentam no sentido sagital, ou seja, para frente e para trás e, portanto, irão corrigir o distúrbio que provocaria a arfagem. Um destes motores, o mais abaixo de cada pata, atua como um joelho, dividindo-a em duas partes e adicionando um grau de liberdade.

A identificação numérica (de 1 a 12) de cada motor pode ser organizada de acordo com sua pata e o distúrbio que corrige conforme a Tabela 1.

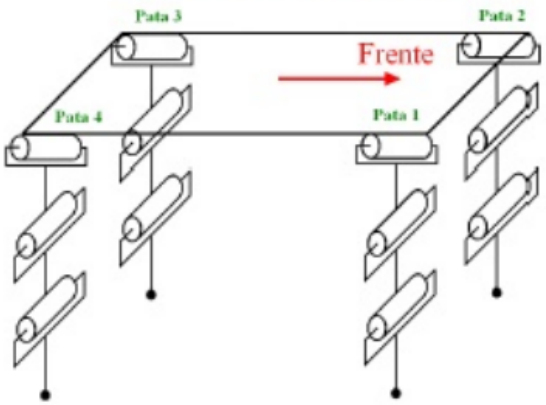


Figura 3.1: Representação gráfica da plataforma quadrúpede (Fonte: [8])

3.2.1 Motores

Os motores utilizados na plataforma são do modelo RX-28 da DYNAMIXEL como o da Figura 3.2 [7]. Este motor, conforme indicado em seu manual[1], possui dois modos de operação: o modo posição (*joint mode*) e o modo velocidade (*wheel mode*).

O modo posição possibilita a determinação da posição ângular de cada junta através da escrita dos bytes 30 e 31 da memória de cada motor. Como pode ser observado na Figura 3.3, dispõem-se de 9 bits (ou 1024 níveis de quantização) para representar ângulos de 0^o até 300^o (há uma zona inválida entre 300^o e 360^o). Dessa forma, pode-se escrever a posição angular desejada com uma resolução de aproximadamente 0.2933^o. Portanto, a relação entre a posição desejada, em graus, e o valor quantizado que deve ser escrito na memória é

Tabela 1: Identificação dos motores

Identificação do motor	Perna	Movimento de Correção
1	3	Rolagem
2	3	Arfagem
3	3	Arfagem (joelho)
4	4	Rolagem
5	4	Arfagem
6	4	Arfagem (joelho)
7	1	Rolagem
8	1	Arfagem
9	1	Arfagem (joelho)
10	2	Rolagem
11	2	Arfagem
12	2	Arfagem (joelho)

[1] http://support.robotis.com/en/product/actuator/dynamixel/rx_series/rx-28.htm#Actuator_Address_06

Figura 3.2: Motor Rx-28 da Dynamixel (Fonte: [8])

$$B_\theta = \frac{1023}{300}\theta_i = 3.41\theta_i, \tag{3.1}$$

em que B_θ é o valor quantizado que deve ser enviado em formato binário para a memória e θ_i é o ângulo desejado para o i-ésimo motor, em graus.

O modo velocidade, por outro lado, permite a escrita da velocidade que o motor irá aplicar à sua respectiva junta. Neste modo, 10 bits são utilizados para escrever a velocidade, sendo o bit mais significativo utilizado para a determinação do sentido de deslocamento (0 para anti-horário e 1 para horário) e os outros 9 para a determinação do módulo da velocidade. Ao contrário do modo posição, o valor máximo de velocidade não é fixo, mas varia conforme a tensão aplicada no motor. Segundo o manual[2], quando aplicados 16V, a velocidade máxima que pode ser atingida é de 79.4 rpm. Portanto, a velocidade máxima pode ser determinada por

$$\omega_{max} = V\frac{79.4}{16}, \tag{3.2}$$

em que ω_{max} é a velocidade máxima, em rpm, que pode ser aplicada a cada motor e V é a sua respectiva tensão. Os motores, na configuração atual da plataforma, trabalha com um nível de tensão de 13.3 V e, portanto, permitem uma velocidade de até 66 rpm, ou 6.91 rad/s. Dessa forma, se ω_i é a velocidade desejada do i-ésimo motor, em rad/s, o valor quantizado B_ω que deve ser enviado à memória (nos bytes 32 e 33) do motor é

$$B_\omega = \frac{1023}{6.91}|\omega_i| + 1024u(-\omega_i) = 148|\omega_i| + 1024u(-\omega_i), \tag{3.3}$$

em que a função u(x) é a função degrau. Ou seja, no caso da equação 3.3, se ω_i for maior que 0 (sentido anti-horário) então o décimo bit de B_ω será igual a zero e se ω_i for menor que zero

[2]http://www.crustcrawler.com/motors/RX28/docs/RX28_Manual.pdf

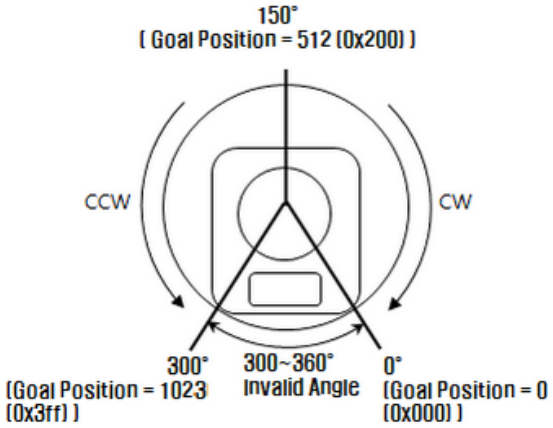


Figura 3.3: Configurações do Modo Posição (Fonte: [3])

(sentido horário), então adicionar-se-a 1024 ao valor que define o módulo, de forma que o décimo bit seja configurado como 1. O caso em que ω_i é igual a zero não afeta a análise da função degrau uma vez que, se os 9 bit menos significativos forem 0 o motor não irá gerar qualquer movimento, independentemente do valor do décimo bit.

