

**UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
CURSO DE ENGENHARIA DE CONTROLE E AUTOMAÇÃO**

Davi Klein

**MODELAGEM DINÂMICA POR MEIO DO MÉTODO DO TRABALHO
VIRTUAL E CONTROLE PD DESCENTRALIZADO DE UMA
PLATAFORMA DE STEWART-GOUGH**

Santa Maria, RS
2022

Davi Klein

**MODELAGEM DINÂMICA POR MEIO DO MÉTODO DO TRABALHO VIRTUAL
E CONTROLE PD DESCENTRALIZADO DE UMA PLATAFORMA DE STEWART-
GOUGH**

Trabalho de Conclusão de curso apresentado ao Curso de Graduação em Engenharia de Controle e Automação da Universidade Federal de Santa Maria (UFSM, RS) como requisito parcial para a obtenção do grau de **Bacharel em Engenharia de Controle e Automação.**

Orientador: Prof Dr. Daniel Fernando Tello Gamarra

Santa Maria, RS
2022

Davi Klein

**MODELAGEM DINÂMICA POR MEIO DO MÉTODO DO TRABALHO VIRTUAL
E CONTROLE PD DESCENTRALIZADO DE UMA PLATAFORMA DE STEWART-
GOUGH**

Trabalho de Conclusão de curso apresentado ao Curso de Graduação em Engenharia de Controle e Automação da Universidade Federal de Santa Maria (UFSM, RS) como requisito parcial para a obtenção do grau de **Bacharel em Engenharia de Controle e Automação.**

Aprovado em 09 de março de 2022:

Daniel Fernando Tello Gamarra, Dr. (UFSM)
(Presidente/Orientador)

Anselmo Rafael Cukla, Dr. (UFSM)

Fabio Ecke Bisogno, Dr. (UFSM)

Santa Maria, RS
2022

AGRADECIMENTOS

Agradeço, primeiramente, a Deus por me abençoar e me dar as devidas condições para o desenvolvimento deste trabalho. Agradeço à minha mãe, Denise Hunsche Klein, ao meu pai Celson Klein (*in memoriam*) e a minha irmã Laura Klein por sempre terem me amado incondicionalmente e sempre terem sido os meus maiores incentivadores dos meus estudos. Agradeço ao meu orientador, Daniel Fernando Tello Gamarra, por ter aceitado o desafio de me orientar, pela confiança, pelo incentivo e dedicação nas orientações por ele dadas. Agradeço a todos os meus amigos que me incentivaram e, de alguma forma, me ajudaram no desenvolvimento deste trabalho: Jair Augusto Bottega, Lucas Cilião Guilherme e Eduardo Cattani Silva. Por último, mas não menos importante, agradeço a minha namorada, Sara de Mello Cardias, por todo amor, parceria, compreensão e força diária que necessitei durante o desenvolvimento deste trabalho.

Houve um tempo em que o homem enfrentou o universo sozinho e sem amigos. Agora ele tem criaturas para ajudá-lo; criaturas mais fortes que ele próprio, mais fiéis, mais úteis e totalmente devotadas a ele. A humanidade não está mais sozinha.

(Isaac Asimov)

RESUMO

MODELAGEM DINÂMICA POR MEIO DO MÉTODO DO TRABALHO VIRTUAL E CONTROLE PD DESCENTRALIZADO DE UMA PLATAFORMA DE STEWART-GOUGH

AUTOR: Davi Klein

ORIENTADOR: Daniel Fernando Tello Gamarra

Este trabalho apresenta a modelagem dinâmica por meio do método do trabalho virtual, bem como o controle Proporcional-Derivativo de trajetória de um manipulador robótico paralelo conhecido como plataforma de Stewart-Gough. Ainda, este trabalho contém a simulação das equações cinemáticas, dinâmicas e sistema de controle. Para realizar essas simulações, foram desenvolvidos algoritmos os quais foram implementados utilizando-se do *software Matlab®*. O trabalho obteve como resultados a modelagem dinâmica do sistema utilizando-se do método do trabalho virtual, a definição das equações do movimento de forma fechada e o projeto do sistema de controle utilizando-se de um controlador Proporcional-Derivativo descentralizado. Ainda, todas as etapas de análise – modelagem cinemática, modelagem dinâmica e controle – foram simuladas para uma determinada trajetória. Conclui-se que é possível encontrar as equações dinâmicas do movimento de forma fechada do manipulador por meio do método do trabalho virtual, bem como que o desempenho do controlador projetado se mostrou satisfatório para a aplicação.

Palavras-chave: Plataforma de Stewart-Gough. Manipuladores em configuração paralela 6-SPS. Modelagem dinâmica pelo princípio do trabalho virtual.

ABSTRACT

DYNAMIC FORMULATION BY MEANS OF THE VIRTUAL WORK PRINCIPLE AND DESCENTRALIZED PD CONTROL OF A STEWART-GOUGH PLATFORM

AUTHOR: DAVI KLEIN

ADVISOR: DANIEL FERNANDO TELLO GAMARRA

This work presents the dynamics modeling through the virtual work principle, as well as the Proportional-Derivative motion control of a parallel robotic manipulator known as Stewart-Gough platform. Also, this work contains the simulation of kinematic, dynamic and control system equations. To perform these simulations, algorithms were developed and implemented using Matlab®. This work obtained as results the dynamics modeling of the system using the virtual work principle, the definition of the closed-form equations of motion and the design of the control system using a decentralized Proportional-Derivative controller. All the steps of the analysis– kinematic modeling, dynamic modeling and control – were simulated for a certain trajectory. It is concluded that it is possible to find the closed-form dynamics equations of motion of the manipulator by means of the virtual work principle, as well as that the performance of the designed controller proved to be suitable for the application.

Key words: Stewart-Gough platform. 6-SPS parallel configuration manipulators. Dynamic modeling by means of the virtual work principle.

LISTA DE ILUSTRAÇÕES

Figura 1 - Configuração genérica do mecanismo elaborado por D. Stewart.....	12
Figura 2 - Plataforma de Stewart-Gough.....	14
Figura 3 - Robô manipulador em configuração série	18
Figura 4 - Robô Paralelo Genérico	18
Figura 5 – Esquemático de uma plataforma de Stewart	20
Figura 6 - Descrição e geometria da plataforma de Stewart-Gough	23
Figura 7 - Rotações por ângulos de Euler no sistema de referência $\{i\}$	40
Figura 8 - Diagrama de corpo livre dos elos	42
Figura 9 - Topologia do sistema de controle proposto	52
Figura 10 - Fluxograma para programa principal.....	55
Figura 11 - Distâncias e_1 e e_2	57
Figura 12 - Algoritmo para cálculo da cinemática inversa.....	59
Figura 13 - Algoritmo para simulação da cinemática direta da plataforma de Stewart	60
Figura 14 - Translação no eixo x da trajetória de referência	64
Figura 15 - Translação no eixo y da trajetória de referência	65
Figura 16 - Translação no eixo z da trajetória de referência	65
Figura 17 - Rotação no eixo x da trajetória de referência	66
Figura 18 - Rotação no eixo y da trajetória de referência	66
Figura 19 - Rotação no eixo z da trajetória de referência.....	67
Figura 20 - Resultado da simulação da cinemática inversa.....	68
Figura 21 - Resultado da simulação da dinâmica inversa do manipulador	70
Figura 22 – Resultado da replicação da primeira simulação de Tsai	71
Figura 23 - Resultado da primeira simulação realizada por Tsai	71
Figura 24 - Resultado da replicação da segunda simulação de Tsai	72
Figura 25 - Resultado da segunda simulação realizada por Tsai.....	72
Figura 26 - Resultado para translação em x	74
Figura 27 - Resultado para translação em y	74
Figura 28- Resultado para translação em y	75
Figura 29 - Resultado para rotação em x	75
Figura 30 - Resultado para rotação em y	76
Figura 31 - Resultado para rotação em z	76

LISTA DE TABELAS

Tabela 1 - Comparação entre manipuladores quanto ao mecanismo	20
Tabela 2 - Comparação entre manipuladores quanto ao controle	21
Tabela 3 - Erro de rastreo em regime permanente	77

SUMÁRIO

1	INTRODUÇÃO.....	12
1.1	DESCRIÇÃO DO PROBLEMA	14
1.2	OBJETIVO GERAL	15
1.3	OBJETIVOS ESPECÍFICOS	15
2	REVISÃO DE LITERATURA	17
2.1	CLASSIFICAÇÃO DE MANIPULADORES ROBÓTICOS.....	17
2.1.1	Manipuladores robóticos em configuração série	17
2.1.2	Manipuladores robóticos em configuração paralela	18
2.2	PLATAFORMA DE STEWART-GOUGH	19
2.3	ROBÔS MANIPULADORES SERIAIS VS. PARALELOS	20
2.4	ANÁLISE CINEMÁTICA DE MANIPULADORES ROBÓTICOS.....	21
2.5	ANÁLISE DA CINEMÁTICA INVERSA DE UMA PLATAFORMA DE STEWART-GOUGH.....	22
2.5.1	Descrição do mecanismo	22
2.5.2	Geometria do manipulador.....	24
2.5.3	Cinemática Inversa	25
2.6	ANÁLISE DA CINEMÁTICA DIRETA DE UMA PLATAFORMA DE STEWART-GOUGH.....	26
2.6.1	Cinemática direta obtida por meio de método numérico computacional	27
2.7	ANÁLISE CINEMÁTICA DIFERENCIAL: MATRIZ JACOBIANA.....	28
2.7.1	Matriz Jacobiana da plataforma de Stewart-Gough.....	28
2.8	ANÁLISE DINÂMICA DE MANIPULADORES ROBÓTICOS.....	31
2.9	FORMULAÇÃO DINÂMICA DE ROBÔS PARALELOS POR MEIO DO MÉTODO DO TRABALHO VIRTUAL	33
2.9.1	Princípio de d'Alembert.....	33
2.9.2	Princípio do Trabalho Virtual.....	34
2.10	FORMULAÇÃO DINÂMICA DA PLATAFORMA DE STEWART-GOUGH POR MEIO DO MÉTODO DO TRABALHO VIRTUAL	39
2.10.1	Análise de posição	39
2.10.2	Análise de velocidade.....	42
2.10.3	Matrizes Jacobianas dos elos	43
2.10.4	Análise de aceleração.....	45
2.10.5	Formulação dinâmica implícita.....	46
2.10.6	Formulação dinâmica explícita.....	47
2.11	SISTEMAS DE CONTROLE PARA ROBÔS PARALELOS	49
2.12	CONTROLE DE ROBÔS PARALELOS	50
2.12.1	Controle Proporcional-Derivativo (PD) descentralizado da plataforma de Stewart-Gough no espaço de tarefa	52
3	MATERIAIS E MÉTODOS	55
3.1	ALGORITMO PRINCIPAL E PARÂMETROS FÍSICOS DO MANIPULADOR	55
3.2	ALGORITMO PARA SIMULAÇÃO DA CINEMÁTICA INVERSA.....	58
3.3	ALGORITMO PARA SIMULAÇÃO DA CINEMÁTICA DIRETA	59
3.4	ALGORITMO PARA SIMULAÇÃO DA DINÂMICA INVERSA	60
3.5	ALGORITMO PARA SIMULAÇÃO DO CONTROLE PD DESCENTRALIZADO ...	62
4	APRESENTAÇÃO E DISCUSSÃO DOS RESULTADOS	64

4.1	SIMULAÇÃO DA CINEMÁTICA INVERSA DO MANIPULADOR.....	64
4.2	SIMULAÇÃO DA CINEMÁTICA DIRETA DO MANIPULADOR.....	68
4.3	SIMULAÇÃO DA DINÂMICA INVERSA DO MANIPULADOR.....	69
4.4	SIMULAÇÃO DO CONTROLE DO MANIPULADOR	73
5	CONCLUSÃO.....	78
	REFERÊNCIAS.....	79
	ANEXO A – CÓDIGOS	81
A.1	CÓDIGO PRINCIPAL	81
A.2	CÁLCULO DA CINEMÁTICA INVERSA	90
A.3	CÁLCULO DA CINEMÁTICA DIRETA.....	91
A.4	CÓDIGO QUE CALCULA A DINÂMICA INVERSA.....	92
A.5	CÓDIGO PRINCIPAL DO CONTROLE.....	95
A.6	CÓDIGO QUE CALCULA A AÇÃO DE CONTROLE.....	96

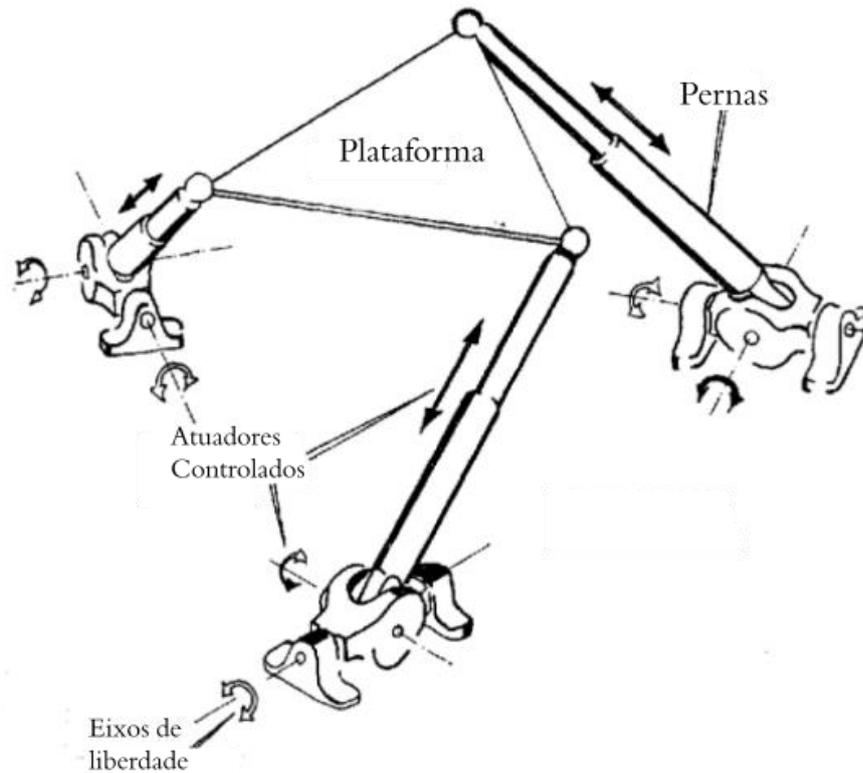
1 INTRODUÇÃO

Desde o início da humanidade, o Homem busca aumentar sua qualidade de vida com a criação de novas tecnologias. Com a primeira Revolução Industrial, essas tecnologias se tornaram mais presentes na vida do ser humano. Ao passo que a tecnologia evoluiu, novos inventos – como mecanismos robóticos – surgiram para facilitar a vida das pessoas. Com o surgimento desses mecanismos, foi criado um conceito para esclarecer a Robótica. No seu livro *Introdução à Robótica*, Niku (2019, p. 4) define a Robótica da seguinte forma: “Robótica é a arte, a base de conhecimento e o *know-how* de concepção, aplicação e uso de robôs em atividades humanas[...]”. O contato do ser humano com esses tipos de máquinas se dá, de forma geral, em dois ambientes: doméstico e industrial. Na indústria, são muito comuns manipuladores robóticos. Esses manipuladores podem ser divididos em duas categorias: manipuladores em configuração série e paralela. Neste trabalho, será abordado um tipo específico de manipulador robótico em configuração paralela: uma plataforma de Stewart-Gough.

Elaborada por D. Stewart em 1965, a plataforma de Stewart surge com a finalidade de “[...]formar um *design* elegante para simulação de condições de voo no treinamento de pilotos [...]” (STEWART, 1965, p. 371). Em seu artigo, Stewart descreve uma base triangular acoplada a uma plataforma igualmente triangular por meio de atuadores lineares presos a juntas com dois graus de liberdade. Além disso, Stewart realiza a análise cinemática e dinâmica da plataforma proposta, bem como discorre sobre as limitações de amplitude, meios de controle e considerações do projeto mecânico da plataforma. Os conceitos propostos por Stewart em seu artigo influenciaram de forma direta o planejamento e execução do presente trabalho. Todavia, apesar de Stewart ter proposto esse tipo de mecanismo em 1965, Gough (1957), já havia publicado um trabalho detalhando um mecanismo parecido com o de Stewart em seu artigo “*Contribution to discussion of papers on research in Automobile Stability, Control and Tyre performance*”. Nesse artigo, Gough discorre sobre a aplicação de um mecanismo robótico paralelo para a determinação das propriedades de pneus sob efeito de cargas combinadas. Assim, a plataforma de Stewart passa a ser chamada de plataforma de Stewart-Gough devido à contribuição prévia de Eric Gough a esse mecanismo. A partir dos trabalhos de Stewart e Gough, estudos mais aprofundados sobre esse tipo de mecanismo surgem. Dentre esses, há a análise dos graus de liberdade efetivos do robô, sua modelagem cinemática, dinâmica e controle. A **Erro! Autoreferência de indicador não válida.** descreve o mecanismo proposto

por Stewart – com duas bases triangulares e três elos acoplados por juntas prismáticas (atuadas) e juntas esféricas (passivas).

Figura 1 - Configuração genérica do mecanismo elaborado por D. Stewart



Fonte: (STEWART, 1965, p.372, traduzido pelo autor).

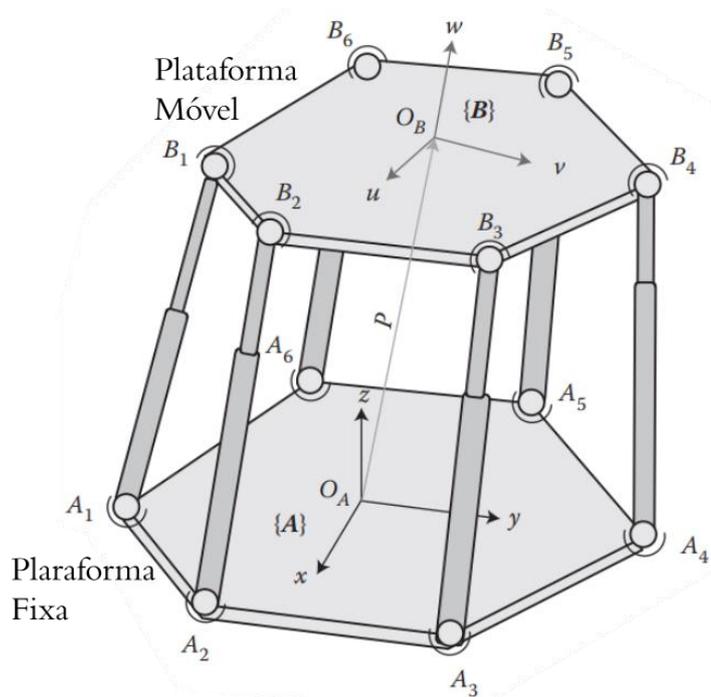
Apesar de ser criada para contemplar os objetivos propostos por Stewart e Gough, a plataforma de Stewart-Gough pode ser aplicada em diversas situações. Em ambientes industriais, esse tipo de mecanismo possui uma ampla vantagem em relação a robôs seriais em aplicações que necessitam manipulações precisas de altas cargas. Além disso, há aplicações em outras áreas, como, por exemplo, a aplicação da plataforma no posicionamento de rádio telescópios por Su e Duan (2012) em seu artigo “*Design genético de uma plataforma de Stewart cinematicamente ótima de ajuste fino para grandes radio telescópios esféricos*”.

Através desse projeto, deseja-se alcançar a modelagem cinemática e dinâmica de uma plataforma de Stewart-Gough, bem como projetar um controlador Proporcional-Derivativo para realizar seu controle de movimento. Ao decorrer desse trabalho, será discorrido sobre as metodologias utilizadas para a dedução e simulação das equações que compõe a cinemática, dinâmica e controle da plataforma, através da aplicação dessas equações a trajetórias a serem executadas pelo manipulador.

1.1 DESCRIÇÃO DO PROBLEMA

Uma plataforma de Stewart-Gough consiste de duas plataformas – sendo uma fixa e outra móvel – ligadas por cadeias cinemáticas fechadas. De acordo com H. Taghirad (2013, p.4) “Uma cadeia cinemática é a construção de elos conectados por juntas. Quando cada elo em uma cadeia cinemática é conectado a outros elos por pelo menos dois caminhos diferentes, essa é chamada de uma cadeia cinemática fechada”. Um exemplo desse manipulador se encontra na Figura 2. As plataformas podem possuir diversas geometrias: triangular, hexagonal regular, hexagonal irregular, circular, dentre outras. As cadeias cinemáticas fechadas são constituídas por atuadores lineares, esses podendo ser pistões hidráulicos, pistões pneumáticos ou atuadores lineares elétricos.

Figura 2 - Plataforma de Stewart-Gough



Fonte: (TAGHIRAD, 2013, p.82, traduzido pelo autor).

Nesse sistema, os atuadores possuem ambas extremidades ligadas a duas juntas esféricas que, por sua vez, estão conectadas às plataformas fixa e móvel. A ligação entre o atuador e a

plataforma fixa se dá por uma junta esférica (S) de três graus de liberdade. O atuador linear possui uma parte fixa - geralmente um cilindro - e uma parte móvel - geralmente um êmbolo. Essa ligação cilindro-êmbolo representa uma junta prismática (P), de um grau de liberdade. Por sua vez, a outra extremidade do atuador se conecta com a plataforma móvel superior por meio de outra junta esférica (S) de três graus de liberdade, constituindo, assim, uma cadeia cinemática fechada. Assim, pode-se definir a plataforma de Stewart-Gough como um manipulador paralelo de seis graus de liberdade com estrutura cinemática 6-SPS. Esse nome 6-SPS se dá exatamente pelo fato de esse tipo de manipulador possuir seis cadeias cinemáticas fechadas (6) constituídas de uma junta esférica (S) seguida de uma junta prismática (P) e, por fim, conectada a uma outra junta esférica (S).

A principal problemática para a análise da plataforma de Stewart-Gough envolve a sua modelagem dinâmica. Essa dificuldade encontra-se na metodologia a ser utilizada em sua modelagem dinâmica. Os métodos de Newton-Euler e Mecânica de Lagrange são implementados, em sua maioria, para a modelagem dinâmica deste manipulador paralelo. Todavia, essas metodologias possuem desempenho computacional baixo. Assim, para aplicações de simulação, deve-se buscar uma metodologia que possua custo computacional baixo o suficiente a fim de se otimizar a simulação. Ainda, um grande problema para a síntese de uma Plataforma de Stewart-Gough é o projeto e execução de seu sistema de controle. Devido ao fato do sistema que representa este manipulador robótico ser altamente não linear, técnicas de controle avançadas precisam, normalmente, ser implementadas para se obter um desempenho ótimo. Após o exposto, surge o seguinte questionamento: utilizando a metodologia de modelagem ótima para simulações, é possível projetar um sistema de controle com desempenho aceitável e com redução significativa na dificuldade de projeto e execução?

1.2 OBJETIVO GERAL

O objetivo geral desse trabalho é realizar a modelagem dinâmica por meio do método do trabalho virtual, controle PD descentralizado e simulação de uma plataforma de Stewart-Gough.

1.3 OBJETIVOS ESPECÍFICOS

- Realizar uma revisão bibliográfica sobre a estrutura mecânica, funcionamento, modelagem cinemática, modelagem dinâmica e controle do sistema;

- Obter os modelos cinemático e dinâmico do manipulador por meio do método do trabalho virtual;
- Realizar o projeto do sistema de controle PD descentralizado;
- Simular os modelos cinemático e dinâmico, bem como o controlador;
- Avaliar os resultados obtidos a partir da simulação, comparando-os com a bibliografia.