Segundo o manual, para definir qual o modo de operação será utilizado, deve-se alterar os ângulos limites de cada motor, permitindo ou não que o movimento seja limitado. Dessa forma, para selecionar o modo velocidade, os ângulos mínimo e máximo devem ser ambos iguais a zero. Consequentemente, o sistema identifica que não deve haver restrição de movimento. Já para selecionar o modo posição, nenhum dos dois pode ser zero, havendo, dessa maneira, uma restrição do movimento. Para garantir a maior abrangência do movimento de cada motor no modo posição, pode-se definir o ângulo mínimo como 1 (valor quantizado) e o ângulo máximo como 1023. A definição dos ângulos mínimo e máximo pode ser realizada através dos bytes de 6 a 9 da memória.

3.2.2 Sistema Embarcado

Os principais dispositivos do sistema embarcado são o *Raspberry Pi* e o *Arduino*. O *Arduino* foi configurado, nos trabalhos anteriores com a plataforma, para receber os dados de sensoriamento do robô, que incluem os sensores de força presentes em cada pata e o acelerômetro. Ele, então, envia para o *Raspberry Pi*, via comunicação serial, estes dados já com um tratamento inicial [8]. No caso específico do acelerômetro, o Arduino envia não apenas os dados dos três eixos (X,Y e Z) mas também os ângulos correspondentes de rolagem e arfagem já calculados.

O *Raspberry Pi*, por sua vez, é a CPU do sistema, recebendo as medições através da conexão com o *Arduino*, calculando a resposta desejada para cada instante de tempo e enviando as instruções para os motores, além de tratar os dados, salvá-los, definir as threads periódicas de controle, escrever as informações na tela e etc. Essas funções são definidas em código em C++, sendo utilizadas as bibliotecas da *Dynamixel* para escrita e leitura da memória de cada motor. A interação do usuário com o *Raspberry Pi* é feita via SSH, utilizando o software *PuTTY*. A Figura 3.4 mostra

[3] http://support.robotis.com/en/product/actuator/dynamixel/rx_series/rx-28.htm#Actuator_Address_06

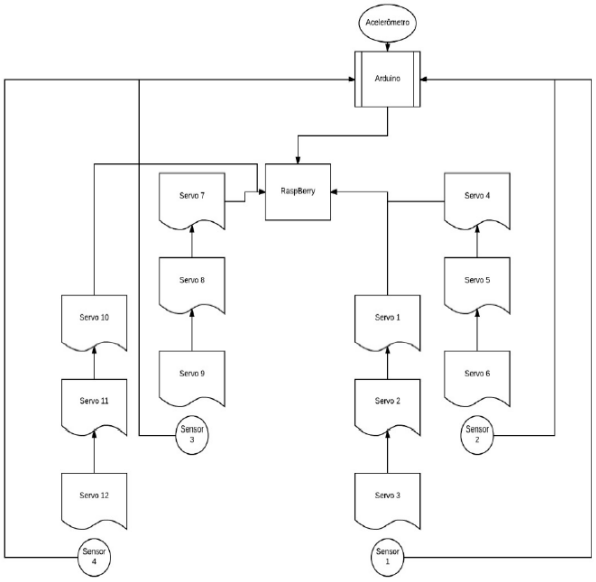


Figura 3.4: Arquitetura do Sistema Embarcado (Fonte: [8])

o esquemático do sistema embarcado acima descrito.

3.3 gDataLogger

O *gDataLogger* é um código em C desenvolvido pelo professor Geovany Araújo Borges para que os dados adquiridos durante o processo (como os ângulos de rolagem e arfagem, a posição dos motores e etc) sejam armazenados em um arquivo com extensão .mat para a posterior leitura pelo software *Matlab*.

O *gDataLogger* foi acoplado no código desenvolvido para o recolhimento e análise dos dados de cada experimento do controle de estabilidade. As principais funções para realizar tal recolhimento são: declarar as variáveis, inserir um valor na variável, atualizar o programa e fechá-lo. Além disso, foi desenvolvido um código no próprio *Matlab* para recolher as informações de cada arquivo .mat, organizá-las e, por fim, plotá-las em um gráfico de forma adequada para a análise.

3.4 Thread Periódica

Segundo [16], "um sistema real é um sistema que deve satisfazer restrições explícitas e delimitadas de tempo de resposta ou sofrer consequências graves, incluindo a falha". Ainda pela mesma referência, tempo de resposta é o "tempo entre a apresentação das entradas ao sistema e o aparecimento das suas respectivas respostas". Dessa forma, o sistema real deve conseguir responder de forma satisfatória às entradas, dentro de um período específico de tempo.

Dada esta definição de sistema em tempo real, nota-se que o sistema de equilíbrio do robô quadrúpede pode ser considerado como tal, uma vez que este necessita responder ao distúrbio

(que, neste caso, é a entrada do sistema) dentro de certos limites de tempo, caso contrário haverá falha ou, no âmbito desta análise, a queda do robô.

Para lidar com tal sistema como um sistema real, implementou-se o controle de estabilidade como uma thread periódica, com período de amostragem T_{am}. Uma thread é um processo que é aberto pelo sistema de processamento (neste caso do *Raspberry Pi*), para ser rodado de forma paralela com outros. Isso permite que o processo não sofra a influência dos demais, especialmente no tempo de resposta (o processo não precisa esperar que outro termine para poder começar). Uma thread periódica, por sua vez, é uma thread que é chamada a cada período de tempo, neste caso, a cada T_{am} segundos.

Desta forma, a implementação do controle de estabilidade como uma thread periódica permite que este não seja influenciado por outros processos, de modo que o tempo de resposta para a correção dos distúrbios fique em limiares aceitáveis para que não haja falhas. No código desenvolvido, a função *main*, depois de configurar todos os parâmetros necessários, cria um timer com T_{am} segundos de periodicidade que, por sua vez, chama a thread de controle sempre que o tempo de amostragem é atingido. Asssim, dois processos acontecem simultaneamente: a função *main*, que determina as condições de parada do código e escreve informações na tela, e a função de controle que lê os dados do acelerômetro, calcula as devidas respostas e as envia para os motores.

O valor de T_{am} foi, a princípio, estipulado como 0.1 segundos. Porém, percebeu-se através de diversos testes, e com auxílio do gDataLogger para a medição do tempo, que a leitura da própria função de controle (linha a linha do código) durava cerca de 35 a 40 ms. Este tempo representa aproximadamente 40% do tempo de amostragem proposto sendo, dessa forma, uma valor bastante elevado e que pode prejudicar a dinâmica do sistema em tempo real. Por este motivo, estabeleceu-se que a leitura do código não deveria exceder em 20% o tempo de amostragem.

Após várias testes para tentar diminuir o tempo de leitura do código, descobriu-se que este tempo elevado ocorre devido às leituras e escritas na memória dos motores, utilizadas para obter os seus valores de posição e velocidade assim como aplicá-las. Uma vez que estas funções são essenciais para o desenvolvimento do projeto e, portanto, não podem ser eliminadas, decidiu-se por aumentar o tempo de amostragem T_{am} para 0.2 segundos. Neste caso, o tempo de leitura do código passa a ser aproximadamente 20% de T_{am} e, assim, um valor aceitável para o sistema em tempo real.

3.5 Controle de Estabilidade

O princípio do controle de estabilidade adotado pelo presente trabalho se baseia, em grande parte, no trabalho desenvolvido em [1] e em [5]. O princípio básico deste controle consiste na utilização de sensores para a aquisição das informações do estado atual do robô e a subsequente aplicação de respostas aos motores baseadas nestes dados. Mais especificamente, a resposta de cada motor i será dada pela multiplicação do distúrbio detectado pelos sensores por um ganho K_i obtido empiricamente para cada junta.

Para a implementação do controle desejado, o sensoriamento será feito através do acelerômetro, responsável pela detecção de distúrbios no robô que possam gerar os movimentos indesejados de rolagem, d_{roll} (queda para direita ou esquerda) ou arfagem, d_{pitch} (queda para frente ou para trás). Então, a resposta da i-ésima junta para compensar tal movimento será dada por

$$q_i(t) = \pm K_i d_{roll/pitch}, \tag{3.4}$$

em que q_i é a resposta do i-ésimo motor (sua posição ou velocidade), K_i é um ganho que será determinado empiricamente para cada junta e $d_{roll/pitch}$ é o distúrbio detectado pelo acelerômetro[4].

O sinal de $q_i(t)$ será determinado empiricamente de modo que a resposta da i-ésima junta seja tal que gere um movimento final no robô que se oponha ao movimento gerado pelo distúrbio. Por exemplo, se o acelerômetro detecta uma queda para à esquerda do robô, as juntas responsáveis pela rolagem devem se movimentar no sentido tal que o robô sofra um movimento para a direita.

Podemos observar que a equação 3.4 é muito semelhante com a equação **??** de [5]. Como a abordagem a ser utilizada também é empírica na determinação de K_i, incorporaremos o sinal $\pm$ ao ganho, assumindo, dessa forma, que determinaremos o sinal através do parâmtero K_i. Assim, K_i será positivo ou negativo dependendo do sentido necessário de cada junta para corrigir o distúrbio.

O distúrbio, $d_{roll/pitch}$, por sua vez, será tratado no presente trabalho como a diferença entre o valor de referência desejado para aquela quantidade (posição ou velocidade) e o valor atual da mesma (medida pelo acelerômetro). Isto se representará como

$$d_{roll/pitch} = r_{roll/pitch} - m_{roll/pitch}, \tag{3.5}$$

em que $r_{roll/pitch}$ é o valor de referência, que no geral será igual a zero, e $m_{roll/pitch}$ é o valor medido a cada instante de amostragem. A equação, portanto, que resume a ideia básica do controle proposto é

$$q_i(t) = K_i(r_{roll/pitch} - m_{roll/pitch}). \tag{3.6}$$

O sistema básico acima descrito pode ser melhor visualizado pela Figura 3.5.

3.5.1 Abordagem Inicial

A ideia inicial para o controle de estabilidade se baseia na utilização de velocidades ao invés de posições e, portanto, do modo velocidade dos motores. Esta abordagem se justifica pela necessidade

[4] A notação $X_{roll/pitch}$, que será utilizada neste trabalho, indica que o termo pode ser substituído por X_{roll} ou X_{pitch} dependendo do motor em questão. Por exemplo, o motor 1 corrige o movimento de rolagem então os termos $X_{roll/pitch}$ serão substituídos por X_{roll}.