2 REVISÃO DE LITERATURA

Este Capítulo é destinado para a explicação da teoria que envolve a modelagem cinemática, dinâmica e controle de uma plataforma de Stewart-Gough. Ainda nesse Capítulo haverá a explicação de conhecimentos relativos ao campo de estudo da Robótica necessários para o desenvolvimento do projeto.

2.1 CLASSIFICAÇÃO DE MANIPULADORES ROBÓTICOS

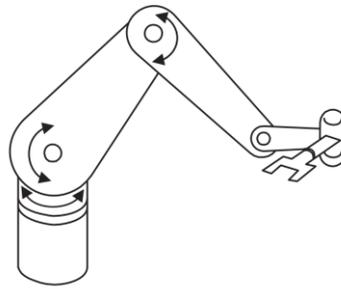
De acordo com Tsai (1999), robôs podem ser classificados de acordo com vários critérios, como: graus de liberdade, estrutura cinemática, tecnologia de atuação, geometria do espaço de trabalho e características de movimento. Para esse trabalho é importante mencionar e detalhar a diferenciação de robôs por sua estrutura cinemática - com a finalidade de entender vantagens e desvantagens do robô manipulador proposto.

Robôs podem ser classificados quanto sua estrutura cinemática como: robôs em configuração série (ou robôs de cadeia cinemática aberta), robôs em configuração paralela (ou robôs de cadeia cinemática fechada) e robôs híbridos.

2.1.1 Manipuladores robóticos em configuração série

Conforme Tsai (1999), um robô é classificado como robô em configuração série se a sua estrutura cinemática assume a forma de uma cadeia cinemática aberta. Esse tipo de robô é amplamente utilizado em ambientes industriais devido ao seu grande espaço de trabalho e mobilidade.

Figura 3 - Robô manipulador em configuração série

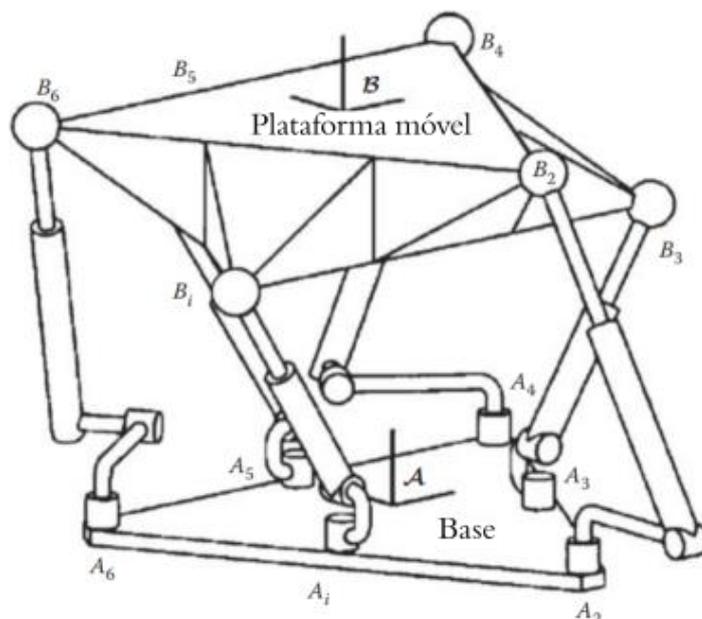


Fonte: (NIKU, 2010, p.11).

2.1.2 Manipuladores robóticos em configuração paralela

De acordo com J.P. Merlet (2006, p.12) “Um manipulador paralelo generalizado é um mecanismo de cadeia cinemática fechada cujo operador terminal é ligado a base por várias cadeias cinemáticas independentes”. Em geral, um robô paralelo é formado por uma base e uma plataforma móvel, sendo essas ligadas uma à outra por meio de elos e juntas. Esses elos podem ser atuados por atuadores lineares ou rotativos e ligados às plataformas por um conjunto de juntas esféricas, prismáticas ou rotativas, de forma geral. A Figura 4 mostra um exemplo de robô paralelo genérico.

Figura 4 - Robô Paralelo Genérico



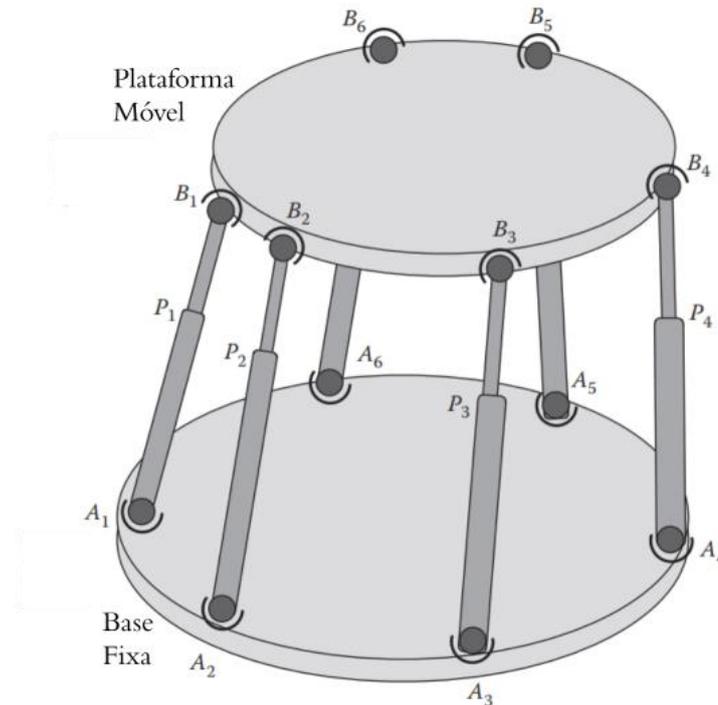
Fonte: (TAGHIRAD, 2013, p.14, traduzido pelo autor).

Os manipuladores paralelos podem ser diferenciados quanto a relação entre seus graus de liberdade e atuadores. Caso um manipulador paralelo possua a mesma quantidade de atuadores de que o número de graus de liberdade ativos, esse manipulador é dito estritamente paralelo ou completamente paralelo. Caso um manipulador paralelo possua maior quantidade de atuadores de que o número de graus de liberdade ativos, esse manipulador é dito redundante. De acordo com H. Taghirad (2013, p.10) “Robôs manipuladores completamente paralelos possuem uma estrutura eficiente em relação ao custo, todavia essa estrutura limita a manipulabilidade do manipulador para executar a tarefa requerida”. Os dois tipos mais comuns de manipuladores paralelos serão explicitados nas seções a seguir.

2.2 PLATAFORMA DE STEWART-GOUGH

A plataforma de Stewart-Gough é um robô estritamente paralelo de seis graus de liberdade movido por meio de atuadores lineares utilizando-se de uma estrutura cinemática 6-SPS. Essa estrutura cinemática é formada por seis cadeias cinemáticas fechadas compostas de um elo preso à base por uma junta esférica (S) e ligado a um segundo elo por meio de uma junta binária prismática (P). Por sua vez, esse segundo elo é fixado na plataforma móvel por meio de uma junta esférica (S). A Figura 5 mostra uma plataforma de Stewart-Gough. Existem algumas variações de estrutura cinemática da plataforma de Stewart-Gough, como, por exemplo, a estrutura cinemática 6-RSS. Essa estrutura possui uma junta rotativa (R) – geralmente um motor - ligada a um elo com duas juntas esféricas (S) presas em suas extremidades.

Figura 5 – Esquemático de uma plataforma de Stewart



Fonte: (TAGHIRAD, 2013, p.9, traduzido pelo autor).

2.3 ROBÔS MANIPULADORES SERIAIS VS. PARALELOS

A partir do exposto nas seções anteriores sobre a classificação de robôs manipuladores quanto a sua estrutura cinemática, pode-se realizar uma análise e uma comparação quanto às vantagens, desvantagens e aplicações desses dois tipos de manipuladores. Na Tabela 1 pode se visualizar uma comparação entre esses tipos de manipuladores quanto ao mecanismo e na Tabela 2 uma comparação quanto ao controle.

Tabela 1 - Comparação entre manipuladores quanto ao mecanismo

Mecanismo	Manipulador série	Manipulador paralelo
Inércia	Grande	Pequeno
Volume de trabalho	Grande	Pequeno
Aparência	Forma antropomórfica	Base estrutura

Fonte: MORETTI (2010).

Tabela 2 - Comparação entre manipuladores quanto ao controle

Controle	Manipulador série	Manipulador paralelo
Controle de posição	Difícil	Fácil
Controle de força	Fácil	Difícil
Deteção de força	Difícil	Fácil
Erro de posição	Acumulado	Médio
Erro de controle de força	Médio	Acumulado
Perto de pontos singulares	Degeneração em controle de força	Diminuição de exatidão
Dinâmica	Complicada	Muito complicada

Fonte: MORETTI (2010).

Enquanto robôs em configuração série possuem pouca rigidez estrutural devido aos momentos gerados na estrutura do robô pela manipulação de carga em seu elemento terminal, robôs paralelos possuem uma rigidez estrutural maior por possuírem elos que sustentam a base móvel. Isso faz com que robôs paralelos sejam mais adequados para aplicações que necessitam uma capacidade de carga maior. Da mesma maneira, por possuírem cadeias cinemáticas abertas, os robôs em configuração série possuem um espaço de trabalho enorme em comparação com robôs paralelos. Isso se dá pelo fato de que robôs paralelos possuem elos que, em determinados pontos de singularidade, podem se colidir ou apresentar baixa rigidez estrutural. Além disso, essa característica pode fazer com que o robô apresente graus de liberdade extras temporários – os quais são incontrolláveis -, fazendo com que o robô perca sua rigidez.

2.4 ANÁLISE CINEMÁTICA DE MANIPULADORES ROBÓTICOS

A cinemática é uma subdivisão da mecânica na qual se realiza o estudo do movimento de corpos sem se preocupar com a origem desses movimentos.

Pode se dividir a análise cinemática de manipuladores robóticos em análise cinemática direta e análise cinemática inversa. A cinemática direta de manipuladores se preocupa em descobrir a posição e orientação finais do operador terminal do robô se baseando na posição e orientação das juntas e elos do mesmo. De acordo com J. Craig (2014, p.5):

O problema da cinemática direta de manipuladores pode ser descrito como computar a posição e orientação do sistema de referência da ferramenta em relação ao sistema de referência base, dado um conjunto de ângulos de juntas.

Em termos gerais, a solução para a cinemática direta de manipuladores seriais é um tanto quanto intuitiva devido à geometria do mesmo. Todavia, para manipuladores paralelos, essa modelagem pode ser muito complexa. Isso se dá pelo fato de que, para manipuladores paralelos, a cinemática inversa resulta em um sistema de equações não lineares. É possível solucionar esse problema de forma analítica utilizando-se de técnicas algébricas ou de forma iterativa computacional. Mesmo assim, o custo computacional dessas soluções é, geralmente, muito alto. Isso faz com que o emprego de manipuladores robóticos paralelos, como a plataforma de Stewart-Gough, tenha aplicações industriais limitadas.

De maneira complementar à cinemática direta de manipuladores, há, também, a cinemática inversa de manipuladores. Esse tópico estuda exatamente o contrário da cinemática direta: se preocupa em descobrir a posição e orientação finais dos ângulos das juntas se baseando na posição e orientação do operador terminal do robô.

Para robôs seriais, o problema da cinemática inversa pode ser muito complexo pela não linearidade das equações obtidas. Entretanto, para robôs paralelos, a cinemática inversa se mostra simples devido à sua geometria.

2.5 ANÁLISE DA CINEMÁTICA INVERSA DE UMA PLATAFORMA DE STEWART-GOUGH

Nesta seção será realizada a dedução das equações que compõe a cinemática inversa de uma plataforma de Stewart-Gough utilizando-se uma abordagem geométrica vetorial. Primeiramente será descrita a estrutura cinemática do manipulador. Após isso a geometria do manipulador será explicitada. Finalmente, será exposta a cinemática inversa desse manipulador.

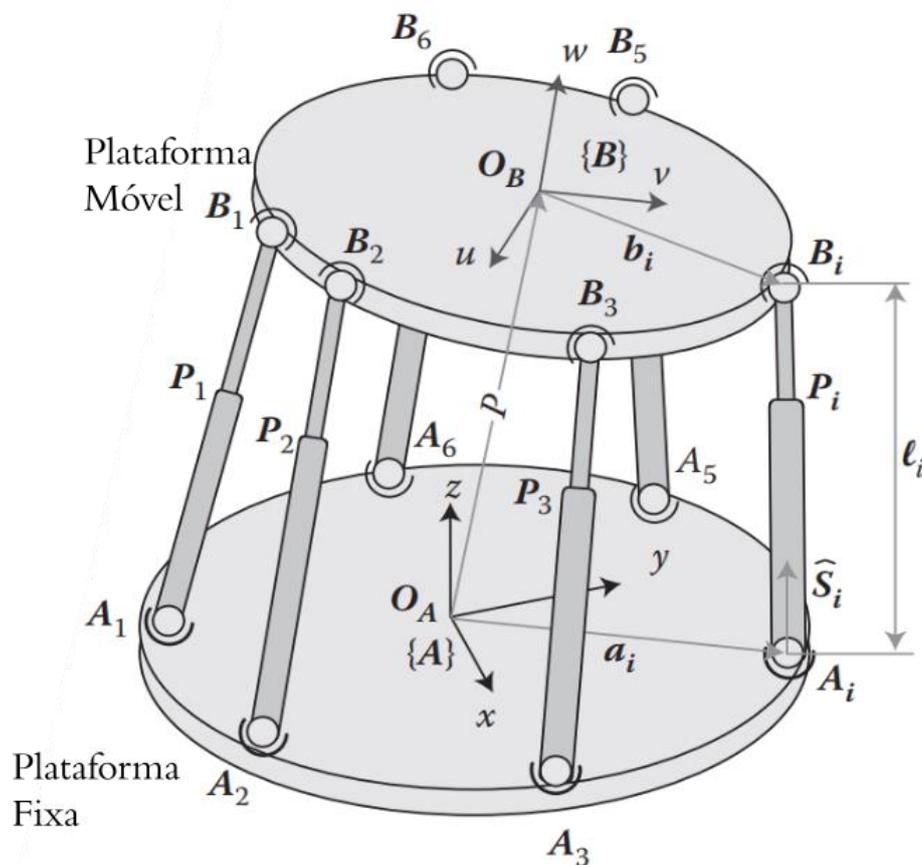
2.5.1 Descrição do mecanismo

A partir da Figura 6, pode-se visualizar a descrição do mecanismo e geometria da plataforma de Stewart-Gough. Conforme exposto na seção 1.1, esse manipulador é composto de seis atuadores lineares cujo movimento é regido por uma junta prismática P_i . Esses atuadores conectam a plataforma fixa com a móvel por meio de juntas esféricas em suas

extremidades. Percebe-se que a plataforma fixa possui um sistema de referência fixo $\{A\}$ afixado em seu ponto central. Esse sistema de referência possui eixos x , y e z , além de sua origem ser identificada por O_A . A plataforma fixa possui seis juntas rotativas identificadas por A_i onde i varia de um a seis. De maneira análoga, a plataforma móvel possui um sistema de referência móvel $\{B\}$ afixado em seu ponto central. Esse sistema de referência possui eixos u , v e w , além de ter sua origem identificada por O_B . As juntas rotativas da plataforma móvel são seis e são identificadas por B_i onde i varia de um a seis.

Como descrito anteriormente, o mecanismo proposto apresenta seis graus de liberdade. Portanto, esse robô pode ser descrito como um manipulador paralelo generalizado com três parâmetros de posição e três parâmetros de orientação. Como há seis atuadores – um para cada grau de liberdade –, esse manipulador é dito completamente paralelo.

Figura 6 - Descrição e geometria da plataforma de Stewart-Gough



Fonte: (TAGHIRAD, 2013, p.77, traduzido pelo autor).

2.5.2 Geometria do manipulador

Observa-se, na Figura 6, que, na plataforma fixa, as juntas representadas por A_i possuem um vetor posição ${}^A\vec{a}_i$ - o qual define a posição das juntas em relação ao sistema de referência fixo {A}. De forma análoga, na plataforma móvel, as juntas representadas por B_i possuem um vetor posição ${}^B\vec{b}_i$ - o qual define a posição das juntas em relação ao sistema de referência móvel {B}. Em relação aos elos, cada um é definido por um comprimento escalar l_i e um vetor direcional unitário ${}^A\hat{s}_i$. Ainda, a posição do sistema de referência {B} em relação ao sistema de referência {A} é dada por um vetor posição ${}^A\vec{P}$ que possui sua origem em O_A e sua extremidade em O_B . Esse vetor posição está referenciado em relação ao sistema de referência fixo {A}. Além disso, a orientação do sistema de referência {B} pode ser definida por diversas representações de orientação de corpos rígidos – como matrizes de rotação, ângulos de Euler e por eixo helicoidal. Neste trabalho, essa representação será feita utilizando-se de ângulos de Euler Z-Y-X. Assim, pode-se definir a matriz de rotação que referencia a orientação do sistema de referência {B} para o sistema de referência {A} de acordo com a Equação (1):

$${}^A_R_B{}^{Z'Y'X'}(\phi, \psi, \theta) = \begin{bmatrix} c\phi c\psi & c\phi s\psi s\theta - s\phi c\theta & c\phi s\psi c\theta + s\phi s\theta \\ s\phi c\psi & s\phi s\psi s\theta + c\phi c\theta & s\phi s\psi c\theta - c\phi s\theta \\ -s\psi & c\psi s\theta & c\psi c\theta \end{bmatrix} \quad (1)$$

Onde:

- ϕ é o ângulo de rotação em torno do eixo Z;
- ψ é o ângulo de rotação em torno do eixo Y;
- θ é o ângulo de rotação em torno do eixo X;
- c é a abreviatura de cosseno;
- s é a abreviatura de seno.

É notável que diferentes geometrias de plataformas podem ser utilizadas, bem como diferentes posições para juntas. De qualquer forma, isso não altera a dedução das equações da cinemática inversa percorridas nesse projeto.

2.5.3 Cinemática Inversa

Para resolver o problema da cinemática inversa desse manipulador, assume-se que a orientação e a posição da plataforma são dadas e deseja-se encontrar o comprimento de cada um dos atuadores lineares. A partir da Figura 6, realiza-se a soma vetorial que representa a cadeia cinemática fechada do sistema, da seguinte maneira:

$$l_i \cdot {}^A\hat{s}_i + {}^A\vec{a}_i = {}^A\vec{P} + {}^A\vec{b}_i \quad (2)$$

Ainda, percebe-se que:

$${}^A\vec{b}_i = {}^A R_{Z'Y'X'} \cdot {}^B\vec{b}_i \quad (3)$$

Substituindo a Equação (3) na Equação (2) e isolando o termo $l_i \cdot {}^A\hat{s}_i$ temos:

$$l_i \cdot {}^A\hat{s}_i = {}^A\vec{P} + {}^A R_{Z'Y'X'} \cdot {}^B\vec{b}_i - {}^A\vec{a}_i \quad (4)$$

Agora, deve-se isolar completamente o termo l_i , eliminando o termo ${}^A\hat{s}_i$. Para isso, realiza-se a multiplicação escalar de cada lado da equação por sua transposta, da seguinte maneira:

$$l_i^2 \cdot [{}^A\hat{s}_i^T \cdot {}^A\hat{s}_i] = [{}^A\vec{P} + {}^A R_{Z'Y'X'} \cdot {}^B\vec{b}_i - {}^A\vec{a}_i]^T \cdot [{}^A\vec{P} + {}^A R_{Z'Y'X'} \cdot {}^B\vec{b}_i - {}^A\vec{a}_i] \quad (5)$$

É notável que o produto escalar entre um vetor unitário e sua transposição é igual a um, assim:

$${}^A\hat{s}_i^T \cdot {}^A\hat{s}_i = 1 \quad (6)$$

Aplicando-se a Equação (6) na Equação (5) e realizando-se algumas operações algébricas, obtém-se:

$$l_i = \left({}^A\bar{P}^T \cdot {}^A\bar{P} + {}^B\bar{b}_i^T \cdot {}^B\bar{b} + {}^A\bar{a}_i^T \cdot {}^A\bar{a}_i - 2 \cdot \left({}^A\bar{P}^T \cdot {}^A\bar{a}_i \right) + 2 \cdot \left({}^A\bar{P}^T \cdot \left[{}^A R_{B Z' Y' X'} \cdot {}^B\bar{b}_i \right] \right) - \right. \\ \left. 2 \cdot \left(\left[{}^A R_{B Z' Y' X'} \cdot {}^B\bar{b}_i \right]^T \cdot {}^A\bar{a}_i \right) \right)^{\frac{1}{2}} \quad (7)$$

A Equação (7) descreve a cinemática inversa da plataforma de Stewart-Gough. Espera-se um único resultado para essa equação caso a orientação e posição da plataforma móvel dada estejam contidas no espaço de trabalho do manipulador. Ainda, nota-se que, se a solução dessa equação for um número complexo, não há possibilidade de o robô alcançar a orientação e posição desejadas.

2.6 ANÁLISE DA CINEMÁTICA DIRETA DE UMA PLATAFORMA DE STEWART-GOUGH

De acordo com o conceito explicitado anteriormente, na análise cinemática direta desse manipulador é considerado que os comprimentos dos atuadores lineares são conhecidos e deseja-se obter a posição e orientação da plataforma móvel. Devido ao fato de que foi escolhida a representação do corpo rígido se utilizando os ângulos de Euler com convenção ZYX, esse problema pode ser reduzido em um sistema não linear de seis equações com seis variáveis desconhecidas – três variáveis relacionadas ao vetor posição e nove variáveis relacionadas com os três ângulos de Euler. Ainda, o número dessas equações pode ser aumentado a sete, quando utilizada a representação por eixo helicoidal, ou aumentado a doze, quando utilizada a representação por uma matriz de rotação.

Como mencionado anteriormente, a análise cinemática direta de manipuladores paralelos é um grande desafio devido a não-linearidade extrema das equações que compõe o sistema. Diversos métodos algébricos e computacionais foram desenvolvidos para solucionar o problema da cinemática inversa da plataforma de Stewart-Gough – desde os métodos de eliminação algébrica e análise de intervalos até a utilização de redes neurais artificiais. Mesmo soluções numéricas mostram-se ineficazes devido ao fato de que é necessário realizar uma estimativa inicial muito precisa para que haja convergência para a solução, sem mencionar o extremo custo computacional envolvido no processo. O fato é que, não importando o método de solução utilizado, encontrar uma única solução para a posição e orientação da plataforma

móvel mostra-se, ainda, um desafio enorme. Por esses motivos explicitados é que a plataforma de Stewart-Gough ainda possui limitadas aplicações em tempo real e que requerem alta velocidade de resposta. (TAGHIRAD, 2013, p. 81)

A seguir, será demonstrado o cálculo da cinemática direta da plataforma de Stewart-Gough por meio de um método numérico. Essa análise será demonstrada nesse trabalho apenas como objeto de estudo e confirmação de todas as limitações e dificuldades de aplicação.

2.6.1 Cinemática direta obtida por meio de método numérico computacional

Para solucionar a cinemática direta da plataforma de Stewart-Gough, toma-se como suposições o que foi explicitado anteriormente: considera-se que os comprimentos dos atuadores lineares são conhecidos e se deseja encontrar a posição e orientação da plataforma móvel. Ainda, baseado na descrição da geometria do manipulador, os vetores ${}^A\bar{b}_i$ e ${}^A\bar{a}_i$ determinam a posição das juntas esféricas das plataformas móvel e fixa, respectivamente. Para solucionar o problema, podemos unir o vetor ${}^A\bar{P}$ e os três ângulos de Euler ϕ , ψ e θ em um vetor x .

$$x = [p_x \quad p_y \quad p_z \quad \phi \quad \psi \quad \theta]^T = [x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5 \quad x_6]^T \quad (8)$$

Assim, pode se definir as variáveis de posição e orientação da seguinte maneira:

$$P(x) = [x_1 \quad x_2 \quad x_3]^T \quad (9)$$

$$R(x) = \begin{bmatrix} cx_4cx_5 & cx_4sx_5sx_6 - sx_4cx_6 & cx_4sx_5cx_6 + sx_4sx_6 \\ sx_4cx_5 & sx_4sx_5sx_6 + cx_4cx_6 & sx_4sx_5cx_6 - cx_4sx_6 \\ -sx_5 & cx_5sx_6 & cx_5cx_6 \end{bmatrix} \quad (10)$$

Por fim, define-se o seguinte sistema de equações não-lineares derivadas da análise da cadeia cinemática fechada do manipulador:

$$\begin{cases} F_i(x) = -l_i + [P(x) + R(x) \cdot {}^A\bar{b}_i - {}^A\bar{a}_i]^T \cdot [P(x) + R(x) \cdot {}^A\bar{b}_i - {}^A\bar{a}_i] \\ i = 1 \dots 6 \end{cases} \quad (11)$$

Finalmente, a partir da Equação (11), é possível achar uma solução para o problema da análise cinemática direta da plataforma de Stewart-Gough. Os métodos numéricos mais comuns usados para resolver esse sistema por meio do método dos mínimos quadrados para sistemas não lineares são o de Gauss-Newton e Levenberg-Marquadt, como será demonstrado na Seção 3.

2.7 ANÁLISE CINEMÁTICA DIFERENCIAL: MATRIZ JACOBIANA

De modo a aprofundar a análise cinemática de manipuladores paralelos, é necessário o estudo de um tópico conhecido como cinemática diferencial. Esse estudo é importante para que se analise a movimentação do robô manipulador – velocidades angulares e lineares - em adendo com a análise puramente posicional realizada nos estudos de cinemáticas direta e inversa. Ainda, a análise da cinemática diferencial nos traz um conceito muito importante no campo de estudo de análise de sistemas de multicorpos rígidos e da Robótica: a matriz Jacobiana. De acordo com Taghirad (2013, p.111):

A matriz Jacobiana não somente revela a relação entre as velocidades das variáveis de junta de um manipulador paralelo com as velocidades angulares e lineares da plataforma móvel, como constrói a transformação necessária para se encontrar as forças dos atuadores a partir das forças e momentos atuantes na plataforma móvel.

Ainda, é a partir da matriz Jacobiana que é possível realizar a análise de espaço de trabalho e singularidades de robôs para que se consiga um projeto otimizado e de máxima *performance*. Portanto, a construção da matriz Jacobiana é de extrema importância para o projeto, uma posterior análise dinâmica e o projeto de um sistema de controle para o manipulador.

2.7.1 Matriz Jacobiana da plataforma de Stewart-Gough

Robôs paralelos possuem, de maneira genérica, um corpo rígido móvel conectado a uma base – ou corpo rígido fixo – por meio de elos constituídos de juntas. Para robôs paralelos ditos estritamente paralelos (caso no qual o número de elos atuados é igual ao de graus de liberdade) há somente uma junta atuada por elo, fazendo com que o restante das juntas seja denominado juntas passivas. A plataforma de Stewart-Gough, como explicitado anteriormente neste trabalho, possui uma plataforma fixa acoplada a uma plataforma móvel por meio de seis elos

constituídos por um sistema de juntas SPS, onde as juntas atuadas são as juntas prismáticas P. Para encontrar a matriz Jacobiana do manipulador, define-se uma variável que representa as juntas atuadas por meio do vetor q e uma variável correspondente à posição e orientação da plataforma móvel por meio do vetor χ . Assim:

$$q = [q_1 \quad q_2 \quad q_3 \quad q_4 \quad q_5 \quad q_6]^T \quad (12)$$

$$\chi = [p_x \quad p_y \quad p_z \quad \phi \quad \psi \quad \theta]^T \quad (13)$$

De acordo com H. Taghirad (2013, p.119) esses dois vetores podem ser relacionados por meio de um sistema de equações algébricas não lineares que representam as restrições cinemáticas impostas pelos elos, da seguinte maneira:

$$f(q, \chi) = \vec{0}_{6 \times 6} \quad (14)$$

Ainda, é necessário derivar a Equação (14) para se obter a relação entre a taxa de variação das variáveis de junta atuadas e as velocidades linear e angular da plataforma móvel. Essa operação é representada da seguinte forma:

$$J_\chi \cdot \frac{d\chi}{dt} = J_q \cdot \frac{dq}{dt} \quad (15)$$

As variáveis J_χ e J_q são denominadas as matrizes jacobianas da plataforma de Stewart-Gough e são definidas da seguinte forma:

$$J_\chi = \frac{\partial f(\chi, q)}{\partial \chi} \quad (16)$$

$$J_q = \frac{\partial f(\chi, q)}{\partial q} \quad (17)$$

Ambas as matrizes Jacobianas possuem, para a plataforma de Stewart-Gough, dimensão de 6x6. Ainda, pode-se definir a matriz Jacobiana geral para a plataforma de Stewart-Gough da seguinte maneira:

$$\frac{dq}{dt} = J \cdot \frac{d\chi}{dt} \quad (18)$$

A Equação (18) pode ser somente usada em casos que a matriz Jacobiana das variáveis de junta atuadas J_q seja singular. Percebe-se, também, que a definição de matriz Jacobiana generalizada exposta na Equação (18) é oposta à definição de matriz Jacobiana generalizada para robôs manipuladores seriais.

Com a finalidade de se encontrar a matriz Jacobiana específica para a plataforma de Stewart-Gough, deve se realizar uma análise de uma malha fechada de velocidade. Para isso, toma-se como base a Equação (4), realizando a sua derivada em relação ao tempo. Assim:

$${}^A v_p + \frac{d {}^A R_B}{dt} \bullet {}^B b_i + {}^A R_B \bullet \frac{d {}^B b_i}{dt} - \frac{d {}^A a_i}{dt} = \frac{dl_i}{dt} \cdot {}^A \hat{s}_i + l_i \cdot \frac{d {}^A \hat{s}_i}{dt} \quad (19)$$

Onde ${}^A v_p$ é a velocidade linear da plataforma móvel em relação ao sistema de referência {A} e t é o tempo. Realizando algumas operações algébricas, obtém-se a seguinte equação:

$${}^A \hat{s}_i \bullet {}^A v_p + ({}^A b_i \times {}^A \hat{s}_i) \bullet \omega = \frac{dl_i}{dt} \quad (20)$$

Onde ω é a velocidade angular da plataforma móvel em relação ao sistema de referência {A}. Por fim, realizando-se as substituições de i pelos números de 1 a 6 e agrupando essas equações em um sistema, é possível obter a seguinte notação matricial:

$$\frac{dL}{dt} = J \cdot \frac{dX}{dt} \quad (21)$$

Onde a matriz $\frac{dL}{dt}$ é a matriz que armazena os valores das derivadas dos comprimentos de cada um dos atuadores, $\frac{dX}{dt}$ é a matriz que armazena os valores de velocidade linear e angular da plataforma móvel em relação ao sistema de referência {A} e J é a matriz Jacobiana da plataforma de Stewart-Gough, podendo ser escrita como:

$$J = \begin{bmatrix} {}^A\hat{s}_1^T & ({}^A b_1 \times {}^A\hat{s}_1)^T \\ {}^A\hat{s}_2^T & ({}^A b_2 \times {}^A\hat{s}_2)^T \\ {}^A\hat{s}_3^T & ({}^A b_3 \times {}^A\hat{s}_3)^T \\ {}^A\hat{s}_4^T & ({}^A b_4 \times {}^A\hat{s}_4)^T \\ {}^A\hat{s}_5^T & ({}^A b_5 \times {}^A\hat{s}_5)^T \\ {}^A\hat{s}_6^T & ({}^A b_6 \times {}^A\hat{s}_6)^T \end{bmatrix} \quad (22)$$

A Equação (22) demonstra explicitamente a matriz Jacobiana específica para a plataforma de Stewart-Gough. A partir dela, é possível dar continuidade nos estudos de análise dinâmica, bem como realizar estudos de forças estáticas e análise de singularidades – etapa essa essencial para melhor otimização do projeto desse tipo de manipulador.

2.8 ANÁLISE DINÂMICA DE MANIPULADORES ROBÓTICOS

Ao contrário da cinemática, parte da Mecânica Clássica que estuda a descrição do movimento de corpos, a dinâmica é definida pela análise das causas do movimento de corpos. Ainda, Nussenzweig (2017, p.89) define a dinâmica como “[...] a determinação do tipo de movimento que ocorrerá em determinadas circunstâncias físicas [...]”. Aplicando o estudo da dinâmica à Robótica, Niku (2019, p.147) ressalta que “Para acelerar os elos de um robô, é necessário que se tenha atuadores grandes os suficientes capazes de exercer forças e torques grandes nos elos e juntas o suficiente para movê-los a uma velocidade e aceleração desejadas [...]”. Com a finalidade de encontrar essas forças e torques necessários para movimentar o robô, é necessário calcular as equações dinâmicas do movimento. Além disso, é necessário encontrar a dinâmica que rege o comportamento do manipulador para realizar simulações precisas. Isso faz com que não haja a necessidade da construção de um modelo físico para validação de projeto. Como explicitado por Craig (2014, p. 9) “Por meio da reformulação das equações dinâmicas de forma que a aceleração é computada como uma função do torque do atuador, é

possível simular como o manipulador irá se mover quando aplicados um conjunto de torques de atuadores”. Para robôs paralelos, a análise da dinâmica tende a ser muito complexa devido a não-linearidade do sistema por sua estrutura cinemática ser em malha fechada. Esse fator contribui para a dificuldade de utilização de manipuladores paralelos em aplicações que requerem respostas em tempo real e resposta dinâmica rápida.

As equações dinâmicas da Plataforma de Stewart-Gough podem ser obtidas, de maneira geral, por meio de três abordagens: análise por formulação de Newton-Euler, análise por mecânica Lagrangiana e, por fim, análise por meio do método do Trabalho Virtual. A maneira mais simples e, todavia, mais utilizada para se realizar essa análise é pela formulação de Newton-Euler. Isso se dá pela simplicidade de entendimento das Leis de Newton-Euler e pela maior facilidade de encontrar as equações do movimento de forma aberta e fechada, ou seja, de forma implícita e explícita. Ainda, de acordo com Taghirad (2013, p.167), todas as forças de reação do manipulador podem ser computadas, o que é muito útil no processo de projeto do manipulador. Todavia, sua principal desvantagem está na quantidade de equações necessárias para a solução do problema. Mais equações significam mais custo computacional e, por fim, menor *performance* do robô. Essa metodologia para obtenção das equações dinâmicas do robô pode ser observada no artigo de Dasgupta (1998) intitulado “*A Newton-Euler formulation for the inverse dynamics of the Stewart platform manipulator*”.

Em adição ao cálculo da dinâmica pelo método de Newton-Euler para o manipulador proposto, é notável a aplicação da Mecânica de Lagrange para a solução do problema. Por meio dessa formulação, é possível reduzir o número de equações necessárias para a solução da dinâmica do manipulador por meio da eliminação de forças reativas indesejáveis utilizando-se de coordenadas generalizadas – acarretando em um menor custo computacional. Essa demonstração é feita por Sumnu, Güzelbey e Çakir (2017) em seu artigo “*Simulation and PID control of a Stewart platform with linear motor*”.

Por fim, pode-se realizar a formulação dinâmica da plataforma de Stewart-Gough se utilizando do Princípio do Trabalho Virtual. Essa metodologia resulta na simplificação dos cálculos necessários para se encontrar as equações de movimento do robô por meio da eliminação das forças de restrição do mesmo, o que, por consequência, como descrito por Tsai (2000), resulta em uma maior eficiência computacional em relação aos métodos de Newton-Euler e Lagrange. Por esse motivo, a modelagem dinâmica do manipulador será realizada por meio da análise por Princípio do Trabalho Virtual, neste trabalho. A principal desvantagem desse método é a necessidade do cálculo da matriz Jacobiana referente ao manipulador e a todos os elos.

2.9 FORMULAÇÃO DINÂMICA DE ROBÔS PARALELOS POR MEIO DO MÉTODO DO TRABALHO VIRTUAL

No método do Trabalho Virtual, como descrito por Taghirad (2013, p. 221), “[...] *forças e momentos inerciais* são computados usando as acelerações angulares e lineares de cada um dos corpos rígidos. Então, o manipulador inteiro é considerado em equilíbrio estático e o princípio do Trabalho Virtual é aplicado para derivar as forças dos atuadores[...]”. Ou seja, por essa metodologia é possível desconsiderar todas as forças e momentos de restrição que atuam sobre o manipulador e considerar somente as *forças e momentos inerciais*, aplicando-se o princípio do Trabalho Virtual. Essas forças e momentos inerciais derivam do princípio de d’Alembert, as quais serão explicitadas na seção a seguir.

2.9.1 Princípio de d’Alembert

O princípio de d’Alembert deriva da segunda Lei de Newton do movimento, transformando a análise do problema de dinâmica para estática. A segunda Lei de Newton explicita-se como:

$$\sum_{i=1}^n \vec{F}_i = m \cdot \vec{a} \quad (23)$$

Onde o somatório de todas as forças atuantes em um sistema dinâmico é igual a massa do sistema vezes a aceleração do sistema em análise. O princípio de d’Alembert assume que o sistema em análise está em equilíbrio estático. Quando o observador se encontra em um sistema de referência coincidente com o centro de massa do corpo rígido ou partícula em movimento, aparenta-se que esse corpo ou partícula encontra-se em equilíbrio estático, com forças fictícias contrárias às forças externas reais atuantes no sistema. D’Alembert modifica a segunda Lei de Newton para o seguinte:

$$\sum \vec{F}_{ext} - m \cdot \vec{a}_c = 0 \quad (24)$$

Na Equação (24), nota-se que o sistema se encontra em equilíbrio, estando sujeito às ações das forças reais \vec{F}_{ext} e das forças fictícias $m \cdot \vec{a}_c$. Essas forças fictícias são as chamadas

forças inerciais. D'Alembert propôs essa modificação, pois com o sistema estando em equilíbrio estático é simplificado o cálculo para se encontrar as forças desconhecidas atuantes no sistema. Nota-se que \vec{a}_c corresponde à aceleração linear no centro de massa do corpo rígido. Ainda, é possível aplicar esse princípio à Equação de Euler, da seguinte forma:

$$\sum {}^c \vec{N}_{ext} - \left({}^c I \cdot \frac{d\vec{\omega}}{dt} + \vec{\omega} \times {}^c I \cdot \vec{\omega} \right) = 0 \quad (25)$$

Onde ${}^c \vec{N}_{ext}$ são os torques reais aplicados ao centro de massa do corpo rígido, ${}^c I$ é o momento de inércia tomado em relação ao centro de massa do corpo rígido, $\vec{\omega}$ é a velocidade angular do corpo rígido e $\frac{d\vec{\omega}}{dt}$ é a aceleração angular do corpo rígido. Ainda, pode-se definir as forças e momentos inerciais da seguinte maneira:

$$\vec{F}^* = -m \cdot \vec{a}_c \quad (26)$$

$${}^c N^* = - \left({}^c I \cdot \frac{d\omega}{dt} + \omega \times {}^c I \cdot \omega \right) \quad (27)$$

Por fim, pode-se realizar as seguintes definições:

$$\hat{F}_{ext} = \vec{F}_{ext} + \vec{F}^* \quad (28)$$

$${}^c \hat{N}_{ext} = {}^c \vec{N}_{ext} + {}^c \vec{N}^* \quad (29)$$

2.9.2 Princípio do Trabalho Virtual

O princípio do trabalho virtual baseia-se no proposto no princípio de d'Alembert. Assume-se que o corpo rígido se encontra em equilíbrio estático sobre a ação de forças externas reais e forças inerciais. A partir dessa assunção, aplica-se um deslocamento virtual ao sistema para que possa ser calculado o trabalho virtual decorrente desse deslocamento e, por consequência, as forças desconhecidas do sistema em equilíbrio estático.

O princípio do trabalho virtual pode ser entendido a partir do princípio do trabalho-energia, onde em um sistema no qual não há perda de energia por atrito ou qualquer outra forma de dissipação energética, pode-se relacionar uma variação infinitesimal de trabalho (W) do sistema com uma variação infinitesimal de energia potencial (U) e cinética (K), da seguinte forma:

$$dW = dK + dU \quad (30)$$

Onde:

$$dK = \sum_i m_i \cdot \vec{a}_{c_i} \cdot dx_i + \sum_i \frac{d {}^c H_i}{dt} \cdot d\theta_i \quad (31)$$

$$dU = \sum_i \vec{P}_i \cdot dy_i + \sum_i k_i \cdot z_i \cdot dx_i \quad (32)$$

Assumindo um sistema interconectado de corpos rígidos, como um robô, temos que m_i é a massa do corpo rígido i , dx_i e $d\theta_i$ representam uma taxa de variação infinitesimal de deslocamento linear e angular, ${}^c H_i$ representa o momento angular do corpo rígido i . Ainda, \vec{P} representa a força peso do corpo rígido, dy_i representa o deslocamento vertical desse corpo rígido, k é o coeficiente de deformação elástica do corpo rígido e z_i representa a deformação elástica do corpo rígido.

Com a explicitação desses conceitos, é possível definir o que é, de fato, o trabalho virtual. Aplicando-se o princípio de d'Alembert em um sistema dinâmico e, por consequência, o tornando em um sistema em equilíbrio estático, deve-se aplicar um deslocamento virtual $\delta(\cdot)$ para que se possa calcular o trabalho virtual e, finalmente, as forças desconhecidas do sistema. Conforme Taghirad (2013, p. 223), “Um deslocamento virtual $\delta(\cdot)$ é qualquer deslocamento assumido e arbitrário, linear ou angular, distante da posição natural ou atual do sistema [...]”. Esse deslocamento virtual deve, sempre, estar de acordo com as forças restritivas do sistema – para que se mantenha a validade do modelo dinâmico. Ainda, Taghirad (2013, p. 223) confirma que “É costumeiro o uso do símbolo diferencial d para se referir a mudanças diferenciais em

deslocamentos reais, enquanto o símbolo δ é usado para significar mudanças diferenciais as quais são assumidas ou chamadas de mudanças virtuais [...].” Devido a isso, o símbolo δ será utilizado neste trabalho para se referir a mudanças diferenciais virtuais.

Aplicando-se o princípio do trabalho-energia a deslocamentos virtuais δx (linear) e $\delta\theta$ (angular) em um sistema de corpos rígidos interconectados, pode-se reescrever a Equação (30) da seguinte maneira:

$$\delta W = \sum_i \vec{F}_i \cdot \delta x_i + \sum_i \vec{N}_i \cdot \delta\theta_i = 0 \quad (33)$$

Percebe-se que a variação virtual infinitesimal do trabalho na Equação (33) é igual a zero devido ao fato do sistema estar em equilíbrio estático. Na Equação (33), \vec{F}_i corresponde a todas as forças aplicadas externamente ao sistema, tais como: a força gravitacional, forças de atenuação, molas, etc. De forma concomitante, \vec{N}_i corresponde a todos os torques aplicados externamente ao sistema de corpos rígidos. De maneira conveniente, pode-se unir os vetores \vec{F}_i e \vec{N}_i em um único vetor $\vec{\Gamma}$ e os deslocamentos x_i e θ_i em um único vetor $\vec{\chi}_i$, da seguinte maneira:

$$\vec{\Gamma}_i = \begin{bmatrix} \vec{F}_i \\ \vec{N}_i \end{bmatrix} \quad (34)$$

$$\vec{\chi}_i = \begin{bmatrix} x_i \\ \theta_i \end{bmatrix} \quad (35)$$

Ainda, pode-se definir, por meio da aplicação do princípio de d’Alembert, um vetor $\vec{\Gamma}_i^*$ correspondente às forças e torques inerciais, da seguinte maneira:

$$\vec{\Gamma}_i^* = \begin{bmatrix} \vec{F}_i^* \\ \vec{N}_i^* \end{bmatrix}_{6 \times 1} = \begin{bmatrix} -m \cdot \vec{a}_c \\ -\left({}^c I \cdot \frac{d\omega}{dt} + \omega \times {}^c I \cdot \omega \right) \end{bmatrix} \quad (36)$$

A partir do exposto, pode-se aplicar o princípio do trabalho virtual ao sistema, obtendo-se a seguinte equação:

$$\delta W = \sum_i (\vec{\Gamma}_i + \vec{\Gamma}_i^*) \cdot \delta \vec{\chi}_i = 0 \quad (37)$$

No caso de um robô manipulador paralelo, deseja-se isolar as forças e toques exercidas pelos atuadores do robô do vetor que compreende as forças externas ao manipulador. Isso pode ser feito da seguinte maneira:

$$\delta W = \vec{\tau} \cdot \delta \vec{q} + \sum_i (\vec{\Gamma}_i + \vec{\Gamma}_i^*) \cdot \delta \vec{\chi}_i = 0 \quad (38)$$

Onde $\vec{\tau}$ é um vetor que representa as forças e torques exercidos por cada um dos atuadores do manipulador, \vec{q} é um vetor que representa as variáveis de junta do manipulador e $\vec{\Gamma}_i$ é um vetor que representa todas as forças externas aplicadas ao manipulador com exceção das forças e torques decorrentes dos atuadores do manipulador.

Aplicando a Equação (38) a um robô paralelo genérico, constituído de uma plataforma fixa, uma plataforma móvel e um número i de elos, pode-se obter a seguinte equação:

$$\delta W = \vec{\tau} \cdot \delta \vec{q} + (\vec{\Gamma}_p + \vec{\Gamma}_p^*) \cdot \delta \vec{\chi}_p + \sum_i (\vec{\Gamma}_i + \vec{\Gamma}_i^*) \cdot \delta \vec{\chi}_i = 0 \quad (39)$$

Na qual $\vec{\Gamma}_p$ é um vetor que armazena os valores de forças e torques externas ao manipulador aplicadas no centro de massa da plataforma móvel – excluindo-se as forças e torques decorrentes dos atuadores -, $\vec{\Gamma}_p^*$ é um vetor que armazena os valores de forças e torques inerciais aplicadas no centro de massa da plataforma móvel, $\vec{\chi}_p$ é o vetor que armazena os deslocamentos virtuais lineares e angulares da plataforma móvel, $\vec{\Gamma}_i$ é um vetor que armazena os valores de forças e torques externas as elos do robô aplicadas em seus centros de massa, $\vec{\Gamma}_i^*$ é um vetor que armazena os valores de forças e torques inerciais aplicadas no centro de massa dos elos do robô e $\vec{\chi}_i$ $\vec{\chi}_p$ é o vetor que armazena os deslocamentos virtuais lineares e angulares de cada um dos centros de massa dos elos do robô.