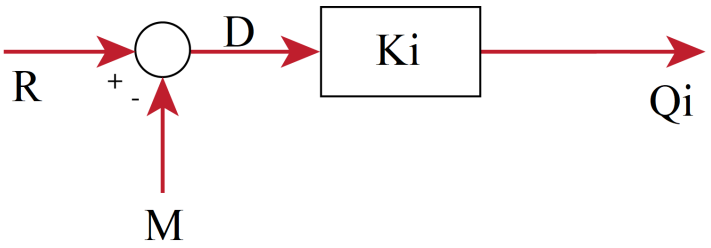


Figura 3.5: Diagrama de Blocos do Sistema Básico

de uma resposta rápida e não cadenciada das juntas. Consequentemente, os termos da equação 3.6 serão caracterizados como velocidades, isto é, $r_{roll/pitch}$ será a velocidade angular desejada para a rolagem ou arfagem e $m_{roll/pitch}$, a velocidade atualmente medida para estes movimentos, que reescreveremos como $\omega_{roll/pitch}$. Da mesma forma, $q_i(t)$ será descrito como a velocidade do motor i no instante t, ou seja, $\omega_i(t)$. Remodelando a equação 3.4 para as novas definções tem-se

$$\omega_i(t) = K_i(r_{roll/pitch} - \omega_{roll/pitch}). \tag{3.7}$$

Como deseja-se a estabilidade do robô, $r_{roll/pitch}$ será estipulado como igual a zero, uma vez que movimentações que provocam rolagem ou arfagem podem levá-lo à queda. Dessa forma, a equação anterior resultará em

$$\omega_i(t) = K_i(-\omega_{roll/pitch}). \tag{3.8}$$

Como o acelerômetro entrega apenas os ângulos de rolagem e arfagens medidos, deve ser feita uma derivação para obter as velocidades $\omega_{roll/pitch}$. Dessa forma

$$\omega_i(t) = -K_i \frac{\mathrm{d}}{\mathrm{d}t}\left(\theta_{roll/pitch}\right), \tag{3.9}$$

em que $\theta_{roll/pitch}$ é o ângulo de rolagem ou arfagem medido pelo acelerômetro. O diagrama de blocos resultante desta configuração pode ser visualizado na Figura 3.6.

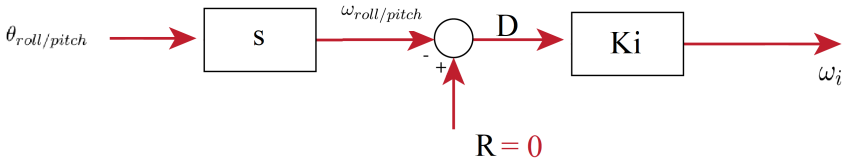


Figura 3.6: Diagrama de Blocos do Sistema no Modo Velocidade

Uma vez que o sistema está digitalizado, as equações devem ser discretizadas com o tempo de amostragem T_{am}. A equação 3.9 discretizada, portanto, resulta em

$$\omega_i[k] = -K_i \frac{\theta_{roll/pitch}[k] - \theta_{roll/pitch}[k-1]}{T_{am}}. \tag{3.10}$$

3.5.2 Filtro

A derivação dos ângulos medidos pelo acelerômetro deixa o sistema suscetível a ruídos, uma vez que as frequências mais elevadas (comum em ruídos) são destacadas com um ganho maior. Para mitigar este efeito, adicionou-se um filtro passa-baixas com ganho DC unitário e frequência de corte de f_c após a derivação. A função de transferência de tal filtro é dada por

$$G_{filtro}(s) = \frac{2\pi f_c}{s + 2\pi f_c}. \tag{3.11}$$

A equação que descreve o filtro de forma discretizada é

$$A_{out}[k] = \frac{1}{1 + T_{am}2\pi f_c} A_{out}[k-1] + \frac{T_{am}2\pi f_c}{1 + T_{am}2\pi f_c} A_{in}[k]. \tag{3.12}$$

Como a frequência de amostragem $f_{am} = \frac{1}{T_{am}} = 5Hz$, é razoável estipular a frequência de corte do filtro com o valor de 1Hz. Como $T_{am} = 0.2$ e $f_c = 1$ a equação 3.12 resulta em

$$A_{out}[k] = 0.4431 A_{out}[k-1] + 0.5569 A_{in}[k]. \tag{3.13}$$

O diagrama de blocos com filtro pode ser visualizado na Figura 3.7.

3.5.3 Torque no Modo Velocidade

Conforme será melhor aprofundado no capítulo de Resultados, após alguns testes, percebeu-se que os motores no modo velocidade não exibem torque suficiente para sustentar o próprio peso. Nestes casos, as juntas se movem facilmente por ações externas ao invés de manter a posição fixa

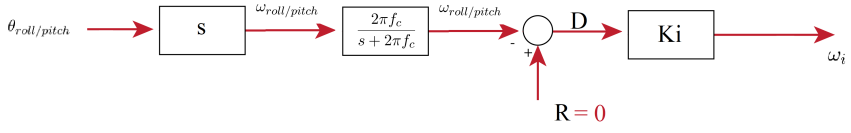


Figura 3.7: Diagrama de Blocos do Sistema no Modo Velocidade com Filtro

(para velocidades nulas). Isto resulta em casos em que, mesmo sem distúrbio, as juntas acabam se movendo pela ação do peso do robô sobre elas e gerando, como resultado, a queda do robô.

Por outro lado, o modo posição apresenta um torque elevado, de modo que ao determinar uma posição específica para cada junta, esta se mantém fixa e não cede facilmente a forças aplicadas externamente. Não foi encontrado ainda o motivo dessa diferença entre os dois modos de operação. Porém, uma vez reconhecida esta diferença, decidiu-se trabalhar no modo posição para obter um controle de estabilidade mais adequado.

3.5.4 Controle de Estabilidade no Modo Posição

Uma vez decidido pelo uso do modo posição dos motores ao invés do modo velocidade, algumas alterações ao modelo anteriormente proposto foram necessárias. A equação 3.10 determina qual a velocidade $\omega_i[k]$ que deve ser aplicada ao i-ésimo motor no instante k. No entanto, como passou-se a trabalhar no modo posição, a informação que deve ser enviada a cada motor é a sua posição e não a sua velocidade. Dessa forma, para obter a posição desejada de cada motor, basta integrar ω_i. No cenário discretizado em que estamos trabalhando, portanto, a posição angular $\theta_i[k]$ de cada motor no instante k será dada por

$$\theta_i[k] = \theta_i[k-1] + T_{am}\omega_i[k] = \theta_i[k-1] - T_{am}K_i\left(\frac{\theta_{roll/pitch}[k] - \theta_{roll/pitch}[k-1]}{T_{am}}\right). \quad (3.14)$$

Dessa forma, o diagrama de blocos correspondente a esta nova estrutura adiciona um integrador ao final, como mostra a Figura 3.8.

3.5.5 Determinação dos Ganhos K_i

Com o sistema de controle de equilíbrio definido e as juntas capazes de sustentar o peso do robô, foi possível realizar os testes das respostas dos motores aos distúrbios. Dessa forma, pode-se determinar empiricamente o valor de K_i para cada motor de modo a corrigir o distúrbio da maneira mais eficiente possível.

Para auxiliar na determinação destes ganhos utilizamos a simetria do robô para agrupar os motores que devem responder de maneira simultânea e no mesmo sentido. No caso da correção do movimento de rolagem, por exemplo, podemos dividir os grupos em patas da direita e patas da esquerda. Neste caso, os motores referentes às patas da esquerda que corrigem a rolagem terão

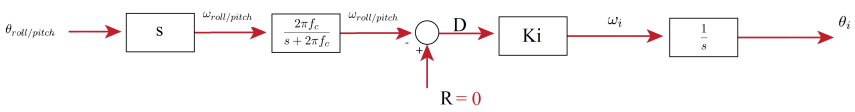


Figura 3.8: Diagrama de Blocos do Sistema no Modo Posição

um ganho $K_i = K_{roll\_L}$ enquanto os que são referentes às patas da direita obedecerão à relação $K_i = K_{roll\_R}$.

Da mesma forma, a correção do movimento de arfagem pode ser feita com as patas da frente, com um ganho K_{pitch_F}, ou com as patas de trás, com um ganho K_{pitch_B}. No entanto, há uma diferença para o caso da arfagem uma vez que os motores que fazem a correção neste sentido podem, ainda, ser divididos em motores de joelho, com um ganho K_{down}, ou os de cima, K_{up}. Dessa forma, o ganho referente aos motores que corrigem o movimento de arfagem terá duas componentes, uma indicando a localização da pata (na frente ou atrás) e outra indicando se a junta é ou não um joelho. O ganho será dado, portanto, pela multiplicação das duas componentes. Por exemplo, um motor da frente que não esteja em uma junta do tipo joelho, terá o ganho dado por $K_i = K_{pitch\_F}K_{up}$.

Assim, observando a Tabela 1, podemos determinar cada K_i:

$$\begin{aligned} K_{4/7} &= K_{roll\_R}, \\ K_{1/10} &= K_{roll\_L}, \\ K_{2/5} &= K_{pitch\_B}K_{up}, \\ K_{3/6} &= K_{pitch\_B}K_{down}, \\ K_{8/11} &= K_{pitch\_F}K_{up}, \\ K_{9/12} &= K_{pitch\_F}K_{down}. \end{aligned} \tag{3.15}$$

Nesta equação, a representação $K_{m/n}$ indica que ambos K_m e K_n são igual ao valor do outro lado da igualdade.

Para que cada par de motores acima agrupados possa seguir o movimento no mesmo sentido alguns valores de K_i devem ser corrigidos com um sinal de negativo, uma vez que alguns motores foram montados em configurações contrárias. Tais motores são: 5, 6, 8 e 9. O motor 7, por sua vez, apresenta um amortecimento levemente maior em relação ao seu par e, portanto, foi multiplicado por 1.1 para a devida correção.

A definição do valor de cada uma dessas componentes será feita de forma empírica através de testes reais. Neste sentido, o primeiro fato detectado através dos testes é que a resposta de cada pata deve seguir o movimento do distúrbio. Por exemplo, se o robô é empurrado para a direita, então as patas devem se deslocar para a direita para que o distúrbio seja corrigido. O mesmo se aplica para o movimento de arfagem.

No entanto, as juntas do tipo joelho diferem das outras neste sentido. Em tal caso, percebeu-se, que o mais adequado para estas juntas é que sigam o movimento contrário ao distúrbio. Isso ocorre porque, ao seguir este sentido, a junta do joelho orienta a pata a aterrissar na vertical. Podemos visualizar melhor os possíveis cenários de orientação desta junta durante a aterrissagem através da Figura 3.9.

A Fig. 3.9(a) mostra a pata em condição de sustentação, sem correção de distúrbio, como referência. A Fig. 3.9(b), por sua vez, apresenta o caso em que a junta do joelho não muda sua

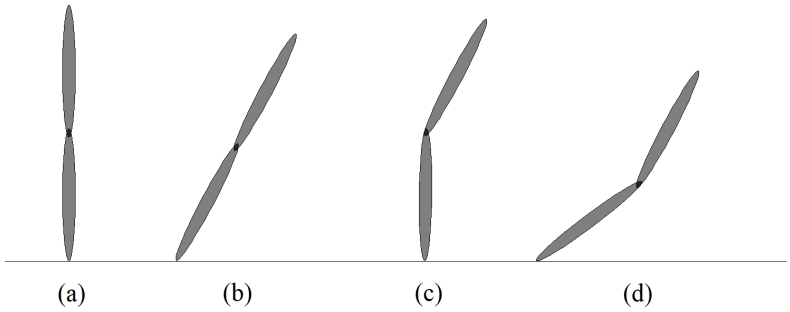


Figura 3.9: Configurações das juntas tipo joelho durante a aterrissagem

configuração durante o distúrbio, ocasionando em uma aterrissagem diagonal. Dessa forma, a pata fica suscetível a deslizar horizontalmente. Já a Fig. 3.9(c) indica a configuração aproximada da pata quando o joelho se move em oposição ao distúrbio. É possível notar que a pata tem uma possibilidade mínima de deslizamento neste caso. Por fim, na Fig. 3.9(d), é possível visualizar a posição da pata quando o joelho se desloca no sentido do distúrbio. Neste cenário, a pata também pode sofrer o deslizamento horizontal, cedendo ao peso do robô ocasionando em sua queda.