Ainda, pode-se aplicar as Equações (28) e (29) na Equação (39), obtendo o seguinte resultado:

$$\delta W = \vec{\tau} \cdot \delta \vec{q} + \delta \vec{\chi}_p^T \cdot \hat{\Gamma}_p + \sum_i \delta \vec{\chi}_i^T \cdot \hat{\Gamma}_i = 0 \quad (40)$$

Por fim, é necessário deixar todos os deslocamentos virtuais em relação à um conjunto de coordenadas generalizadas – neste caso foi escolhido, por conveniência, o deslocamento virtual referente a plataforma móvel -, com a finalidade de se criar um conjunto generalizado de deslocamentos virtuais que respeitem as restrições cinemáticas do manipulador. É possível relacionar os deslocamentos virtuais $\delta \vec{q}$ e $\delta \vec{\chi}_i$ com o deslocamento virtual $\delta \vec{\chi}_p$ por meio das matrizes Jacobianas do manipulador J e a matriz Jacobiana de cada um dos elos J_i , da seguinte maneira:

$$\delta \vec{q} = J \cdot \delta \vec{\chi}_p \quad (41)$$

$$\delta \vec{\chi}_i = J_i \cdot \delta \vec{\chi}_p \quad (42)$$

Finalmente, substituindo as Equações (41) e (42) na Equação (40), tem-se que:

$$\delta \vec{\chi}_p^T \cdot \left(J^T \cdot \vec{\tau} + \hat{\Gamma}_p + \sum_i J_i^T \cdot \hat{\Gamma}_i \right) = 0 \quad (43)$$

Sendo $\delta \vec{\chi}_p^T$ válido para qualquer valor arbitrário escolhido, a Equação (43) pode ser simplificada da seguinte maneira:

$$J^T \cdot \vec{\tau} + \hat{\Gamma}_p + \sum_i J_i^T \cdot \hat{\Gamma}_i = 0 \quad (44)$$

A Equação (44) demonstra a formulação da dinâmica de manipuladores paralelos por meio do princípio do trabalho virtual. Na seção a seguir será explicitada a aplicação dessa formulação para o caso da plataforma de Stewart-Gough.

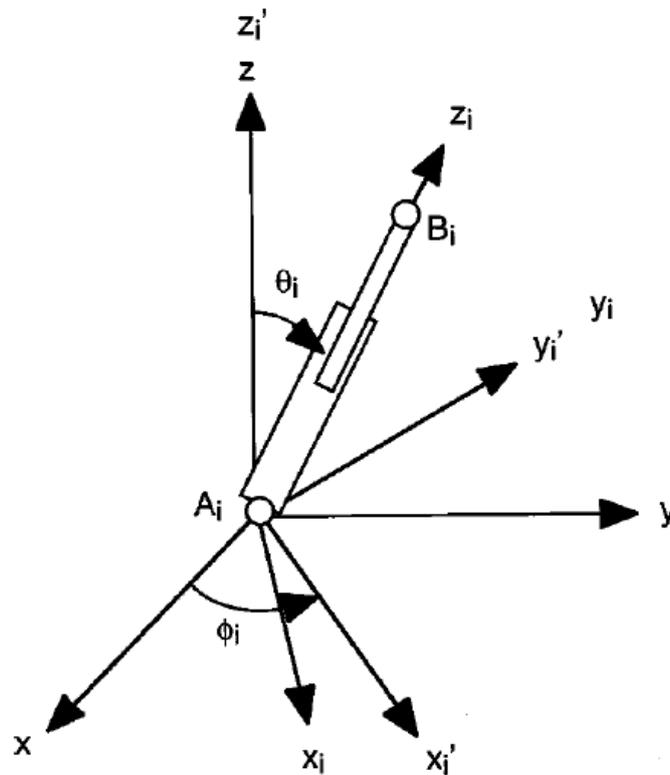
2.10 FORMULAÇÃO DINÂMICA DA PLATAFORMA DE STEWART-GOUGH POR MEIO DO MÉTODO DO TRABALHO VIRTUAL

A partir das equações deduzidas para um robô manipulador genérico, pode-se realizar a modelagem dinâmica pelo princípio do trabalho virtual da plataforma de Stewart-Gough. A análise cinemática do manipulador foi descrita na seção 2.4. Além disso, a metodologia de cálculo para as matrizes Jacobianas do manipulador e dos elos foi explicitada na seção 2.7. A estrutura de análise e dedução das equações foram realizadas a partir do exposto nos artigos “*Solving the Inverse Dynamics of a Stewart-Gough Manipulator by the Principle of Virtual Work*” de Tsai (2000) e “*Inverse dynamic explicit formulation of a Stewart-Gough manipulator based on the Virtual Work approach*” de Lemus, Rodriguez e Yime (2011).

2.10.1 Análise de posição

Primeiramente, deve-se realizar uma análise de posição referente aos elos do manipulador. Percebe-se que os elos são conectados à plataforma fixa por uma junta esférica. Por isso, pode-se descrever sua orientação por meio de dois ângulos de Euler. De maneira intuitiva, pode-se considerar um sistema de referência local $\{i\}$ coincidente com o sistema de referência $\{A\}$. Esse sistema de referência pode ser definido como, primeiramente, uma rotação de um ângulo ϕ_i ao redor do eixo z_i resultando no sistema de eixos de referência intermediário (x_i', y_i', z_i') . Após isso, realiza-se uma rotação de um ângulo θ_i ao redor de y_i' . A Figura 7 demonstra essas sucessivas rotações dos elos.

Figura 7 - Rotações por ângulos de Euler no sistema de referência {i}



Fonte: (TSAI, 2000, p.4).

Após essas sucessivas rotações, podemos definir uma matriz de rotação ${}^A R_i$ que transforma as coordenadas do sistema de referência {i} para o sistema de referência {A}, da seguinte maneira:

$${}^A R_i = \begin{bmatrix} c\phi_i \cdot c\theta_i & -s\phi_i & c\phi_i \cdot s\theta_i \\ s\phi_i \cdot c\theta_i & c\phi_i & s\phi_i \cdot s\theta_i \\ -s\theta_i & 0 & c\theta_i \end{bmatrix} \quad (45)$$

Nota-se que a terceira coluna da matriz demonstrada na Equação (45) corresponde às coordenadas do vetor ${}^A \hat{s}_i$, o qual dá direção aos elos. Assim:

$${}^A \hat{s}_i = \begin{bmatrix} c\phi_i \cdot s\theta_i \\ s\phi_i \cdot s\theta_i \\ c\theta_i \end{bmatrix} \quad (46)$$

Ainda, isolando-se o termo ${}^A\hat{s}_i$ da Equação (4), pode-se obter a seguinte relação:

$${}^A\hat{s}_i = \frac{({}^A\bar{P} + {}^B R_{ZY'X'} \cdot {}^B\bar{b}_i - {}^A\bar{a}_i)}{l_i} \quad (47)$$

Pode-se encontrar uma relação entre os senos e cossenos dos ângulos de Euler ϕ_i e θ_i e as coordenadas de ${}^A\hat{s}_i$ da seguinte maneira:

$$c\theta_i = {}^A\hat{s}_{iz} \quad (48)$$

$$s\theta_i = \sqrt{{}^A\hat{s}_{ix}^2 + {}^A\hat{s}_{iy}^2} \quad (0 \leq \theta_i \leq \pi) \quad (49)$$

$$s\phi_i = \frac{{}^A\hat{s}_{iy}}{s\theta}$$

$$(50)$$

$$c\phi_i = \frac{{}^A\hat{s}_{ix}}{s\theta} \quad (51)$$

Os termos ${}^A\hat{s}_{ix}$, ${}^A\hat{s}_{iy}$ e ${}^A\hat{s}_{iz}$, presentes nas equações acima, correspondem às coordenadas x, y e z do vetor ${}^A\hat{s}_i$. Por fim, observando-se a

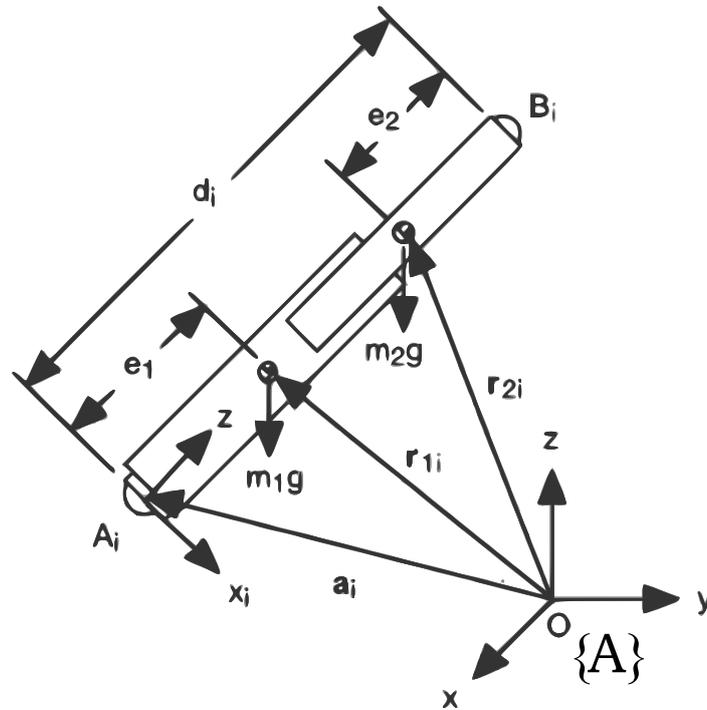
Figura 8, podem-se observar as seguintes relações:

$$\vec{r}_{1i} = {}^A\bar{a}_i + e_1 \cdot {}^A\hat{s}_i \quad (52)$$

$$\vec{r}_{2i} = {}^A\bar{a}_i + (l_i - e_2) \cdot {}^A\hat{s}_i \quad (53)$$

Nota-se que \vec{r}_{1i} corresponde ao vetor posição do cilindro de um dos elos i e \vec{r}_{2i} o vetor posição de um dos pistões de um dos elos i .

Figura 8 - Diagrama de corpo livre dos elos



Fonte: (TSAI, 2000, p.4, modificado pelo autor).

2.10.2 Análise de velocidade

O primeiro passo para a análise de velocidade é encontrar a velocidade do centro de massa das juntas B_i . Para isso, deriva-se o lado direito da Equação (2), obtendo-se:

$$\vec{v}_{bi} = \vec{v}_p + \vec{\omega}_p \times \vec{b}_i \quad (54)$$

Ainda, derivando-se o lado esquerdo da Equação (2), obtém-se:

$${}^i\vec{v}_{bi} = d_i \cdot {}^i\vec{\omega}_i \times {}^i\vec{s}_i + \frac{dl_i}{dt} \cdot {}^i\vec{s}_i \quad (55)$$

Onde ${}^i\vec{v}_{bi} = {}^A R_i^T \cdot \vec{v}_{bi}$. Realizando algumas simplificações, pode-se obter a seguinte equação para a velocidade angular do elo i com relação ao sistema de referência i :

$${}^i\vec{\omega}_i = l_i^{-1} \cdot ({}^i\vec{s}_i \times {}^i\vec{v}_{bi}) = l_i^{-1} \cdot \begin{bmatrix} -{}^i v_{bi,y} \\ {}^i v_{bi,x} \\ 0 \end{bmatrix} \quad (56)$$

Onde ${}^i v_{bi,x}$, ${}^i v_{bi,y}$ e ${}^i v_{bi,z}$ são as componentes x, y e z do vetor ${}^i\vec{v}_{bi}$. Após isso, pode-se definir as velocidades dos centros de massa do cilindro e do pistão que compõe o elo i , da seguinte maneira:

$${}^i\vec{v}_{1i} = e_1 \cdot {}^i\vec{\omega}_i \times {}^i\vec{s}_i = \frac{e_1}{l_i} \cdot \begin{bmatrix} {}^i v_{bi,x} \\ {}^i v_{bi,y} \\ 0 \end{bmatrix} \quad (57)$$

$${}^i\vec{v}_{2i} = (l_i - e_2) \cdot {}^i\vec{\omega}_i \times {}^i\vec{s}_i + \frac{dl_i}{dt} \cdot {}^i\vec{s}_i = l_i^{-1} \cdot \begin{bmatrix} (l_i - e_2) \cdot {}^i v_{bi,x} \\ (l_i - e_2) \cdot {}^i v_{bi,y} \\ l_i \cdot {}^i v_{bi,z} \end{bmatrix} \quad (58)$$

2.10.3 Matrizes Jacobianas dos elos

Conforme Tsai (2000, p. 5), “Uma etapa crítica na formulação das equações do movimento é a derivação das matrizes Jacobianas dos elos”. Primeiramente, deve-se definir a matriz Jacobiana J_{bi} . Essa é uma matriz que relaciona a velocidade da plataforma móvel com a velocidade do centro de massa das juntas B_i , com essa relação podendo ser escrita da seguinte maneira:

$${}^i\vec{v}_{bi} = J_{bi} \cdot \vec{v}_p \quad (59)$$

Ainda, define-se J_{bi} propriamente dita da seguinte maneira:

$$J_{bi} = \begin{bmatrix} 1 & 0 & 0 & 0 & b_{iz} & -b_{iy} \\ 0 & 1 & 0 & -b_{iz} & 0 & b_{ix} \\ 0 & 0 & 1 & b_{iy} & -b_{ix} & 0 \end{bmatrix} \quad (60)$$

Onde b_{ix} , b_{iy} e b_{iz} correspondem às componentes x, y e z do vetor ${}^A\vec{b}_i$. Ainda, pode-se referenciar J_{bi} em relação ao sistema de referência $\{i\}$, da seguinte forma:

$${}^i J_{bi} = {}^A R_i^T \cdot J_{bi} = \begin{bmatrix} {}^i J_{bix} \\ {}^i J_{biy} \\ {}^i J_{biz} \end{bmatrix} \quad (61)$$

Onde ${}^i J_{bix}$ corresponde à primeira linha da matriz J_{bi} , ${}^i J_{biy}$ corresponde à segunda linha da matriz J_{bi} e ${}^i J_{biz}$ corresponde à terceira linha da matriz J_{bi} . A próxima etapa é escrever a matriz Jacobiana do manipulador, como explicitada na seção 2.7. Assim:

$$J_p = \begin{bmatrix} {}^1 J_{1,z} \\ {}^2 J_{2,z} \\ {}^3 J_{3,z} \\ {}^4 J_{4,z} \\ {}^5 J_{5,z} \\ {}^6 J_{6,z} \end{bmatrix} \quad (62)$$

Finalmente, pode-se obter as matrizes jacobianas dos elos, onde, como explicitado anteriormente, o número 1 refere-se ao cilindro do atuador e o número 2 ao pistão. Assim:

$${}^i \vec{v}_{1i} = {}^i J_{1i} \cdot \frac{d\vec{\chi}_p}{dt} \quad (63)$$

$${}^i \vec{v}_{2i} = {}^i J_{2i} \cdot \frac{d\vec{\chi}_p}{dt} \quad (64)$$

Onde:

$${}^i J_{1i} = l_i^{-1} \cdot \begin{bmatrix} e_1 \cdot {}^i J_{bix} \\ e_1 \cdot {}^i J_{biy} \\ 0_{1 \times 6} \\ -{}^i J_{biy} \\ {}^i J_{bix} \\ 0_{1 \times 6} \end{bmatrix} \quad (65)$$

$${}^i J_{2i} = l_i^{-1} \cdot \begin{bmatrix} (l_i - e_2) \cdot {}^i J_{bix} \\ (l_i - e_2) \cdot {}^i J_{biy} \\ l_i \cdot {}^i J_{biz} \\ -{}^i J_{biy} \\ {}^i J_{bix} \\ 0_{1 \times 6} \end{bmatrix} \quad (66)$$

2.10.4 Análise de aceleração

Para realizar a análise de aceleração, deve-se, primeiramente, derivar a Equação (54) em relação ao tempo, obtendo como resultado da operação a seguinte equação:

$$\vec{a}_{bi} = \vec{a}_p + \vec{\alpha}_p \times \vec{b}_i + \vec{\omega}_p \times (\vec{\omega}_p \times \vec{b}_i) \quad (67)$$

Ainda, pode-se encontrar ${}^i \vec{a}_{bi}$ derivando a Equação (55) em relação ao tempo, da seguinte forma:

$${}^i \vec{a}_{bi} = \frac{d^2 l_i}{dt^2} \cdot {}^i \vec{s}_i + l_i \cdot {}^i \vec{\alpha}_i \times {}^i \vec{s}_i + l_i \cdot {}^i \vec{\omega}_i \times ({}^i \vec{\omega}_i \times {}^i \vec{s}_i) + 2 \cdot \frac{dl_i}{dt} \cdot {}^i \vec{\omega}_i \times {}^i \vec{s}_i \quad (68)$$

Finalmente, após algumas operações algébricas e simplificações, pode-se definir a aceleração angular ${}^i \vec{\alpha}_i$ da seguinte maneira:

$${}^i\vec{\alpha}_i = l_i^{-1} \cdot ({}^i\vec{s}_i \times {}^i\vec{v}_{bi}) - 2 \cdot \frac{dl_i}{dt} \cdot l_i^{-1} \cdot {}^i\vec{\omega}_i = l_i^{-2} \cdot \begin{bmatrix} -l_i \cdot {}^i a_{bi,y} + 2 \cdot {}^i v_{bi,z} \cdot {}^i v_{bi,y} \\ l_i \cdot {}^i a_{bi,x} + 2 \cdot {}^i v_{bi,z} \cdot {}^i v_{bi,x} \\ 0 \end{bmatrix} \quad (69)$$

Para se encontrar as acelerações do centro de massa de cada cilindro e pistão para cada elo, é utilizada a seguinte relação, proposta por Lemus, Rodriguez e Yime (2011):

$$\frac{d^2 {}^i\vec{\chi}_{li}}{dt^2} = \frac{d^i J_{li}}{dt} \cdot \frac{d\vec{\chi}_p}{dt} + {}^i J_{li} \cdot \frac{d^2 \vec{\chi}_p}{dt^2} \quad (70)$$

$$\frac{d^2 {}^i\vec{\chi}_{2i}}{dt^2} = \frac{d^i J_{2i}}{dt} \cdot \frac{d\vec{\chi}_p}{dt} + {}^i J_{2i} \cdot \frac{d^2 \vec{\chi}_p}{dt^2} \quad (71)$$

Onde:

$${}^i\vec{\chi}_{li} = \begin{bmatrix} {}^i\vec{a}_{li} \\ {}^i\vec{\alpha}_{li} \end{bmatrix} \quad (72)$$

$${}^i\vec{\chi}_{2i} = \begin{bmatrix} {}^i\vec{a}_{2i} \\ {}^i\vec{\alpha}_{2i} \end{bmatrix} \quad (73)$$

2.10.5 Formulação dinâmica implícita

Com as definições e equações deduzidas nas seções anteriores, pode-se formular as equações dinâmicas do movimento da plataforma de Stewart-Gough de forma implícita, por meio do método do trabalho virtual. Analisando-se a Equação (44), pode-se reescreve-la para o caso específico da plataforma de Stewart-Gough, da seguinte maneira:

$$\mathbf{J}_p^T \cdot \vec{c} + \hat{\Gamma}_p + \sum_{i=1}^6 \left({}^i J_{i1}^T \cdot {}^i \hat{\Gamma}_{i1} + {}^i J_{i2}^T \cdot {}^i \hat{\Gamma}_{i2} \right) = \mathbf{0} \quad (74)$$

Nota-se que os vetores de forças e torques aplicados aos elos foram divididos em dois, levando em consideração os centros de massa de cada pistão e cilindro de cada elo. Pode se definir o vetor $\hat{\Gamma}_p$ da seguinte forma:

$$\hat{\Gamma}_p = \begin{bmatrix} \vec{F}_p \\ \vec{N}_p \end{bmatrix} = \begin{bmatrix} \vec{F}_e + m_p \cdot (\vec{g} - \vec{a}_p) \\ \vec{N}_e - {}^A I_p \cdot \vec{\alpha}_p - \vec{\omega}_p \times ({}^A I_p \cdot \vec{\omega}_p) \end{bmatrix} \quad (75)$$

Onde \vec{F}_e e \vec{N}_e são as forças e torques externos aplicados à plataforma de Stewart-Gough, m_p é a massa da plataforma móvel, \vec{g} é o vetor representando a aceleração gravitacional e ${}^A I_p$ é a matriz que armazena os momentos de inércia da plataforma móvel – tomados no centro de massa dela. Por seguinte, pode se definir ${}^i \hat{\Gamma}_{1i}$ e ${}^i \hat{\Gamma}_{2i}$ como:

$${}^i \hat{\Gamma}_{1i} = \begin{bmatrix} {}^i \vec{F}_{1i} \\ {}^i \vec{N}_{1i} \end{bmatrix} = \begin{bmatrix} m_{1i} \cdot {}^A R_i^T \cdot \vec{g} - m_{1i} \cdot {}^i \vec{a}_{1i} \\ -{}^i I_{1i} \cdot {}^i \vec{\alpha}_i - {}^i \vec{\omega}_i \times ({}^i I_{1i} \cdot {}^i \vec{\omega}_i) \end{bmatrix} \quad (76)$$

$${}^i \hat{\Gamma}_{2i} = \begin{bmatrix} {}^i \vec{F}_{2i} \\ {}^i \vec{N}_{2i} \end{bmatrix} = \begin{bmatrix} m_{2i} \cdot {}^A R_i^T \cdot \vec{g} - m_{2i} \cdot {}^i \vec{a}_{2i} \\ -{}^i I_{2i} \cdot {}^i \vec{\alpha}_i - {}^i \vec{\omega}_i \times ({}^i I_{2i} \cdot {}^i \vec{\omega}_i) \end{bmatrix} \quad (77)$$

Nas equações (76) e (77), assume-se que a única força externa atuante nos elos é a de atração gravitacional. Ainda, todas as resultantes de forças e torques são aplicadas nos centros de massa de cada cilindro e pistão de cada elo, expressadas no sistema de referência {i}.

Finalmente, substituindo as equações (62), (65), (66), (75), (76) e (77) na Equação (74) pode-se obter um modelo dinâmico implícito da plataforma de Stewart-Gough. Essa equação descreve como o manipulador se comporta dinamicamente e pode ser usada para simulações precisas da dinâmica inversa do manipulador. Posteriormente, será realizada essa simulação da dinâmica inversa do manipulador com a finalidade de validação do modelo dinâmico proposto.