Este movimento do joelho de cada pata não exige um deslocamento elevado quando comparado com as demais juntas. Por isso, determinou-se através dos testes que o valor mais adequado para K_{down} é -0.4. Este valor possibilita que os joelhos se movimentem o suficiente para garantir uma aterrissagem sem deslizamentos quando o robô sofre um distúrbio de arfagem. Os demais motores que a corrigem não sofrem influências relevantes por movimentarem a parte de cima da pata e, portanto, pode ser caracterizados com $K_{up} = 1$.

A análise seguinte que foi observada através dos testes com a plataforma foi que as patas que devem corrigir o distúrbio devem ser apenas aquelas referentes à direção do mesmo. Por exemplo, se é detectado um empurrão no robô que o faria tombar para a direita então apenas as patas da direita devem responder a tal. Isso pode ser descrito através da equação

$$K_{roll/pitch\_j} = \begin{cases} K_{resp\_j} & \text{se } \omega_{roll/pitch} \text{ for no sentido j,} \\ 0 & \text{caso contrário.} \end{cases} \tag{3.16}$$

Isto é, se a velocidade detectada ocorrer no sentido j (esquerda, direita, para frente ou para trás), o ganho referente a este sentido será diferente de zero e igual ao ganho de resposta, K_{resp_j}, a ser caracterizado empiricamente. Caso não seja detectado nenhum distúrbio no sentido j o ganho referente àquela direção será igual a zero.

Os testes realizados na plataforma indicaram que o valor de K_{resp_j} mais adequado para o controle de estabilidade é igual a 1.5 para todos os sentidos.

3.5.6 Paralelismo com a Posição Desejada

Determinados os ganhos K_i de todos os motores, o controle de estabilidade está apto a corrigir os distúrbios, conforme será melhor abordados no capítulo dos Resultados. No entanto, o controle proposto até aqui, caracterizado pela equação 3.14, aplica uma determinada posição θ_i em cada junta e não a modifica mais até a ocorrência de um novo distúrbio. Isto resulta na permanência do robô na posição de correção, mesmo após o fim do distúrbio.

A ação mais adequada para o quadrúpede é de, após a correção do distúrbio, cada junta voltar a uma posição desejada para o regime permanente. Dessa forma, se o robô estiver parado, deseja-se que ele volte à posição inicial após a correção. Da mesma maneira, caso a plataforma esteja em processo de marcha, ela deve continuar sua movimentação normalmente após a estabilidade ser alcançada.

À vista disso, podemos remodelar o controle de equilíbrio com um novo diagrama de blocos, como o da Figura 3.10. Podemos notar que, neste novo modelo, a posição desejada para cada junta é a soma de duas componentes: uma determinada pelo controle de estabilidade $G_1(s)$, que tem como entrada o distúrbio detectado pelo acelerômetro e a outra determinada pela posição desejada em regime permanente θ_i^*. Esta será definida pelo geração de marchas da plataforma mas, inicialmente, será dada apenas pela posição inicial das juntas para que o robô permaneça parado.

Nesta remodelação, consideraremos $d_{roll/pitch}$ agora como o erro da posição angular dos ângulos de rolagem e arfagem detectados pelo acelerômetro, e não mais suas velocidades. Consequentemente

$$d_{roll/pitch} = r_{roll/pitch} - \theta_{roll/pitch}, \tag{3.17}$$

e, como deseja-se a estabilidade do robô, considerar-se-a $r_{roll/pitch} = 0$. Ou seja, o controle será a feito de modo a manter os ângulos de rolagem e arfagem próximos ou iguais a zero. Logo

$$d_{roll/pitch} = -\theta_{roll/pitch}. \tag{3.18}$$

A posição do i-ésimo motor, no domínio s será

$$\Theta_i(s) = G_1(s)D(s) + G_2(s)\Theta_i^*(s). \tag{3.19}$$

Para determinar os controladores $G_1(s)$ e $G_2(s)$ mais adequados, analisaremos as condições de regime permanente desejadas através do teorema do valor final. Segundo este

$$\lim_{t\to\infty} \theta_i(t) = \theta_i(\infty) = \lim_{s\to 0} s\Theta_i(s) = \lim_{s\to 0} s(G_1(s)D(s) + G_2(s)\Theta_i^*(s)). \tag{3.20}$$

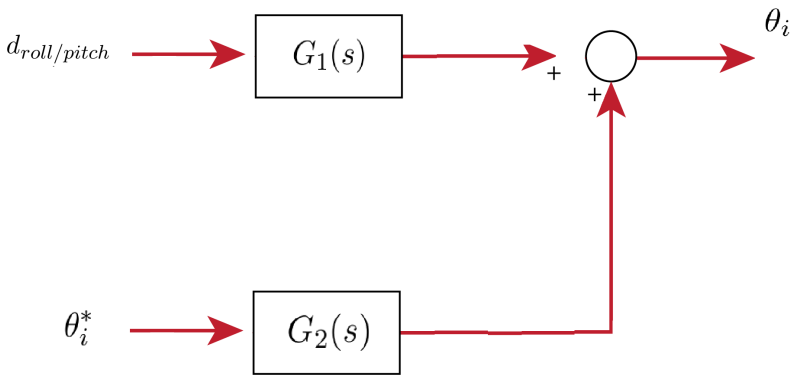


Figura 3.10: Diagrama de Blocos com Paralelismo

Deseja-se que $\theta_i(\infty) = \theta_i^*(t)$. Para tal, a componente referente ao distúrbio $sG_1(s)D(s)$ deve ser anulada quando s tende a zero, enquanto a componente $sG_2(s)\Theta_i^*(s)$ deve ser igual a $\Theta_i^*(s)$, nesta mesma condição. Para que a primeira condição seja satisfeita, consideremos um distúrbio do tipo degrau, ou seja, $D(s) = \frac{D_k}{s}$, em que D_k é a amplitude do degrau. Logo

$$sG_1(s)D(s) = D_kG_1(s). \tag{3.21}$$

Para que esta equação seja nula no limite de s tentendo a zero, basta que $G_1(s)$ tenha um zero em $s = 0$. Isto é

$$G_1(s) = \frac{K_ds}{H_1(s)}, \tag{3.22}$$

em que K_d é um ganho que será determinado posteriormente baseado nas abordagens anteriores e $H_1(s)$ é o denominador de $G_1(s)$. Podemos observar que a presença do termo s no numerador da função de transferência implica que a resposta ao distúrbio deve ser alocada em frequências maiores. Dessa forma, podemos caracterizar $G_1(s)$ como um filtro passa-alta, uma vez que a componente DC é desejada como nula e as frequências mais elevadas são utilizadas para corrigir o distúrbio. Além do mais, essa caracterização nos permite determinar o tempo de resposta do sistema para a correção do equilíbrio. Assim

$$G_1(s) = \frac{K_ds}{\tau_ds + 1}, \tag{3.23}$$

em que τ_d é a constante de tempo do controlador de estabilidade.

Já para que a segunda condição seja atendida também consideraremos $\Theta_i^*(s)$ como um degrau, ou seja, $\Theta_i^*(s) = \frac{\Theta_i^*}{s}$, em que Θ_i^* é a amplitude do degrau. Dessa forma,

$$sG_2(s)\Theta_i^*(s) = G_2(s)\Theta_i^*. \tag{3.24}$$

Para que a equação acima seja igual Θ_i^* quando s tender a zero, $G_2(s)$ deve ser igual 1 nessas condições. Podemos concluir, portanto, que esta componente depende que um ganho unitário seja garantido em DC. A resposta em frequências maiores determina apenas o tempo de resposta do sistema para que este atinja o valor final desejado. Por isso, podemos implmentar $G_2(s)$ como um filtro passa-baixas com ganho DC unitário e frequência de corte variável, a ser determinada empiricamente para obtenção da resposta mais adequada. Portanto

$$G_2(s) = \frac{1}{\tau_p s + 1}, \tag{3.25}$$

em que τ_p é a constante de tempo da resposta em regime permanente (ou então o inverso da frequência de corte).

Portanto, a posição de cada motor será determinada, no domínio s, por

$$\Theta_i(s) = \frac{K_d s}{\tau_d s + 1} D(s) + \frac{1}{\tau_p s + 1}\Theta_i^*(s). \tag{3.26}$$

Reorganizando a equação 3.26 e passando-a para o domínio do tempo, podemos obter a equação diferencial que representa o sistema proposto como

$$\ddot{\theta}_i(t)\tau_d\tau_p + \dot{\theta}_i(t)(\tau_d + \tau_p) + \theta_i(t) = K_d\tau_p\ddot{d}(t) + K_d\dot{d}(t) + \tau_d\dot{\theta}_i^*(t) + \theta_i^*(t). \tag{3.27}$$

Para obter a equação do sistema discretizado, representar-se-a a quantidade $x(t)$ (em que x pode ser θ_i, d ou θ_i^*) como $x[k]$ e as suas derivadas como

$$\begin{aligned} \dot{x}(t) &= \frac{x[k] - x[k-1]}{T_{am}}, \text{e} \\ \ddot{x}(t) &= \frac{x[k] - 2x[k-1] + x[k-2]}{T_{am}^2}. \end{aligned} \tag{3.28}$$

Dessa forma, a equação de diferenças do sistema discretizado será

$$\theta_i[k]\,(\alpha + \beta + 1) + \theta_i[k-1]\,(-2\alpha - \beta) + \theta_i[k-2]\,(\alpha) = b[k], \tag{3.29}$$

em que $b[k]$ será dado como

$$b[k] = d[k]\,(\gamma + \eta) + d[k-1]\,(-2\gamma - \eta) + d[k-2]\,(2\gamma) + \theta_i^*[k]\,(\lambda + 1) - \theta_i^*[k-1]\lambda, \quad (3.30)$$

e os coeficientes α, β, γ, η e λ são dados por

$$\begin{aligned} \alpha &= \frac{\tau_d \tau_p}{T_{am}^2}, \\ \beta &= \frac{(\tau_d + \tau_p)}{T_{am}}, \\ \gamma &= \frac{K_d \tau_p}{T_{am}^2}, \\ \eta &= \frac{K_d}{T_{am}}, \text{e} \\ \lambda &= \frac{\tau_d}{T_{am}}. \end{aligned} \quad (3.31)$$

Portanto, o ângulo $\theta_i[k]$ que deverá ser enviado para cada motor i no instante k é dado por

$$\theta_i[k] = \frac{b[k] - \theta_i[k-1]\,(-2\alpha - \beta) - \theta_i[k-2]\alpha}{\alpha + \beta + 1}. \quad (3.32)$$

Para determinar o parâmetro K_d do controlador, podemos observar comparativamente as equações 3.14 e 3.26. No primeiro caso, é possível observar que a posição final de cada motor é dado pela soma de uma posição prévia a uma velocidade multiplicada pelo tempo de amostragem e uma constante adimensional. A análise dimensional, portanto, confere para a dimensão de posição angular. Já para o segundo caso, como já foi mencionado anteriormente, o distúrbio $D(s)$ é uma posição angular. A composição $sD(s)$, dessa forma, é a velocidade angular do distúrbio, assim como $\omega_{roll/pitch}$ da primeira equação. Assim, para que a resposta do controlador seja similar ao determinado anteriormente, K_d deve ser a mesma constante multiplicando a velocidade, isto é, $K_i T_{am}$. Portanto,

$$K_d = K_i T_{am}. \quad (3.33)$$

Como já mencionado anteriormente, as constantes τ_d e τ_p serão determinada empiricamente através de testes reais na plataforma. Tais experimentos determinarão o tempo mais adequado de resposta para a correção de equilíbrio e para que o robô atinja a resposta em regime permanente. Uma vez que estes valores estiverem estipulados, todos os parâmetros da equação 3.31 estarão devidamente determinados.