2.10.6 Formulação dinâmica explícita

A partir do exposto na Equação (74), pôde-se obter a formulação dinâmica implícita do manipulador. Apesar da Equação (74) ser útil para o cálculo da dinâmica inversa do robô, de

acordo com Taghirad (2013, p. 207), “[...] sua natureza implícita a torna desagradável para se realizar a análise dinâmica e controle[...]”. A partir disso, é possível alterar a Equação (74) para a forma explícita da equação do movimento da plataforma de Stewart-Gough. Essa forma explícita – ou de forma fechada - possui o formato conforme a Equação (78) e é comparável, de acordo com Taghirad (2013, p.207), “[...] com aquela usualmente obtida para manipuladores seriais [...]”.

$$M(\vec{\chi}) \cdot \frac{d^2 \vec{\chi}}{dt^2} + C\left(\vec{\chi}, \frac{d\vec{\chi}}{dt}\right) \cdot \frac{d\vec{\chi}}{dt} + G(\vec{\chi}) = \vec{\Gamma} \quad (78)$$

Onde M é uma matriz de inércia do manipulador, C é uma matriz de forças Coriolis e centrífugas do manipulador e G é a matriz de força gravitacional do manipulador. Conforme explicitado por Lemus, Rodriguez e Yime (2011), pode se encontrar essas matrizes da seguinte maneira:

$$M = M_p + \sum_{i=1}^6 \left(\sum_{j=1}^2 J_{i,j}^T \cdot M_{i,j} \cdot J_{i,j} \right) \quad (79)$$

$$C = C_p + \sum_{i=1}^6 \left(\sum_{j=1}^2 J_{i,j}^T \cdot C_{i,j} \cdot J_{i,j} + J_{i,j}^T \cdot M_{i,j} \cdot \frac{dJ_{i,j}}{dt} \right) \quad (80)$$

$$G = G_p + \sum_{i=1}^6 \left(\sum_{j=1}^2 J_{i,j}^T \cdot G_{i,j} \right) \quad (81)$$

Onde M_p , C_p e G_p são as matrizes de inércia, forças Coriolis e centrífugas e a matriz de força gravitacional da plataforma móvel, $M_{i,j}$, $C_{i,j}$ e $G_{i,j}$ são as matrizes de inércia, forças Coriolis e centrífugas e a matriz de força gravitacional de cada um dos cilindros e pistões dos elos e $J_{i,j}$ e $\frac{dJ_{i,j}}{dt}$ são a matriz jacobiana e a derivada temporal da matriz jacobiana de cada um dos cilindros e pistões dos elos. De acordo com Lemus, Rodriguez e Yime (2011), as matrizes de inércia, de forças Coriolis e centrífugas e de força gravitacional, para a plataforma móvel e para os componentes dos elos, podem ser encontradas a partir das equações a seguir:

$$M_k = \begin{bmatrix} m_k \cdot I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & {}^A I_k \end{bmatrix}_{6 \times 6} \quad (82)$$

$$C_k = \begin{bmatrix} 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & {}^A \vec{\omega}_{k \times} \cdot {}^A I_k \end{bmatrix}_{6 \times 6} \quad (83)$$

$$G_k = \begin{bmatrix} -m_k \cdot \vec{g} \\ 0_{3 \times 1} \end{bmatrix}_{6 \times 1} \quad (84)$$

Onde k pode assumir a representação da plataforma móvel p e dos componentes dos elos i, j . Ainda, m representa a massa do k -ésimo elemento, $I_{3 \times 3}$ representa uma matriz identidade de dimensão 3, ${}^A I_k$ representa o tensor de inércia do k -ésimo elemento, ${}^A \vec{\omega}_{k \times}$, representa a matriz antissimétrica da velocidade angular do k -ésimo elemento e \vec{g} é o vetor de aceleração gravitacional.

Realizando os cálculos necessários, substitui-se as equações (79) a (84) na Equação (78) e, finalmente, obtém-se o modelo dinâmico explícito (ou de forma fechada) da plataforma de Stewart-Gough. A partir desse modelo dinâmico é que será realizado o projeto do controlador para o manipulador, como descrito na seção seguinte.

2.11 SISTEMAS DE CONTROLE PARA ROBÔS PARALELOS

Conforme Di Steffano, Stubberud e Williams (1994, p. 1), “Um sistema de controle é um arranjo de componentes físicos conectados ou relacionados de maneira a comandar, dirigir ou regular a si mesmo ou a outro sistema”. Apresentada essa definição, pode se dividir sistemas de controle em duas classificações: sistemas de controle em malha fechada e sistemas de controle em malha aberta.

Sistemas de controle em malha fechada são, por diversas vezes, denominados sistemas de controle com *feedback* – ou realimentação. De acordo com Ogata (2010, p.7),

Em um sistema de controle de malha fechada, o sinal de erro atuante, que é a diferença entre o sinal de entrada e o sinal de realimentação (que pode ser o próprio sinal de saída ou uma função do sinal de saída e suas derivadas e/ou integrais), realimenta o controlador, de modo a minimizar o erro e acertar a saída do sistema ao valor desejado.

A principal vantagem dos sistemas de controle em malha fechada é a sua robustez devido ao fato de que, de acordo com Ogata (2010), a resposta do sistema fica relativamente insensível a

distúrbios externos e a variações internas nos parâmetros do sistema. Todavia, são mais difíceis de serem construídos e necessitam de constante medição da saída da malha de controle.

De forma análoga, os sistemas de controle em malha aberta são, de acordo com Ogata (2010, p.7), “[...] aqueles em que o sinal de saída não exerce nenhuma ação de controle no sistema”. As principais vantagens desse tipo de sistema de controle são, conforme Ogata (2010), a simplicidade de construção e fácil manutenção, a não apresentação de problemas de estabilidade, e sua adequação quando há aplicações na qual a medição da saída se torna muito difícil. Por fim, suas principais desvantagens são, pela análise de Ogata (2010), a ocorrência de erros por distúrbios e mudanças de calibração e a necessidade de uma regulagem periódica.

Para o sistema dinâmico proposto neste trabalho, será desenvolvido o dimensionamento de um caso específico de um sistema de controle em malha fechada: o controlador Proporcional-Derivativo. A escolha pelo controle em malha fechada se dá pela sua vantagem de tornar a resposta do sistema insensível a distúrbios externos e variações internas dos parâmetros do sistema.

2.12 CONTROLE DE ROBÔS PARALELOS

No quesito de controle de robôs paralelos, deve-se, primeiramente, definir para qual tipo de aplicação o robô será usado. No caso específico de robôs paralelos, destaca-se a aplicação de execução precisa de trajetória, onde, conforme Taghirad (2013, p. 269) “[...] a plataforma móvel do robô segue precisamente uma posição e orientação desejadas em um período de tempo específico, enquanto nenhuma força interativa precisa ser aplicada no ambiente”. Ainda, é notável a aplicação na qual o robô necessita aplicar uma força externa no ambiente, na qual são incluídas, de acordo com Taghirad (2013, p. 269), “[...] situações onde a plataforma móvel do robô está em contato com um ambiente rígido”. Para a aplicação na qual o robô precisa executar uma força em um ambiente externo, é utilizada a técnica de controle de força. Para a aplicação de execução precisa de trajetória, utiliza-se uma técnica de controle denominada controle de movimento. Taghirad (2013, p. 270) ressalta que “Para o controle de movimento de um manipulador, o controlador precisa computar as forças/torques dos atuadores requeridas para causar a movimentação da plataforma móvel de acordo com a trajetória desejada.”. Para este trabalho, foi escolhida a aplicação de controle de movimento do robô proposto, devido à sua menor complexidade de análise e projeto em relação à aplicação na qual o robô precisa exercer uma força externa no ambiente.

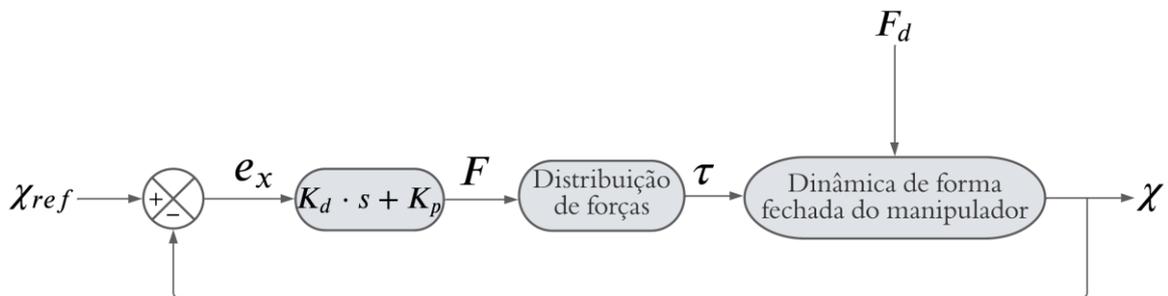
Com a definição da aplicação e técnica de controle a ser utilizada, é necessário realizar uma avaliação relacionada com a determinação e medição da variável de saída. No controle de movimento, de forma geral, representa-se a movimentação da plataforma móvel utilizando-se de um vetor que armazena os valores de posição e orientação dela. A técnica de controle na qual se escolhe realizar o controle utilizando esse vetor de posição e orientação como variável a ser medida na saída, chama-se controle no espaço de tarefa. Essa técnica possui como vantagem a medição e controle diretos da variável de saída, sem haver necessidade de se utilizar de variáveis intermediárias. Todavia, na prática, a medição direta de posição e orientação, por meio de acelerômetros e giroscópios precisos, encontra-se, ainda, em desenvolvimento. Isso acarreta em uma grande complexidade de implementação desse sistema de instrumentação e, por consequência, do sistema de controle. Para contornar esse problema, utiliza-se outra técnica de controle chamada de controle no espaço das juntas. Nessa técnica, as variáveis de junta são medidas diretamente e, por meio da cinemática inversa, são convertidas para a posição e orientação da plataforma. Na prática, é muito mais fácil realizar a medição das variáveis de junta, todavia possuem a desvantagem dessa conversão por meio do cálculo da cinemática inversa – o que pode acarretar na diminuição de desempenho do sistema de controle. Para este trabalho, foi escolhido o controle no espaço de tarefa. Essa escolha se justifica pelo fato de que o manipulador proposto foi somente concebido em um ambiente de simulação computacional, o que não traz desvantagem nenhuma pelo fato de que não há um modelo real para validação – eliminando a necessidade de um sistema complexo de instrumentação. Entretanto, caso o robô vier a ser construído fisicamente, é recomendado se utilizar do controle no espaço das juntas.

É notável, também, que robôs paralelos são representados, normalmente, por sistemas do tipo MIMO (*Multi-Input/ Multi-Output*) ou por sistemas não lineares. Todavia, muitos controladores industriais realizam a descentralização desse sistema de multivariáveis em várias malhas de controle lineares de representação SISO (*Single-Input/ Single-Output*), realizando o controle de movimentações de juntas individualmente. Essa descentralização torna mais simples o projeto do controlador, pois utiliza métodos clássicos para esse projeto. Entretanto, essa técnica pode limitar a *performance* do robô devido às restrições que a estrutura em malha fechada das cadeias cinemáticas traz. Portanto, para aplicações que necessitam de um desempenho maior do robô, recomenda-se utilizar técnicas avançadas de controle, tais como controle não linear e modelagem por sistemas MIMO. (TAGHIRAD, 2013). Ainda, foi escolhida como técnica de controle a descentralização do sistema MIMO em várias malhas de controle lineares do tipo SISO, devido a maior facilidade de se projetar e analisar controladores lineares.

2.12.1 Controle Proporcional-Derivativo (PD) descentralizado da plataforma de Stewart-Gough no espaço de tarefa

O controle PD da plataforma de Stewart-Gough mostra-se o caso mais simples de controle descentralizado em malha fechada para esse manipulador. Como explicitado por Taghirad (2013), diferentes alternativas de controle linear podem ser utilizadas ao invés do controlador Proporcional-Derivativo. Todavia, a aplicação desse tipo de controlador é a forma mais simples de se preservar a estabilidade do manipulador, enquanto se obtém um desempenho de rastreamento adequado. Ainda, a aplicação desse controlador em um ambiente virtual consome pouco poder computacional. Esses são os principais motivos para a escolha dessa técnica de controle, neste trabalho. A topologia desse sistema de controle pode ser vista na Figura 9.

Figura 9 - Topologia do sistema de controle proposto



Fonte: Autor.

Na Figura 9, o termo $\chi_{ref} = [p_x \ p_y \ p_z \ \phi \ \psi \ \theta]^T$ refere-se ao vetor que armazena os valores de referência de posição e orientação para a trajetória que se deseja que o manipulador execute. O vetor $e_x = [e_{px} \ e_{py} \ e_{pz} \ e_{\phi} \ e_{\psi} \ e_{\theta}]^T$ possui seis componentes referentes ao erro de rastreamento de cada variável representada pelo vetor χ_{ref} . Após o cálculo do erro, o controlador PD transforma esse erro em uma ação de controle por meio de um vetor de forças e torques realizadas pelo manipulador, representados por

$F = [F_x \ F_y \ F_z \ T_x \ T_y \ T_z]^T$. Após isso, esse vetor de forças é passado por uma operação de distribuição de forças. Essa operação nada mais é do que a multiplicação do vetor de forças F pela inversa da transposta da matriz jacobiana J^{-T} do manipulador – o que faz com que o vetor F seja mapeado para cada um dos atuadores. O resultado dessa multiplicação resulta no vetor $\tau = [\tau_1 \ \tau_2 \ \tau_3 \ \tau_4 \ \tau_5 \ \tau_6]^T$ - o qual possui como componentes as forças e torques realizadas por cada um dos atuadores do manipulador. Por fim, o vetor τ é aplicado à dinâmica de forma fechada do manipulador – denotada pela Equação (78). O resultado dessa equação somado ao vetor de distúrbios de forças externas F_d resulta no vetor χ que representa os valores de posição e orientação executados de fato pelo manipulador, a partir da referência dada.

Por fim, as constantes proporcional e derivativa, para este controlador, devem ser obtidas por tentativa e erro. Na prática, a aplicação deste controlador possui algumas desvantagens. Dentre elas está o baixo desempenho no rastreamento do erro, em comparação a outras técnicas de controle mais avançadas – como controle *feed forward*, controle por dinâmica inversa, controle robusto, etc. Ainda, a determinação dos ganhos para operação ótima do controlador se mostra uma tarefa muito difícil. Por esses motivos, para uma aplicação real a uma plataforma de Stewart-Gough construída fisicamente, recomenda-se a aplicação de técnicas avançadas de controle – especialmente as que tratam o sistema dinâmico como MIMO e aplicam controladores não lineares.

Como exemplos de soluções de controle da plataforma de Stewart, pode-se citar o trabalho de conclusão de curso de Rossi (2016), no qual o autor realiza o controle PID da trajetória de uma bola que corre livremente em cima de uma Plataforma de Stewart-Gough – gerando um sistema bola-chapa. Ainda, é notável a dissertação de mestrado de Mello (2011), na qual a autora discorre sobre diferentes estratégias de controle para a plataforma de Stewart-Gough, como controle PID baseado na cinemática, controle por lógica *Fuzzy* e controle de torque computado. Sumnu, Güzelbey e Çakir (2017) apresentam em seu artigo denominado “*Simulation and PID control of a Stewart platform with linear motor*” o controle PID de posição de uma plataforma de Stewart-Gough utilizando-se de motores lineares elétricos – cujas dinâmicas são levadas em consideração na modelagem do sistema. Por fim, Taghirad (2013) em seu livro “*Parallel Robots: mechanics and control*” apresenta diversas técnicas de controle de força e posição para a plataforma de Stewart-Gough, tais como: controle PD descentralizado,

controle *feed-forward*, controle por dinâmica inversa, controle por linearização parcial por dinâmica inversa, controle adaptativo e controle robusto.

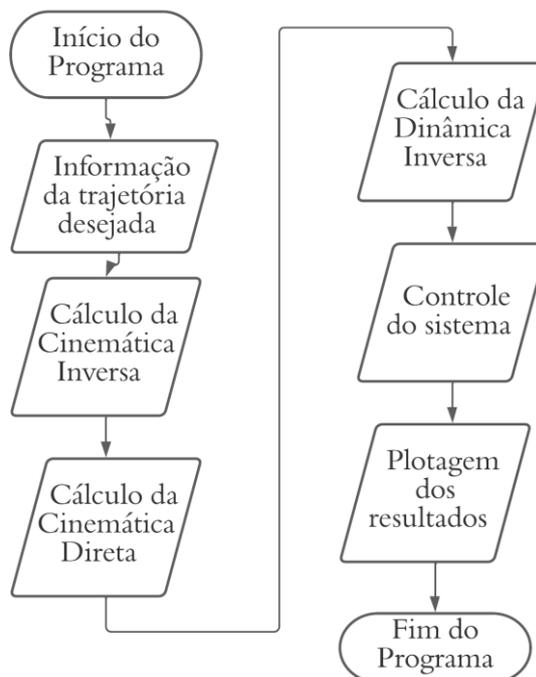
3 MATERIAIS E MÉTODOS

Neste Capítulo serão explicitados os procedimentos e ferramentas computacionais de análise e simulação virtual das cinemáticas direta e inversa do manipulador proposto, da dinâmica inversa do manipulador proposto e, por fim, do sistema de controle aplicado ao manipulador proposto. Será dada uma ênfase maior aos procedimentos de cálculo e simulação da dinâmica inversa e do algoritmo de controle. Todos esses algoritmos foram desenvolvidos para validação das equações deduzidas e modelos propostos durante o Capítulo 2, utilizando-se do *software Matlab*®.

3.1 ALGORITMO PRINCIPAL E PARÂMETROS FÍSICOS DO MANIPULADOR

Com a finalidade de organizar as etapas de simulação, foi criado um algoritmo principal, cuja lógica de funcionamento é demonstrada pelo fluxograma da Figura 10. Nele, encontram-se as chamadas de sub-rotinas que calculam a cinemática inversa, cinemática direta, dinâmica inversa e controle do sistema. Cada etapa do fluxograma corresponde a um código que calcula o demonstrado – os quais serão explicitados a seguir.

Figura 10 - Fluxograma para programa principal



A partir disso, deve se definir as propriedades físicas do manipulador a ser simulado. Neste trabalho, foram utilizados os mesmos parâmetros apresentados por Wang e Gosselin (1998). Essa escolha foi feita para realizar a validação do modelo proposto utilizando resultados já conhecidos como parâmetros.

Primeiramente, deve-se definir a posição das juntas da plataforma fixa e da plataforma móvel, representados pelos pontos A_i e B_i na Figura 6, da seguinte forma:

$$A = \begin{bmatrix} -2,120 & -2,380 & -2,380 & -2,120 & 0 & 0 \\ 1,374 & 1,224 & -1,224 & -1,374 & -0,15 & 0,15 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} m \quad (85)$$

$$B = \begin{bmatrix} 0,170 & -0,660 & -0,660 & -0,170 & 0,430 & 0,430 \\ 0,595 & 0,150 & -0,150 & -0,595 & -0,445 & 0,445 \\ -0,4 & -0,4 & -0,4 & -0,4 & -0,4 & -0,4 \end{bmatrix} m \quad (86)$$

Para esses dois vetores explicitados acima, as linhas representam as coordenadas x, y e z – contando-se de cima para baixo -, enquanto as colunas são as juntas – variando de 1 a 6. As coordenadas de A são em relação ao sistema de referência {A} e as de B em relação ao sistema de referência {B}. Após essa definição, deve se explicitar as propriedades de massa do manipulador e dos atuadores, da seguinte maneira:

$$m_p = 1,5Kg \quad (87)$$

$$m_{1,i} = m_{2,i} = 0,1Kg \quad (88)$$

$$e_1 = e_2 = 0,5m \quad (89)$$

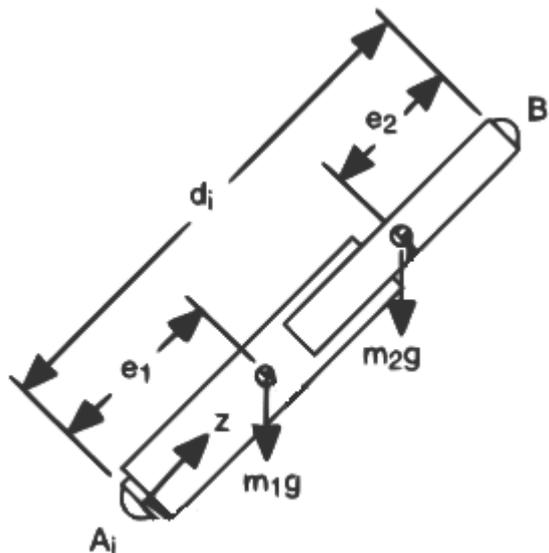
$${}^B I_p = \begin{bmatrix} 0,08 & 0 & 0 \\ 0 & 0,08 & 0 \\ 0 & 0 & 0,08 \end{bmatrix} Kg \cdot m^2 \quad (90)$$

$${}^i I_{1,i} = {}^i I_{2,i} = \begin{bmatrix} 0,00625 & 0 & 0 \\ 0 & 0,00625 & 0 \\ 0 & 0 & 0 \end{bmatrix} \text{Kg} \cdot \text{m}^2 \quad (91)$$

$$\vec{g} = [0 \quad 0 \quad -9,807]^T \text{m/s}^2 \quad (92)$$

Onde m_p é a massa da plataforma móvel, $m_{1,i}$ e $m_{2,i}$ são as massas do cilindro e do pistão de cada atuador, respectivamente. Ainda, ${}^B I_p$ é o momento de inércia da plataforma móvel e ${}^i I_{1,i}$ e ${}^i I_{2,i}$ são os momentos de inércia do cilindro e do pistão de cada atuador, respectivamente. Por fim, o vetor \vec{g} corresponde à aceleração gravitacional as distâncias e_1 e e_2 mostram-se na Figura 11. É notável que a única força de distúrbio a ser considerada no sistema é a força gravitacional.

Figura 11 - Distâncias e_1 e e_2



Fonte: (TSAI, 2000, p.4, modificado pelo autor).

A partir da definição desses parâmetros físicos, é necessário definir uma trajetória de referência que se deseja que o manipulador execute. Assim, foi utilizado neste trabalho a

trajetória proposta por Sumnu, Güzelbey e Çakir (2017), consistindo do seguinte vetor de trajetória para a plataforma móvel:

$$\chi_{ref}(t) = \begin{bmatrix} -1,5 + 0,1 \cdot \sin(\omega \cdot t) \\ 0 \\ 1 \\ 0,1 \cdot \sin(\omega \cdot t) \\ 0 \\ 1 \end{bmatrix} \quad (93)$$

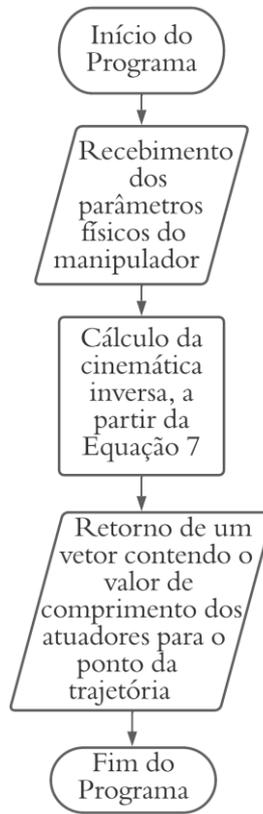
Onde ω é igual a 3 rad/s e $\omega \cdot t$ varia de 0 a 6,2832.

Finalmente, a partir do exposto nessa seção, pode-se realizar as simulações dos modelos descritos.

3.2 ALGORITMO PARA SIMULAÇÃO DA CINEMÁTICA INVERSA

Na simulação da cinemática inversa, busca-se encontrar o comprimento de cada um dos atuadores para a trajetória descrita pela Equação (93). Para a realização da simulação da cinemática inversa, foi utilizado o algoritmo representado na Figura 12. Nota-se que, pela simplicidade da equação e do cálculo, o algoritmo se torna muito simples, da mesma forma. A cinemática inversa se mostra extremamente importante, pois é utilizada para o cálculo da dinâmica inversa do manipulador e, posteriormente, no controle.

Figura 12 - Algoritmo para cálculo da cinemática inversa

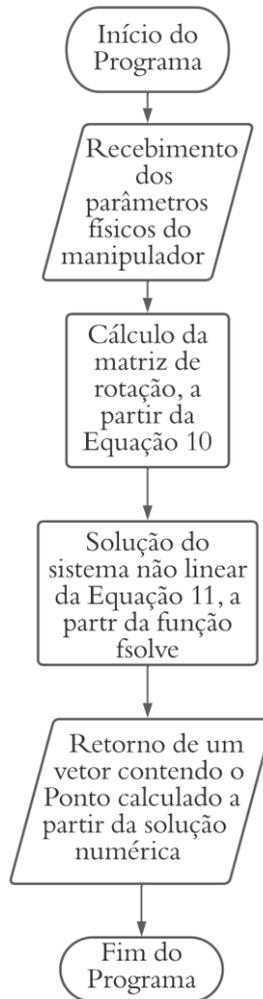


Fonte: Autor.