3.6 Estrutura Geral do Código

Podemos, então, condensar as principais informações descritas nas seções anteriores para resumir a estrutura básica do código desenvolvido para o controle de estabilidade. O código foi escrito

em C++ e possui as funções *main* e controle como os processos principais.

A função *main*, em um primeiro momento, faz todas as inicializações necessárias. Isto inclui abrir a comunicação com os motores e com o *Arduino*, declarar as variáveis que serão alocadas no gDataLogger para aquisição de dados, ativar o modo posição dos motores e declarar a posição inicial assim como enviá-la para os motores. Em seguida, ela espera alguma tecla ser pressionada para seguir os próximos passos.

Uma vez detectada resposta do teclado, a função *main* cria um timer com tempo de amostragem T_{am} que é configurado para chamar a função de controle sempre T_{am} segundos forem decorridos. Com isso, a thread de controle é criada e a função *main*, a partir de então, permanece em um loop apenas para verificar se a condição de parada foi satisfeita (neste caso a condição de parada é um novo toque em qualquer tecla) e atualizar o gDataLogger.

Se detectada a condição de parada, a função *main* para o timer, fecha o gDataLogger e a conexão com o *Arduino*, libera os motores e, se configurado, exporta os dados para um diretório no site *GitHub*, onde os dados podem ser acessados.

Já a função de controle, a cada vez que é chamada, começa com a medição dos dados do acelerômetro. Em seguida ela passa os dados pelo filtro e calcula os ganhos K_i com base nos dados obtidos. Posteriormente, ela passa os dados para o controlador que, após os devidos cálculos, retorna a posição desejada para a i-ésima junta que, então, é mandada para seu respectivo motor. Por fim os dados são atualizados para serem utilizados na próxima vez que a função for chamada. Para tal a maioria das variáveis são definidas como variáveis globais, assim elas podem ser armazenadas enquanto a thread não é chamada novamente.

Capítulo 4

Resultados Experimentais

Resumo opcional.

4.1 Introdução

Capítulo 5

Conclusões

Este capítulo é em geral formado por: um breve resumo do que foi apresentado, conclusões mais pertinentes e propostas de trabalhos futuros.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] RAIBERT, M. et al. Bigdog, the rough-terrain quaduped robot. *Boston Dynamics*.

[2] LI, M. et al. Control of a quadruped robot with bionic springy legs in trotting gait. *Journal of Bionic Engineering*, v. 11, n. 2, p. 188–198, 2014.

[3] SEMINI, C. et al. Design of hyq - a hydraulically and electrically actuated quadruped robot. *Journal of Systems and Control Engineering*, v. 225, p. 831–849, August 2011.

[4] HTTP://OLD.IIT.IT/EN/DLS-ROBOTS/HYQ-ROBOT.HTML. June 2014.

[5] SOUSA, J.; MATOS, V.; SANTOS, C. P. dos. A bio-inspired postural control for a quadruped robot: an attractor-based dynamics. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, p. 5329–5334, October 2010.

[6] MCGHEE, R. B.; FRANK, A. A. On the stability properties of quadruped creeping gaits. *Mathematical Biosciences*, v. 3, p. 331–351, 1968.

[7] SANTOS, J. H. S. Plataforma quadrupede: Uma nova estrutura para robo quadrupede do lara. July 2016.

[8] PORPHIRIO, C. de F.; SANTANA, P. H. M. Geração de marchas para a plataforma quadrúpede utilizando algoritmo genético. Junho 2017.

[9] SANTOS, P. G. de; GARCIA, E.; ESTREMERA, J. *Quadrupedal Locomotion: an introduction to the control of four-legged robots*. Berlin, Germany: Springer, 2007.

[10] RAIBERT, M. H.; CHEPPONIS, M.; BROWN, H. B. Running on four legs as though they were one. *IEEE Journal of Robotics and Automation*, RA-2, n. 2, p. 70–82, June 1986.

[11] MENG, J.; LI, Y.; LI, B. A dynamic balancing approach for a quadruped robot supported by diagonal legs. *International Journal of Advanced Robotic Systems*, v. 12, n. 142, 2015.

[12] WON, M.; KANG, T. H.; CHUNG, W. K. Gait planning for quadruped robot based on dynamic stability: landing accordance ratio. *Intel Serv Robotics*, v. 2, p. 105–112, March 2009.

[13] SOUTO, R. F. Modelagem cinemática de um robô quadrúpede e geração de seus movimentos usando filtragem estocástica. Julho 2007.

[14] RUNBIN, C. et al. Inverse kinematics of a new quadruped robot control method. *International Journal of Advanced Robotic Systems*, v. 10, n. 46, 2013.

[15] FEATHERSTONE, R. Quantitative measures of a robot's physical ability to balance. *The International Journal of Robotics Research*, v. 35, n. 14, p. 1681–1696, 2016.

[16] LAPLANTE, P. A. *Real-Time Systems Design and Analysis : an Engineer's Handbook*. New York, United States: iEEE Press, 1993.

ANEXOS

I. DIAGRAMAS ESQUEMÁTICOS

II. DESCRIÇÃO DO CONTEÚDO DO CD