3.3 ALGORITMO PARA SIMULAÇÃO DA CINEMÁTICA DIRETA

Utilizada somente como objeto de estudo neste trabalho, o cálculo da cinemática direta, como descrito anteriormente, se mostra difícil para manipuladores em configuração paralela. Para este estudo, foram aplicadas as equações 9 a 11 no *software Matlab*®. A solução do sistema mostrado na Equação (11) foi resolvido numericamente por meio do método dos quadrados mínimos não lineares de Gauss-Newton com gradiente numérico e Jacobiano – utilizando-se da função *fsolve* do *Matlab*®. O algoritmo desenvolvido encontra-se na Figura 13

Figura 13 - Algoritmo para simulação da cinemática direta da plataforma de Stewart



Fonte: Autor.

3.4 ALGORITMO PARA SIMULAÇÃO DA DINÂMICA INVERSA

O problema da dinâmica inversa consiste no cálculo das forças as quais os atuadores precisam realizar para que a plataforma móvel execute uma dada trajetória. Foi escolhida a realização dessa simulação para validar o modelo dinâmico do robô – o qual será utilizado na simulação do algoritmo de controle. Para se realizar a simulação da dinâmica inversa do manipulador, foram seguidos os passos do algoritmo computacional proposto por Tsai (2000), realizando a implementação dele no *Matlab*®. Os passos são os seguintes:

1. Realizar a análise da cinemática inversa do manipulador, computando as variáveis para $i=1...6$:
 - a. Calcular o vetor b_i e o momento de inércia da plataforma móvel em relação ao sistema de referência {A},
 - b. Calcular o comprimento dos atuadores e do vetor direcional dos mesmos pelas equações (7) e (47),
 - c. Calcular os ângulos referentes ao sistema de referência {i}, bem como sua matriz de rotação pelas equações (45) e (48) a (51),
 - d. Calcular as velocidades lineares representadas pelas equações (54) e (55),
 - e. Calcular a velocidade angular, representada pela Equação (56),
 - f. Calcular as acelerações lineares, representadas pelas equações (67) e (68),
 - g. Calcular a aceleração angular, representada pela Equação (69),
 - h. Calcular as acelerações do cilindro e do pistão dos atuadores, dadas pelas equações (70) e (71).

2. Realizar o cálculo das matrizes Jacobianas:
 - a. Calcular as matrizes Jacobianas dadas pelas equações (60) e (61),
 - b. Calcular a matriz Jacobiana dada pela Equação (62).

3. Calcular a resultante das forças:
 - a. Calcular a resultante de forças e torques da plataforma, demonstradas pela Equação (75),
 - b. Calcular a resultante de forças e torques do cilindro e do pistão dos atuadores, dadas pelas equações (76) e (77).

4. Calcular a força nos atuadores pela seguinte equação, baseada na Equação (74), isolando-se o termo $\vec{\tau}$, da seguinte forma:

$$\vec{\tau} = -J_p^{-T} \cdot \left(\hat{\Gamma}_p + \sum_{i=1}^6 \left({}^i J_{i1}^T \cdot {}^i \hat{\Gamma}_{1i} + {}^i J_{i2}^T \cdot {}^i \hat{\Gamma}_{2i} \right) \right) \quad (94)$$

3.5 ALGORITMO PARA SIMULAÇÃO DO CONTROLE PD DESCENTRALIZADO

Para a explicação do algoritmo utilizado na simulação do controle do sistema, deve-se definir alguns parâmetros. O primeiro desses parâmetros é que o controlador foi implementado em um algoritmo computacional, portanto, é a versão digital de um controlador Proporcional-Derivativo – da mesma maneira demonstrada por Taghirad (2013). Esse controlador é demonstrado na Equação (95) – no qual são representadas todas as seis malhas de controle descentralizadas em uma única equação matricial.

$$u(t) = Kp \cdot (X - x) + Kv \cdot \left(\frac{dX}{dt} - \frac{dx}{dt} \right) \quad (95)$$

Nessa equação, $u(t)$ corresponde a ação de controle (representada pelo vetor F na Figura 9), Kp é a matriz de constantes proporcionais, Kv a matriz de constantes derivativas, X é a posição e orientação para a execução da trajetória de referência e x é a medição da orientação e posição simuladas da plataforma móvel. Ainda, após a realização de alguns testes práticos, as matrizes Kp e Kv foram definidas da seguinte forma:

$$Kp = \begin{bmatrix} 10000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10000 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 100000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 100000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100000 \end{bmatrix} \quad (96)$$

$$Kv = \begin{bmatrix} 1000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1000 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1000 \end{bmatrix} \quad (97)$$

Ainda, a trajetória executada pelo manipulador, baseada em uma trajetória de referência, é calculada a partir de um método numérico de solução de um sistema de equações diferenciais a partir de integrações numéricas – chamado de *ode45* no *Matlab*®. A utilização desse

algoritmo se dá pelo fato de que a obtenção da posição executada pelo manipulador se dá pela Equação (98) - encontrada pelo isolamento da derivada segunda do vetor de posição e orientação da plataforma móvel demonstrados na Equação (78). (SUMNU, GÜZELBEY E ÇAKIR, 2017).

$$\frac{d^2 \vec{\chi}}{dt^2} = M(\vec{\chi})^{-1} \cdot \left(J^{-T} \cdot \tau - C\left(\vec{\chi}, \frac{d\vec{\chi}}{dt}\right) \cdot \frac{d\vec{\chi}}{dt} + G(\vec{\chi}) \right) \quad (98)$$

A partir da Equação (98) é realizada uma função de custo que será integrada pela função *ode45* no *Matlab*® - resultando no valor medido virtualmente da execução da trajetória do manipulador. Essa função de custo é um vetor contendo a Equação (98) e a derivada da posição da trajetória executada, formando o seguinte sistema de equações diferenciais:

$$\left[\frac{d\vec{\chi}}{dt}, M(\vec{\chi})^{-1} \cdot \left(J^{-T} \cdot \tau - C\left(\vec{\chi}, \frac{d\vec{\chi}}{dt}\right) \cdot \frac{d\vec{\chi}}{dt} + G(\vec{\chi}) \right) \right] \quad (99)$$

O algoritmo implementado no *Matlab*® segue os seguintes passos:

1. Fornecer os parâmetros físicos e trajetória desejada,
2. Realizar a integração numérica iterativa, fornecendo um tempo de integração t:
 - a. Calcular a cinemática inversa da trajetória executada pelo manipulador no instante de tempo t,
 - b. Calcular o modelo dinâmico de forma fechada do manipulador no instante de tempo t,
 - c. Realizar a implementação digital do controlador proposto pela Equação (95)
 - d. Realizar a distribuição de forças na ação de controle,
 - e. Aplicar as forças dos atuadores no modelo dinâmico de forma fechada,
 - f. Aplicar todos os itens calculados na função de custo, estabelecida pela Equação (99),
 - g. Repetir o item 2 até completar o tempo de integração definido.

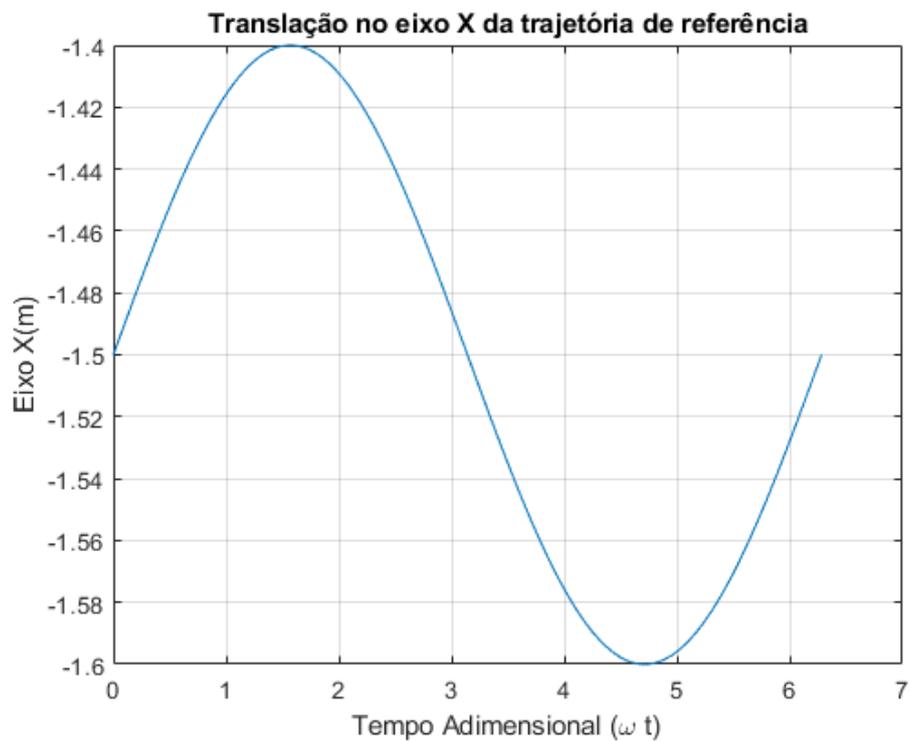
4 APRESENTAÇÃO E DISCUSSÃO DOS RESULTADOS

Neste Capítulo serão demonstrados e discutidos os resultados das simulações explicitadas no Capítulo 3: cinemáticas direta e inversa, dinâmica inversa e controle.

4.1 SIMULAÇÃO DA CINEMÁTICA INVERSA DO MANIPULADOR

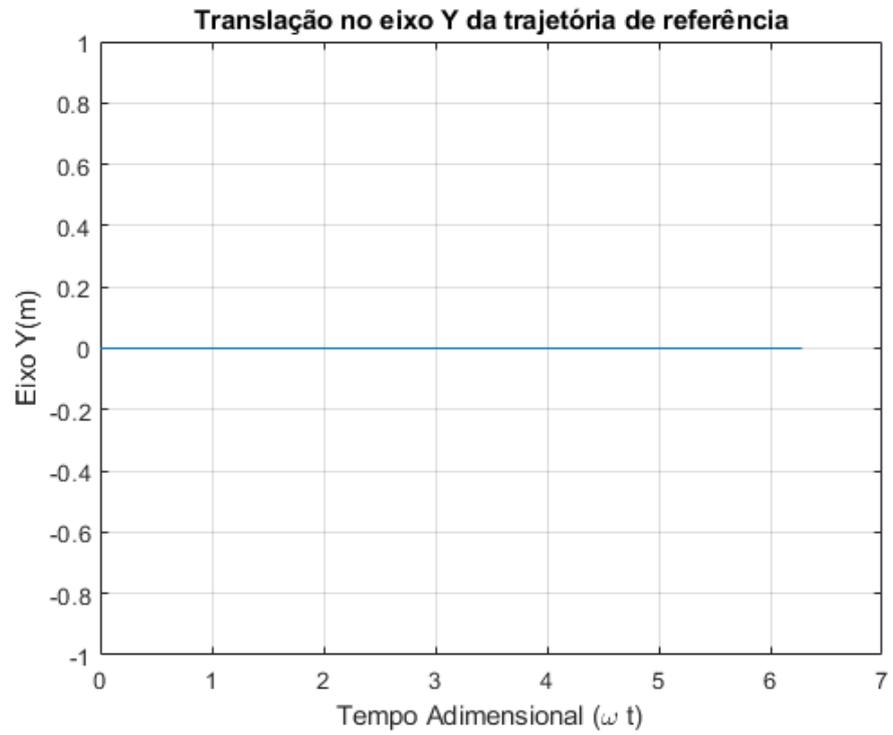
Como exposto no Capítulo 3, a cinemática inversa foi simulada utilizando-se da trajetória proposta. As figuras 13 a 18 demonstram as componentes em x, y e z de orientação e posição da trajetória de referência *versus* o tempo adimensional ωt - conforme demonstrado na seção 3.1.

Figura 14 - Translação no eixo x da trajetória de referência



Fonte: Autor.

Figura 15 - Translação no eixo y da trajetória de referência



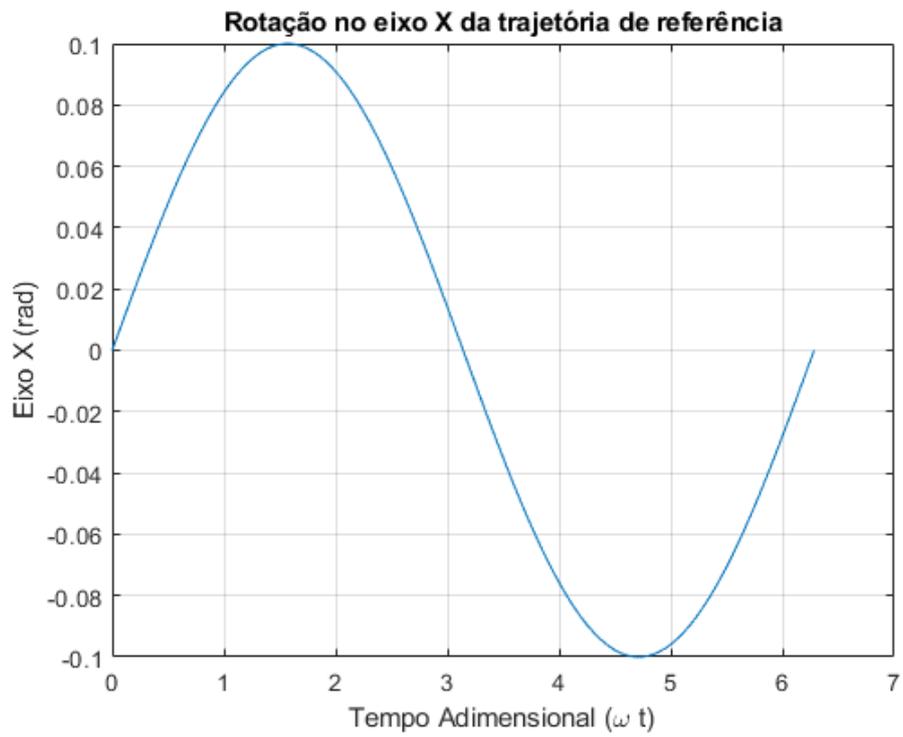
Fonte: Autor.

Figura 16 - Translação no eixo z da trajetória de referência



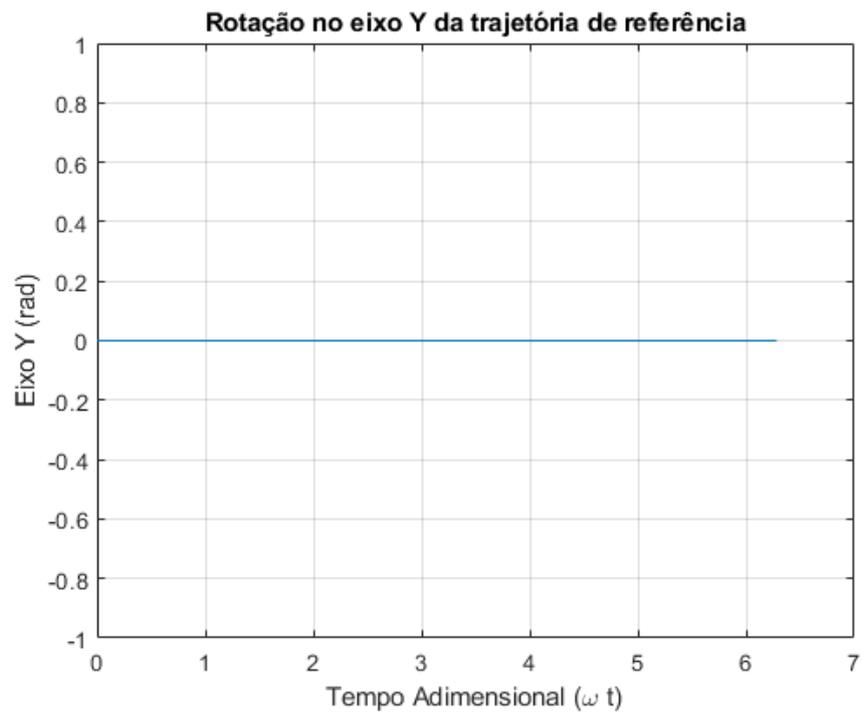
Fonte: Autor.

Figura 17 - Rotação no eixo x da trajetória de referência



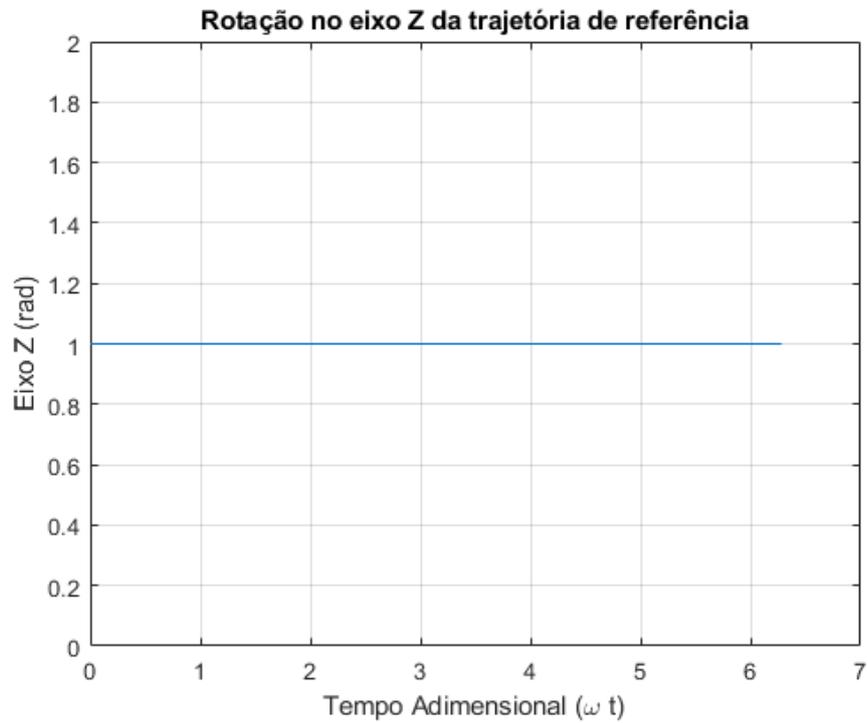
Fonte: Autor.

Figura 18 - Rotação no eixo y da trajetória de referência



Fonte: Autor.

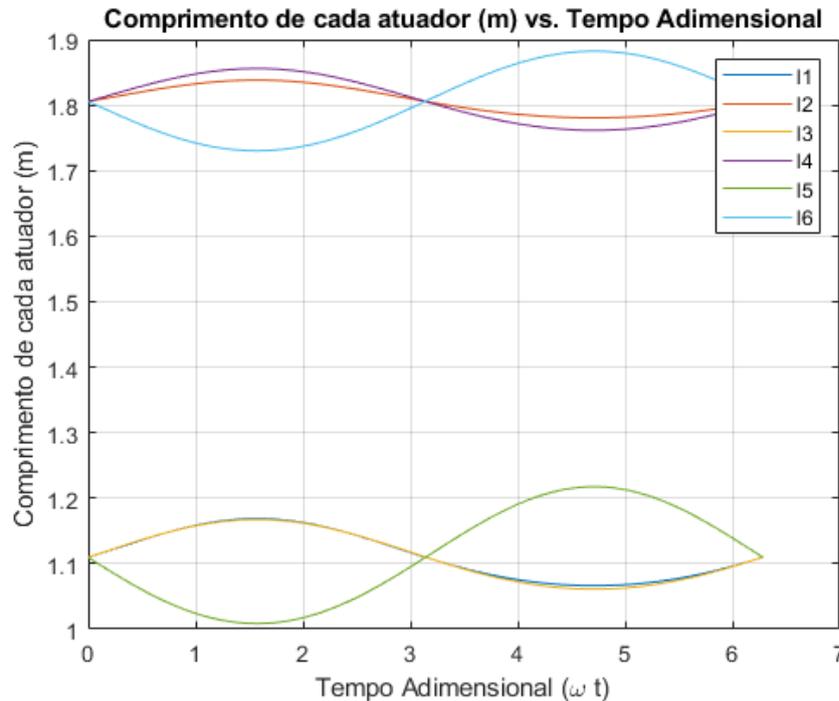
Figura 19 - Rotação no eixo z da trajetória de referência



Fonte: Autor.

A Figura 20 mostra um gráfico contendo em seu eixo x o tempo adimensional ωt e em seu eixo y o comprimento que cada um dos atuadores possuía em cada ponto da trajetória executada, em metros. Nota-se que o comprimento dos atuadores 11, 13 e 15 ficam juntos e distintos da variação de comprimento dos atuadores 12, 14 e 16 – com ambos os grupos possuindo comportamento senoidal. Isso se dá pela estrutura cinemática de forma fechada do manipulador e pela natureza senoidal da trajetória aplicada.

Figura 20 - Resultado da simulação da cinemática inversa



Fonte: Autor.

4.2 SIMULAÇÃO DA CINEMÁTICA DIRETA DO MANIPULADOR

Como explicitado anteriormente, o problema da cinemática direta consiste no cálculo de um vetor de posição e orientação da plataforma móvel baseado na informação de comprimento dos seis atuadores do manipulador. Para a realização da simulação da cinemática direta do manipulador, foi utilizado, como descrito anteriormente, o método *fsolve* do *Matlab*®. Como exemplo, foram passados para a função que calcula a cinemática direta os valores de comprimento da trajetória de referência no instante em que $t = 0$. Ainda, a função *fsolve* necessita de um chute inicial para convergência do algoritmo. Como chute inicial, após alguns testes, foram utilizados os seguintes valores:

$$x_{ini} = [-1 \ 0 \ 0,8 \ 0 \ 0 \ 0]^T m \quad (100)$$

Nota-se, que o vetor da Equação (100) – bem como todos os próximos vetores de posição e orientação a serem apresentados – é do mesmo formato do apresentado na Equação (8). Ainda, a função recebeu os seguintes valores de comprimento para cada atuador:

$$L(0) = [1,1098 \quad 1,8055 \quad 1,1101 \quad 1,8054 \quad 1,1097 \quad 1,8059]m \quad (101)$$

O resultado esperado de saída da função é o seguinte vetor:

$$x = [-1,5 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1]^T m \quad (102)$$

A função obteve como resultado do cálculo numérico o seguinte vetor de posição e orientação:

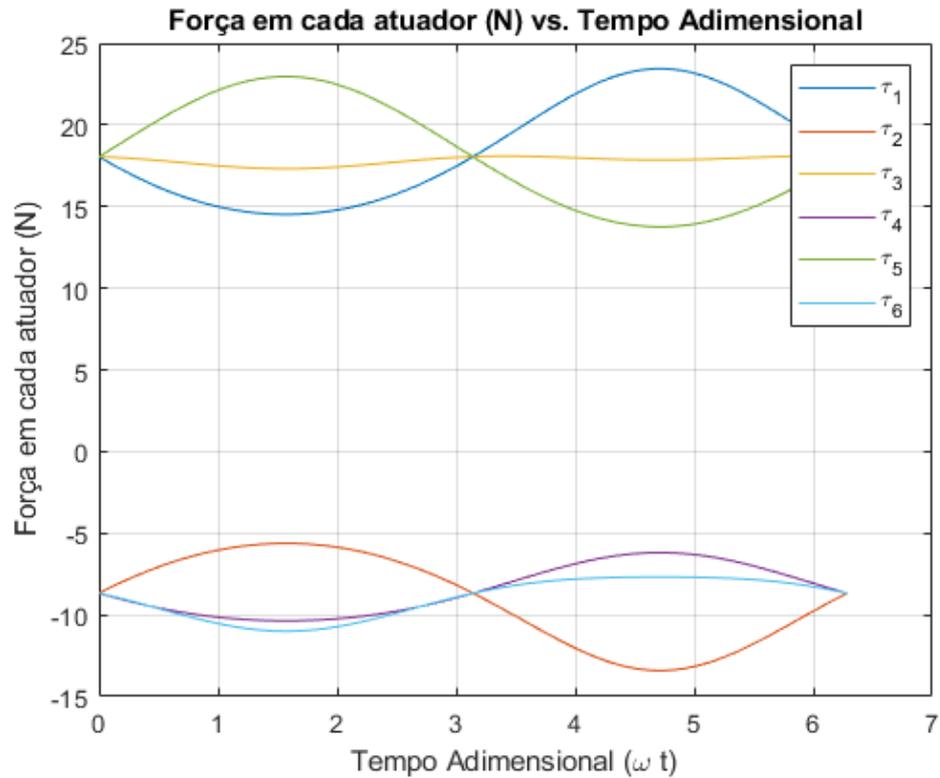
$$x_{calc} = \begin{bmatrix} -1,5000 \cdot 10^0 \\ -9,0816 \cdot 10^{-9} \\ 1,0000 \cdot 10^0 \\ 19,3649 \cdot 10^{-9} \\ -15,2686 \cdot 10^{-9} \\ 1,0000 \cdot 10^0 \end{bmatrix} m \quad (103)$$

Realizando uma comparação entre a Equação 103 e a Equação 102, nota-se que o resultado obtido foi satisfatório – salvo erros da solução numérica na qual os números se aproximam de zero. Apesar do resultado satisfatório obtido por meio da simulação, nota-se que o método numérico se mostra ineficiente devido ao fato de que o chute inicial precisa ser muito próximo do valor de convergência. Esse fato comprova o explicitado no Capítulo 2 – que a obtenção da dinâmica direta para a plataforma de Stewart-Gough encontra-se em fase de estudos, mesmo num ambiente computacional, para sua otimização.

4.3 SIMULAÇÃO DA DINÂMICA INVERSA DO MANIPULADOR

A dinâmica inversa consiste em um problema no qual se deseja descobrir as forças necessárias que os atuadores devem aplicar para o manipulador executar determinada trajetória pré-estabelecida. Para a realização da simulação da dinâmica inversa do manipulador, foi utilizado o algoritmo descrito no Capítulo 3. Para a trajetória descrita, obteve-se o seguinte resultado, demonstrado pela Figura 21.

Figura 21 - Resultado da simulação da dinâmica inversa do manipulador

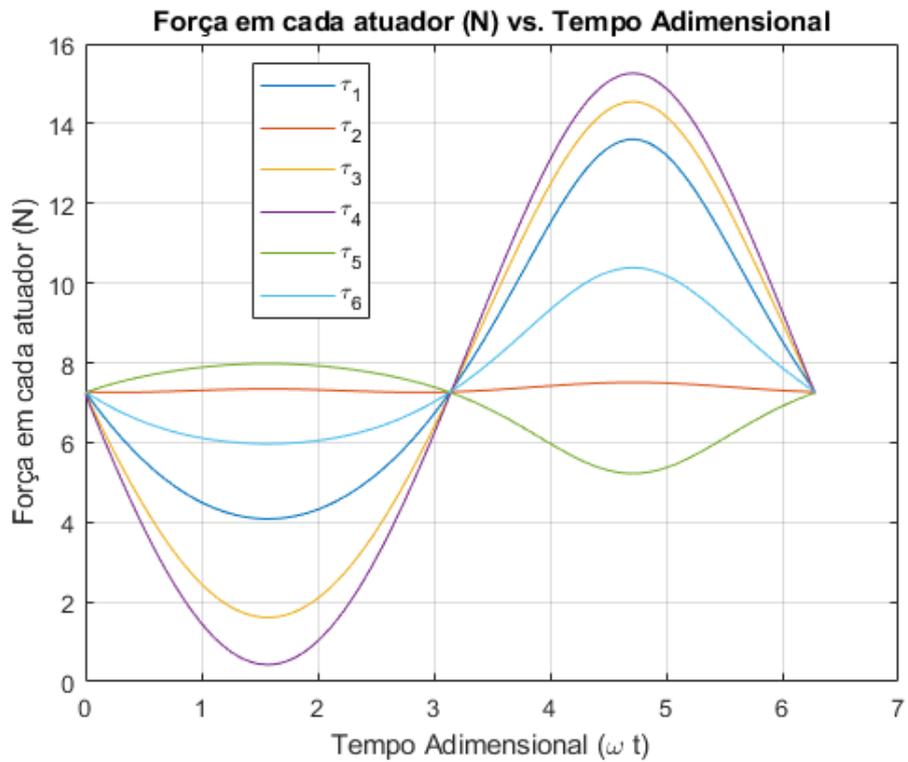


Fonte: Autor.

Da mesma forma, como explicitado na simulação da cinemática inversa do manipulador, há distinção nos grupos de atuadores de numeração par e ímpar, bem como um comportamento senoidal na aplicação das forças no período de tempo. Igualmente, isso se dá pela natureza do manipulador possuir estrutura cinemática de forma fechada e, ainda, da trajetória a qual o manipulador executou ser de natureza senoidal, tanto na translação quanto na orientação.

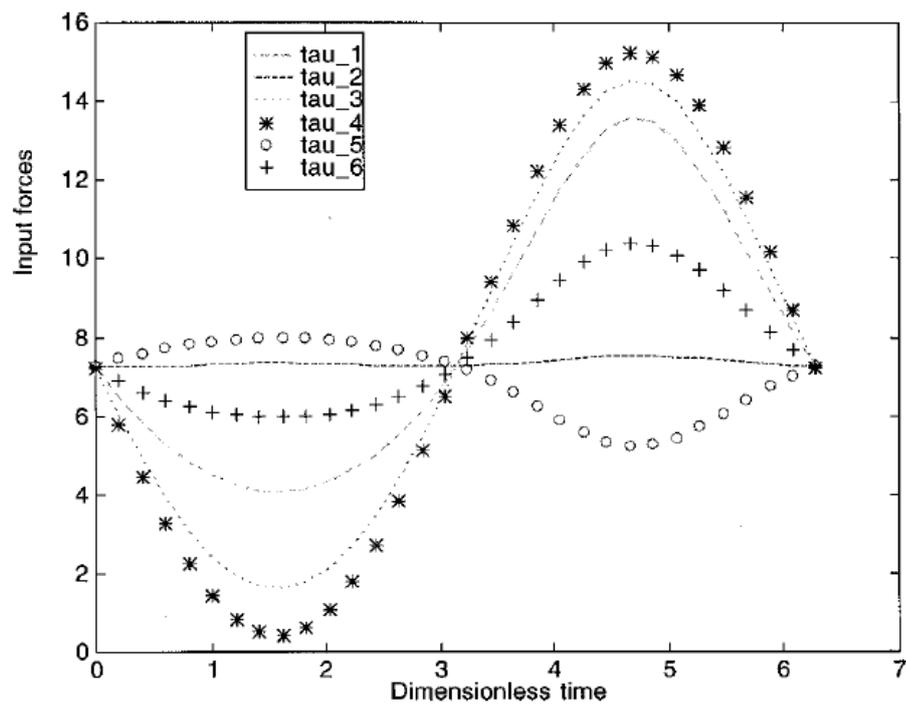
Foi simulada, também, para fins de validação da implementação, a trajetória descrita no artigo de Tsai (2000) intitulado “*Solving the inverse dynamics of a Stewart-Gough manipulator by the principle of Virtual Work*”. Nas figuras seguintes encontram-se os resultados obtidos pela minha implementação e pela implementação de Tsai, respectivamente, utilizando as duas trajetórias propostas por ele em seu artigo.

Figura 22 – Resultado da replicação da primeira simulação de Tsai



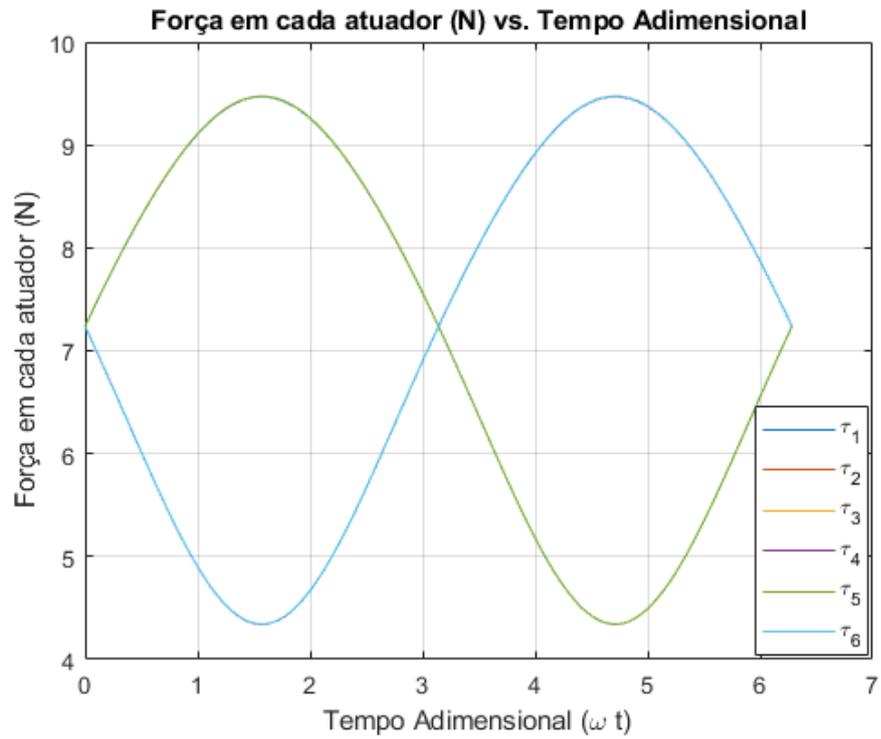
Fonte: Autor.

Figura 23 - Resultado da primeira simulação realizada por Tsai



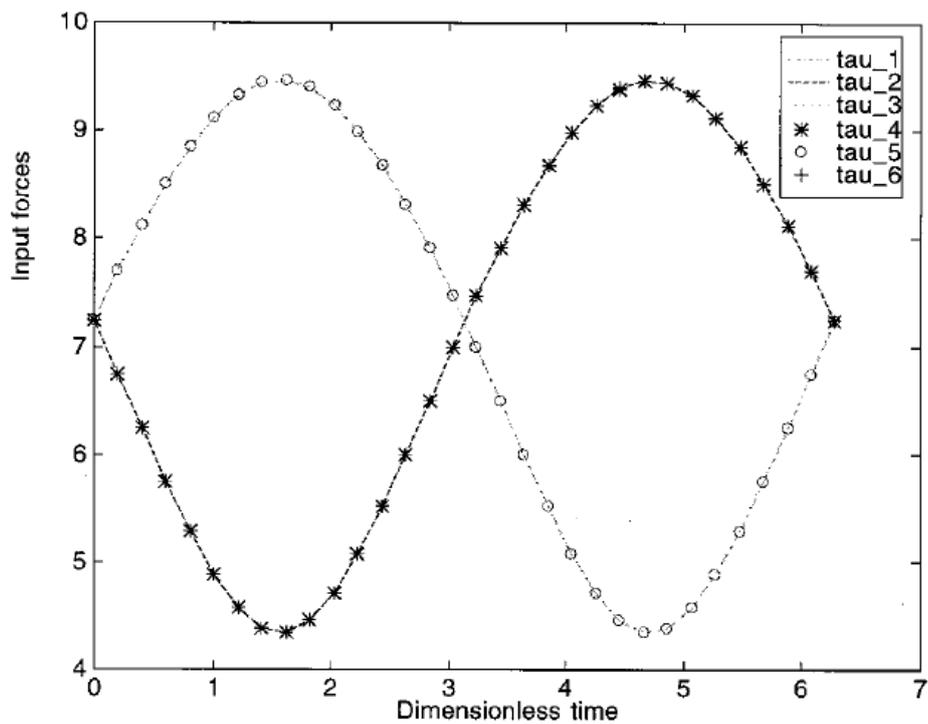
Fonte: (TSAI, 2000, p. 7).

Figura 24 - Resultado da replicação da segunda simulação de Tsai



Fonte: Autor.

Figura 25 - Resultado da segunda simulação realizada por Tsai



Fonte: (TSAI, 2000, p. 7).

Realizando uma comparação entre os resultados de Tsai e dos resultados obtidos pela implementação realizada neste trabalho, percebe-se que os mesmos são, praticamente, idênticos. Essa comparação de resultados e replicação bem sucedida do algoritmo proposto por Tsai demonstra a validação da implementação proposta neste trabalho e da correta modelagem dinâmica do manipulador.

4.4 SIMULAÇÃO DO CONTROLE DO MANIPULADOR

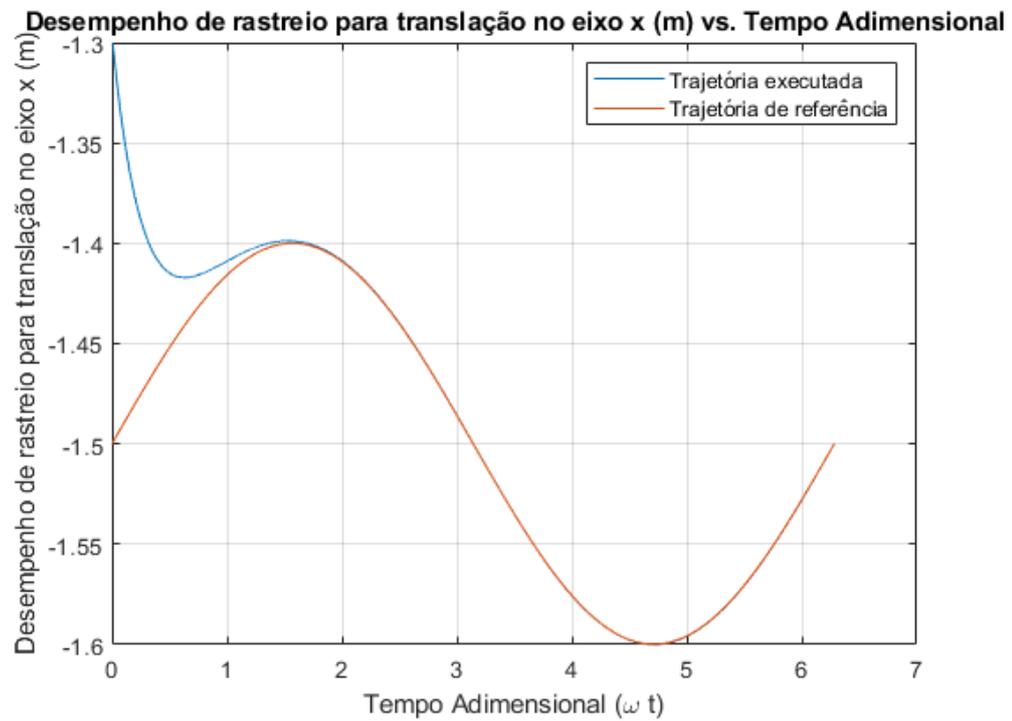
Por fim, foi realizada a simulação do algoritmo de controle proposto para a trajetória explicitada – com a finalidade da validação do sistema de controle e modelo dinâmico da planta. Como descrito anteriormente, foi utilizada a função *ode45* para realizar a solução do sistema de equações diferenciais demonstrado na Equação (99). Para a utilização dessa função, deve-se, também, passar a condição de valor inicial do sistema, nesse caso a posição e a velocidade iniciais da plataforma móvel. Da mesma maneira como feito por Sumnu, Güzelbey e Çakir (2017), foram utilizados os seguintes valores para a posição e velocidade iniciais da plataforma móvel, respectivamente:

$$\chi(0) = [-1,3 \quad 0,2 \quad 1,2 \quad 0 \quad 0 \quad 0]^T m \quad (104)$$

$$\frac{d\chi}{dt}(0) = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T m/s \quad (105)$$

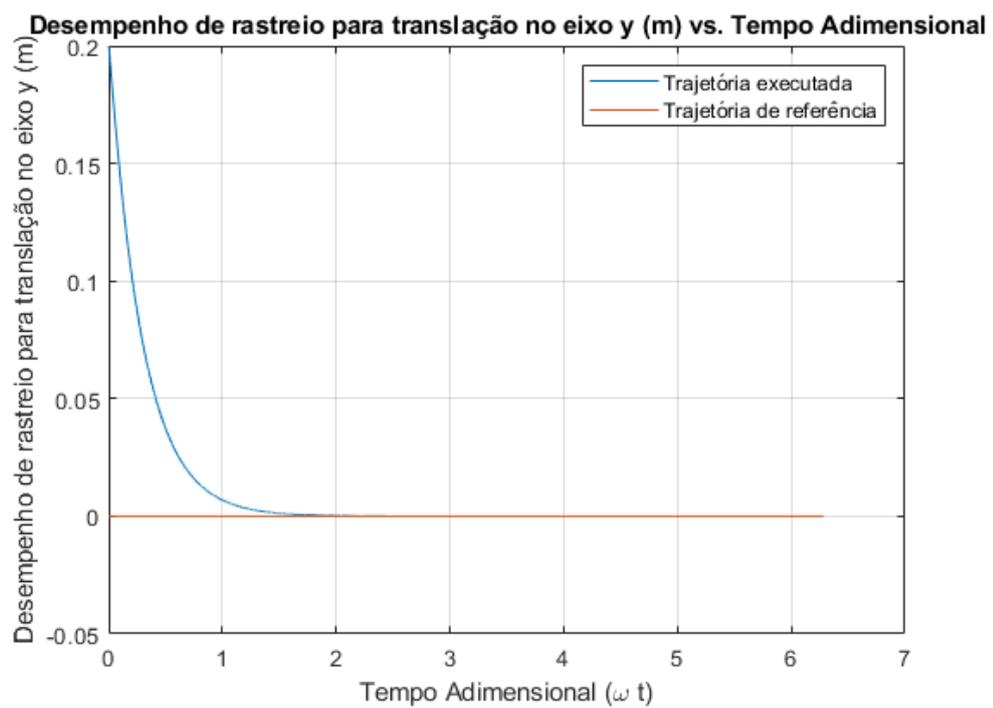
Utilizando-se do algoritmo demonstrado no Capítulo 3, foram obtidos os seguintes resultados (mostrados nas figuras abaixo) para cada componente x, y e z, tanto da parte translacional da trajetória, quanto da parte rotacional.

Figura 26 - Resultado para translação em x



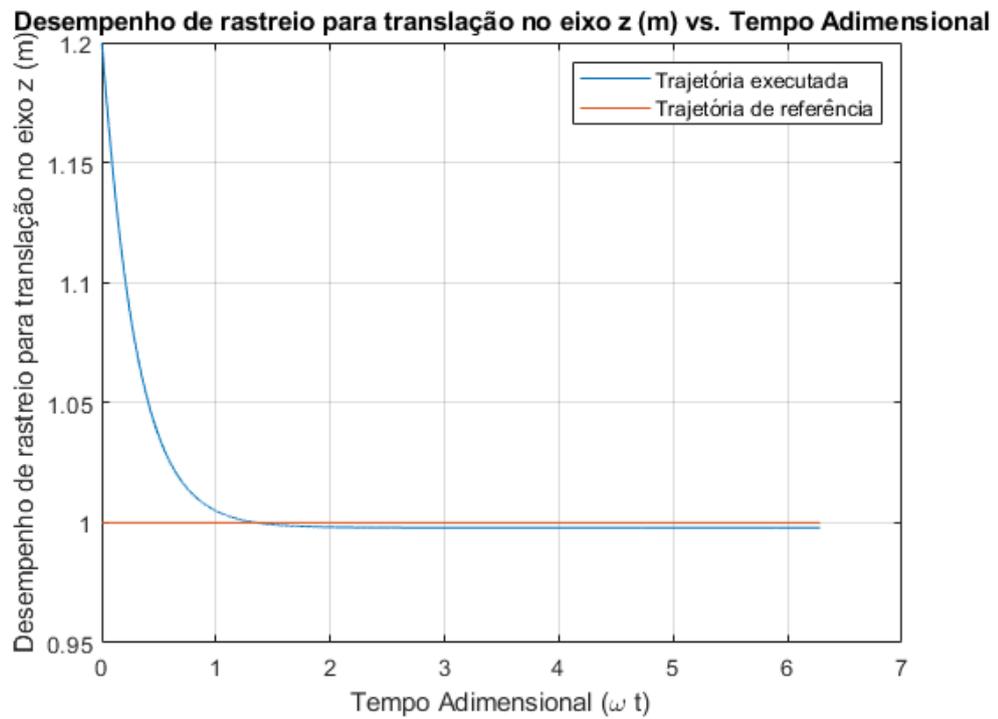
Fonte: Autor.

Figura 27 - Resultado para translação em y



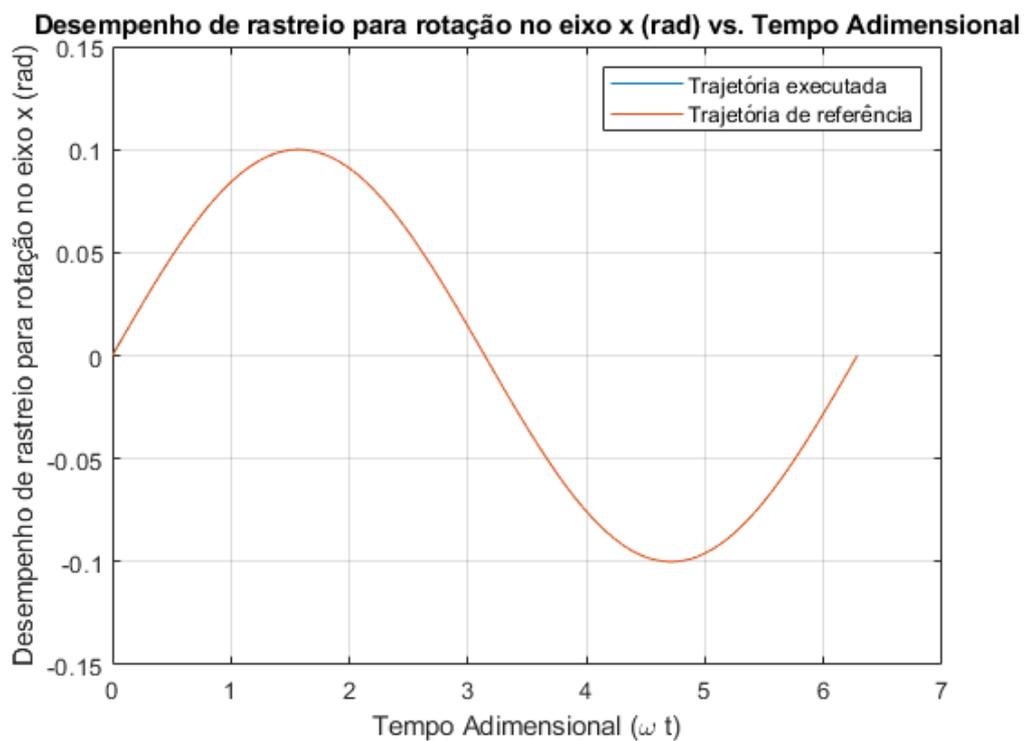
Fonte: Autor.

Figura 28- Resultado para translação em y



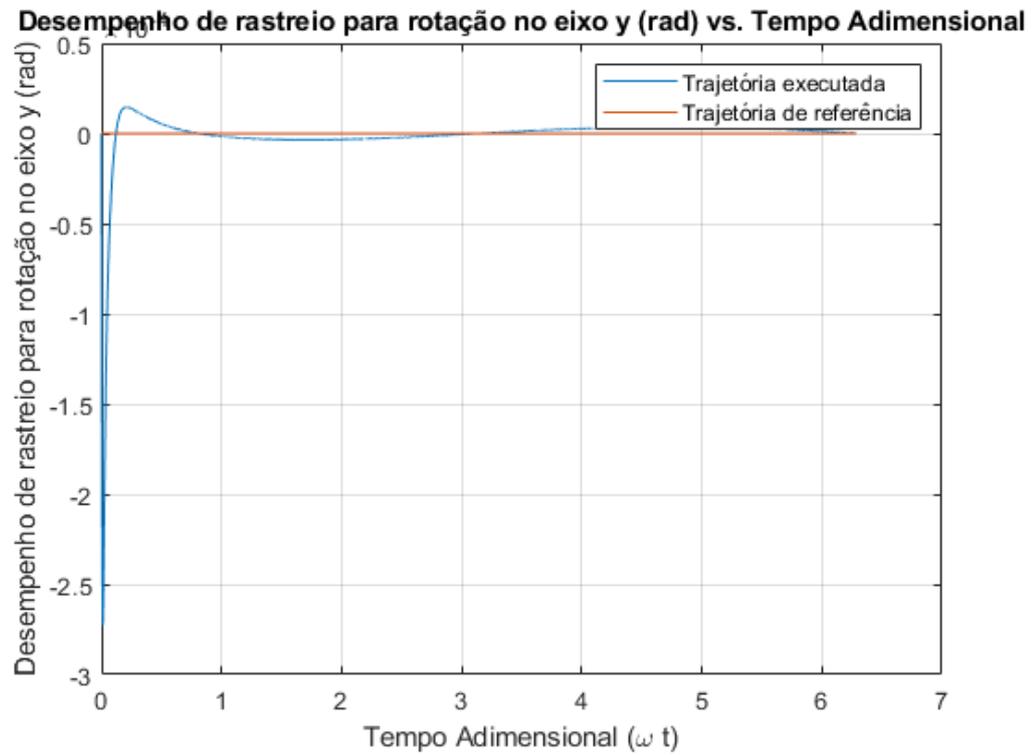
Fonte: Autor.

Figura 29 - Resultado para rotação em x



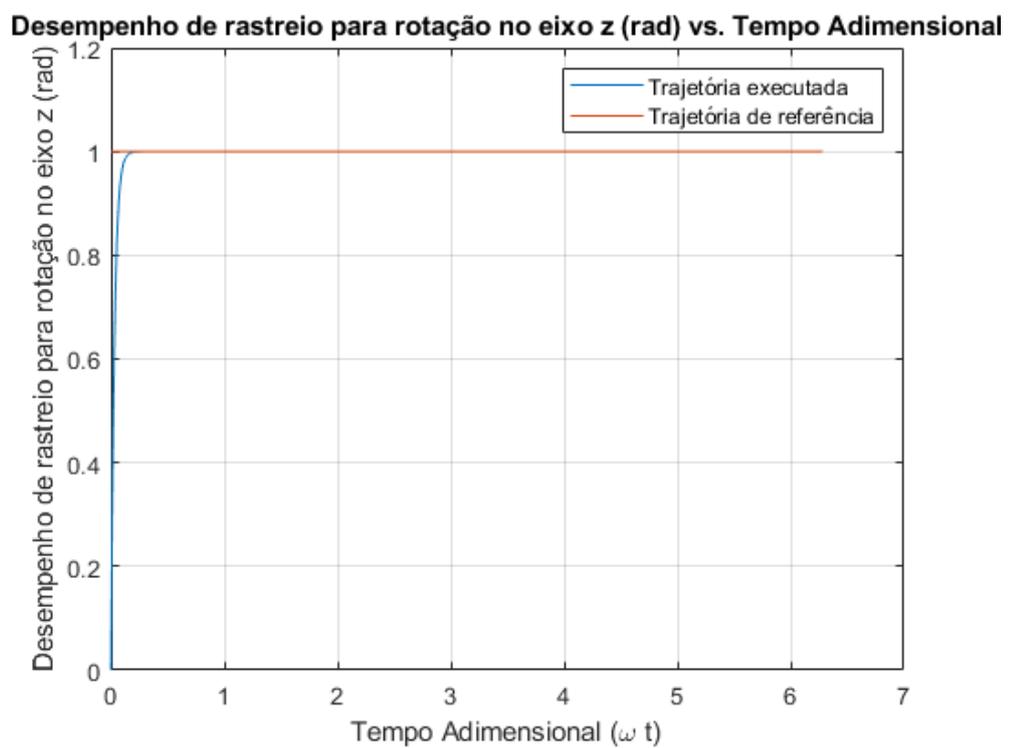
Fonte: Autor.

Figura 30 - Resultado para rotação em y



Fonte: Autor.

Figura 31 - Resultado para rotação em z



Fonte: Autor.

A partir do demonstrado nas figuras 25 a 30, é possível perceber que o sistema de controle apresentou um desempenho aceitável para a trajetória proposta. Nota-se, ainda, que a dinâmica do sistema é relativamente lenta em estado transitório e possui erro de regime permanente no desempenho de rastreamento em todas as trajetórias – apesar de, em algumas, ser muito pequeno. Esses fatores podem ser corrigidos aprimorando o controlador com uma ação Integral – o que eliminaria o erro em regime permanente e traria um melhor desempenho dinâmico do sistema em regime transitório. A aplicação de um controlador PID para a mesma trajetória – porém para um sistema com constantes diferentes - pode ser vista no artigo de Sumnu, Güzelbey e Çakir (2017) intitulado “*Simulation and PID Control of a Stewart Platform with linear motor*”. O desempenho do sistema pode ser medido através das componentes de erro de rastreamento da trajetória, em regime permanente. A Tabela 3 mostra as componentes de erro de rastreamento em regime permanente para a cada componente da trajetória – levando em consideração que o sistema está em regime permanente a partir do instante de tempo $\omega t = 1,2$ – conforme o critério também utilizado por Sumnu, Güzelbey e Çakir (2017).

Tabela 3 - Erro de rastreamento em regime permanente

	x	y	z
Erro máximo na translação	2,8 mm	3,4 mm	1,5 mm
Erro máximo na rotação	118,9332 μ rad	273,1690 μ rad	1 μ rad

Fonte: Autor.

5 CONCLUSÃO

Este trabalho possuiu como objetivo a realização da modelagem dinâmica por meio do método do trabalho virtual, controle PD descentralizado e simulação de uma plataforma de Stewart-Gough. A partir do exposto durante o trabalho, é notável a realização da modelagem dinâmica da plataforma de Stewart-Gough por meio do método do trabalho virtual. Ainda, foi projetado e simulado um controlador Proporcional-Derivativo descentralizado com a finalidade de realizar o controle de movimento do manipulador no espaço de tarefa. Pode se concluir, a partir do demonstrado pelos resultados, que os objetivos deste trabalho foram atingidos com sucesso, sendo realizadas todas as etapas estabelecidas.

As principais contribuições deste trabalho foram a obtenção das equações dinâmicas do movimento de forma fechada do manipulador utilizando-se o método do trabalho virtual e a simulação do sistema completo sendo controlado por um controlador Proporcional-Derivativo descentralizado no espaço de tarefa.

Com a finalidade de se melhorar o desempenho do sistema de controle apresentado, trabalhos futuros indicam a utilização de outros tipos de técnicas de controle – tais como controle não linear, controle LQR, controle LQG, controle por dinâmica inversa, controle por linearização parcial por dinâmica inversa e controles robusto e adaptativo, por exemplo. Ainda, é possível realizar estudos realizando uma comparação entre o desempenho computacional dessas diferentes técnicas de controle. O desempenho da simulação da dinâmica do sistema ainda pode ser estudado mais a fundo, da mesma maneira. Além disso, para se obter uma simulação dinâmica e de controle mais precisa do sistema real, em trabalhos futuros, deve-se levar em consideração a dinâmica dos atuadores a serem utilizados – como, por exemplo, atuadores elétricos lineares ou atuadores pneumáticos.

REFERÊNCIAS

- CRAIG, J. **Introduction to robotics: mechanics and control**. Editora Pearson 3ª edição, 2014.
- DASGUBTA, B.; MRUTHYUNJAYA, T.S. **The Stewart platform manipulator: a review**. Mechanism and machine theory, v. 35, pp. 15 – 40, 1998.
- EULER, L. **Nova methodus motum corporum rigidorum determinandi**. Novi Commentarii academiae scientiarum Petropolitanae, v. 20, pp. 208-238, 1776.
- GOUGH, V.E.; WHITEHALL, S.G. **Universal tire test machine**. Proceedings of the FISITA Ninth International Technical Congress, pp. 117-137, 1962.
- LEMUS, D.; RODRIGUEZ, C.; YIME, E. **Inverse dynamic explicit formulation of a Stewart-Gough manipulator based on the Virtual Work approach**. European Nonlinear Dynamics Conference, 2011.
- MAMOON, M. **Inverse kinematics and path planning of Stewart platform using crank arm actuator**. International Bhurban Conference of Applied Sciences & Technology (IBCAST), v. 1, pp. 175 - 181, 2014.
- MORETTI, M. **Estudo Dinâmico e Simulação de uma Plataforma de Stewart com Ênfase na Implementação do Sistema de Controle**. Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, Campinas, 2010.
- MELLO, C.B. **Controle de Trajetória de uma Plataforma Stewart para a Simulação de Transferência de Carga Fora de Porto**. Instituto Luiz Coimbra, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2011
- MERLET, J.P. **Parallel robots**. Editora Springer Netherlands 2ª edição, 2006.
- NIKU, S. B. **Introdução à Robótica**. Editora LTC 2ª edição, 2019.
- NUSSENZWEIG, M.H. **Curso de física básica – Vol. 1**. Editora Blucher 4ª edição, 2002.

OGATA, K. **Engenharia de Controle Moderno**. Editora Pearson 5^a edição, 2010.

RASTEGARPANAH, A.; SAADAT, M.; RAKHODAEI, H. **Analysis and simulation of various Stewart Platform configurations for lower limb rehabilitation**. BEAR PGR Conference – University of Birmingham, UK, v. 1, pp 1 – 8, 2012.

ROSSI, R. R. **Plataforma de Stewart aplicada a um sistema bola-chapa**. 2016. Trabalho de conclusão de curso (Graduação em Engenharia de Controle e Automação) – Universidade de Caxias do Sul, Caxias do Sul, 2016.

STEWART, D. **A platform with six degrees of freedom**. Proceedings of the institution of mechanical engineers, v. 180, pp. 371-386, 1965.

SU, Y.X.; DUAN, B.Y. **Genetic design of kinematically optimal fine-tuning Stewart platform for large spherical radio telescope**. Elsevier Mechatronics Journal, v. 1, pp. 821 – 835, 2001.

SUMNU, A.; GÜZELBEY, I.H; ÇAKIR, M.V. **Simulation and PID control of a Stewart platform with linear motor**. Journal of Mechanical Science and Technology, v. 31, pp. 345-356, 2017.

TAGHIRAD, H. **Parallel robots: mechanics and control**. Editora CRC Press, 1^a edição, 2013.

TSAI, L.W. **Robot analysis: The Mechanics of Serial and Parallel manipulators**. Editora John Wiley & Sons 1^a edição, 1999.

TSAI, L.W. **Solving the Inverse Dynamics of a Stewart-Gough Manipulator by the Principle of Virtual Work**. Journal of Mechanical Design, v. 122, pp. 3-9, 2000.

VALENTE, V.T. **Análise Matemática e Simulação de Um Manipulador Paralelo com Seis Graus de Liberdade**. Escola de Engenharia, UFRGS, Porto Alegre, 2012.

ANEXO A – CÓDIGOS

A.1 CÓDIGO PRINCIPAL

```

%-----%
%           UFSM - Universidade Federal de Santa Maria           %
%           Curso de Engenharia de Controle e Automação         %
%
%   Programador:                                               %
%   Davi Klein                                                %
%
%   Versão: 1.0                                               30/10/2020 %
%===== %
%           Descrição do Programa                             %
%===== %
%   Arquivo principal para execucao de simulacao de trajetorias %
%   com a cinematica inversa de uma SGP 6-6 SPS.              %
%
%   v1.0 - Versão inicial.                                     %
%-----%
close all           % Fecha todos os gráficos
clc                % Limpa a tela
format short eng   % Formato para exibição
numérica

%-----%
% Parametros de simulacao                                     %
%-----%

step = 0.01; % passo da simulacao
Ts = 2.5; % tempo de simulacao
N = round(Ts/step); % numero de iteracoes
w = 3.0;
to = 0;
syms to;
global e
e = zeros(6,1);

global aux
aux = 0;

%-----%
% Caracteristicas construtivas/ Parâmetros de Projeto       %
%-----%

% Caracteristicas físicas do atuador linear

ex_max = 0.305; % extensao maxima do atuador em m - PREENCHIDO PELO USUARIO
ex_min = 0.205; % extensao minima do atuador em m - PREENCHIDO PELO USUARIO
e1 = 0.5; % distancia da junta Ai até o centro de massa do cilindro em m
e2 = 0.5; % distancia da junta Bi até o centro de massa do pistão em m
m1 = [0.1,0.1,0.1,0.1,0.1,0.1]; % massa do cilindro de cada elo em Kg
m2 = [0.1,0.1,0.1,0.1,0.1,0.1]; % massa do pistão de cada elo em Kg
I1ii = zeros(3,3,6);
I2ii = zeros(3,3,6);

for i = 1:6

```

```

I1ii(:, :, i) = [0.00625 0 0; 0 0.00625 0; 0 0 0];
I2ii(:, :, i) = [0.00625 0 0; 0 0.00625 0; 0 0 0];
end

% Vetores posicao das juntas da plataforma fixa (ai)

a = zeros(3,6); % matriz que armazena a posicao das juntas da plataforma
fixa
a_abs = 2;%85.1285; % modulo do vetor posicao das juntas em m - PREENCHIDO
PELO USUARIO

alfa_o = 0;%-120; % angulo entre a junta inicial e o eixo x do sistema de
referencia em graus - PREENCHIDO PELO USUARIO
aj = 60; % angulo entre as juntas mais proximas - PREENCHIDO PELO USUARIO

alfa = [alfa_o alfa_o+aj alfa_o+120 alfa_o+120+aj alfa_o-120 alfa_o-
120+aj]; % vetor que armazena o angulo alfa, em graus, dos

% vetores posicao das juntas de ambas plataformas

% converte os elementos da matriz alfa de graus para radianos
for i = 1:6
    alfa(i) = deg2rad(alfa(i));
end

% preenche a matriz a com a posicao das juntas da plataforma fixa
for i = 1:3
    for j = 1:6
        if i == 1
            a(i,j) = a_abs*cos(alfa(1,j));
        elseif i == 2
            a(i,j) = a_abs*sin(alfa(1,j));
        else
            a(i,j) = 0;
        end
    end
end

a = [-2.120 -2.380 -2.380 -2.120 0 0; 1.374 1.224 -1.224 -1.374 -0.15 0.15;
0 0 0 0 0 0];
% Vetores posicao das juntas da plataforma movel (bi)

% u = 0.5; % fator de escala de modulo dos vetores ai e bi - PREENCHIDO
PELO USUARIO
% b = u*a; % geracao do vetor b como escala de a

b = [0.170 -0.600 -0.600 0.170 0.430 0.430; 0.595 0.150 -0.150 -0.595 -
0.445 0.445; -0.4 -0.4 -0.4 -0.4 -0.4 -0.4];

%-----%
% Aplicação da Trajetória e Cálculo da Cinemática Inversa %
%-----%

init = [0;0;ex_min]; % valor que descreve a posicao inicial da plataforma
movel
                % em relacao a fixa

% %-----Trajetoria Quadrada-----%
-%

```

```

% L = 150; % lado do quadrado desejado para a trajetoria - PREENCHIDO PELO
USUARIO
%
% [lenq, pointsq] = TrajQuad(init,N,L,a,b); % chama a funcao que calcula a
% % trajetoria quadrada
%
% %verifica se os atuadores dimensionados são capazes de executar a
% %trajetoria desejada
%
% maxlenq = max(lenq);
%
% if max(maxlenq) > ex_max
% disp('Atuador com extensao maxima insuficiente para executar a trajetoria
quadrada');
%
% else
%     disp('Atuador capaz de executar a trajetoria quadrada')
% end
%
% %-----Trajetoria Circular-----
-
%
% rc = 108; % raio desejado para a trajetoria - PREENCHIDO PELO USUARIO
%
% [lenc, pointsc] = TrajCirc(init,N,rc,a,b); % chama a funcao que calcula a
% % trajetoria circular
%
% %verifica se os atuadores dimensionados são capazes de executar a
% %trajetoria desejada
%
% maxlenc = max(lenc);
%
% if max(maxlenc) > ex_max
% disp('Atuador com extensao maxima insuficiente para executar a trajetoria
circular');
%
% else
%     disp('Atuador capaz de executar a trajetoria circular')
% end
%
%-----Trajetoria Helicoidal-----%

rh = 50; % raio desejado para a trajetoria - PREENCHIDO PELO USUARIO
H = 290; % altura maxima da trajetoria - PREENCHIDO PELO USUARIO
tu = 10; % numero de helices da trajetoria - PREENCHIDO PELO USUARIO

[lenh, pointsh] = TrajHelic(init,N,H,rh,tu,a,b);% chama a funcao que
calcula a
% trajetoria helicoidal

%verifica se os atuadores dimensionados são capazes de executar a
%trajetoria desejada

maxlenh = max(lenh);

if max(maxlenh) > ex_max
disp('Atuador com extensao maxima insuficiente para executar a trajetoria
helicoidal');
else

```

```

disp('Atuador capaz de executar a trajetoria helicoidal')
end

```

```

%-----%
% Criação dos gráficos com os resultados obtidos %
%-----%

% grafico de variacao de comprimento do atuador em relacao ao tempo da
% trajetoria quadrada
% l1q = lenq(:,1);
% l2q = lenq(:,2);
% l3q = lenq(:,3);
% l4q = lenq(:,4);
% l5q = lenq(:,5);
% l6q = lenq(:,6);
% time = linspace(0,Ts,N);
%
% figure(1)
% plot(time,l1q,time,l2q,time,l3q,time,l4q,time,l5q,time,l6q)
% title('Variação de comprimento dos atuadores vs. Tempo')
% xlabel('Tempo (s)')
% ylabel('Comprimento dos atuadores (mm)')
% legend('l1','l2','l3','l4','l5','l6')
% grid on
%
% % grafico da trajetoria quadrada
% xq = pointsq(1,:);
% yq = pointsq(2,:);
% zq = zeros(1,N);
% zq(1,:) = init(3,1);
%
%
%
% figure(2)
% plot3(xq,yq,zq,'k')
% title("Trajetória Quadrada desenvolvida pelo manipulador")
% xlabel('mm')
% ylabel('mm')
% zlabel('mm')
% grid on
%
% % grafico de variacao de comprimento do atuador em relacao ao tempo da
% % trajetoria circular
%
% l1c = lenc(:,1);
% l2c = lenc(:,2);
% l3c = lenc(:,3);
% l4c = lenc(:,4);
% l5c = lenc(:,5);
% l6c = lenc(:,6);
%
% figure(3)
% plot(time,l1c,time,l2c,time,l3c,time,l4c,time,l5c,time,l6c)
% title('Variação de comprimento dos atuadores vs. Tempo')
% xlabel('Tempo (s)')
% ylabel('Comprimento dos atuadores (mm)')
% legend('l1','l2','l3','l4','l5','l6')
% grid on
%
%
% % grafico da trajetoria circular

```

```

% xc = pointsc(1,:);
% yc = pointsc(2,:);
% zc = zeros(1,N);
% zc(1,:) = init(3,1);
%
%
% figure(4)
% plot3(xc,yc,zc,'k')
% title("Trajetória Curvilínea Circular desenvolvida pelo manipulador")
% xlabel('mm')
% ylabel('mm')
% zlabel('mm')
% grid on
%
% % grafico de variacao de comprimento do atuador em relacao ao tempo da
% % trajetoria helicoidal
%
% l1h = lenh(:,1);
% l2h = lenh(:,2);
% l3h = lenh(:,3);
% l4h = lenh(:,4);
% l5h = lenh(:,5);
% l6h = lenh(:,6);
% %time2 = linspace(0,Ts,1700);
% figure(5)
% plot(time,l1h,time,l2h,time,l3h,time,l4h,time,l5h,time,l6h)
% title('Variação de comprimento dos atuadores vs. Tempo')
% xlabel('Tempo (s)')
% ylabel('Comprimento dos atuadores (mm)')
% legend('l1','l2','l3','l4','l5','l6')
% grid on
%
% % grafico da trajetoria helicoidal
% xh = pointsh(1,:);
% yh = pointsh(2,:);
% zh = pointsh(3,:);
%
% figure(6)
% plot3(xh,yh,zh,'k')
% title("Trajetória Helicoidal desenvolvida pelo manipulador")
% xlabel('mm')
% ylabel('mm')
% zlabel('mm')
% grid on

%-----%
%                                     Cálculo da Cinemática Direta                                     %
%-----%

x0 = [28; 0; 0.002; 0; 0; 0]; % chute inicial para o cálculo numérico onde
%                               x0 = [P; s], sendo P o vetor posição da
%                               plataforma movel em relacao a fixa e s o
%                               eixo de parafuso que descreve o movimento
%                               da plataforma movel
L = lenh(30,:); % vetor com o comprimento dos atuadores desejável

fk = Cin_dir_v2(L,a,b,x0); % vetor que recebe o calculo da cinematica
direta

```

```

%-----%
%                               %
%                               %
%-----%

%invkinanim(a,b,pointsc,N,init)

%-----%
%                               %
%                               %
%-----%

%----- Parâmetros físicos e de simulação -----%
mp = 1.5; %Kg
Ipb = [0.08 0 0; 0 0.08 0; 0 0 0.08]; %kg.m^2

L = zeros(1,6);
s = zeros(3,6,N);
sx = zeros(3,3,6,N);
su = zeros(3,6,N);

P = [-1.5+(0.1*sin(w*to)); 0;1.0];%[%[-1.5;0;1.0]; [-1.5+(0.1*sin(w*to));
0;1.0]; [-1.5+(0.2*sin(w*to)); 0.2*sin(w*to);1.0+(0.2*sin(w*to))]];

dP = diff(P);
ddP = diff(dP);
sdP = size(dP);
sddP = size(ddP);

if sdP(1) < 3
    dP = zeros(3,N);
end

if sddP(1) < 3
    ddP = zeros(3,N);
end

if sddP(2) < N
    for i = 1:N
        ddP(:,i) = ddP(:,1);
    end
end
ea = [0.1*sin(w*to);0;1];% %zeros(3,N); [0;0;0.35*sin(w*to)];
dea = diff(ea);
ddea = diff(dea);
sdea = size(dea);
sddea = size(ddea);

if sdea(1) < 3
    dea = zeros(3,N);
end

if sddea(1) < 3
    ddea = zeros(3,N);
end

to = linspace(0,2.0944,N);%2.0944

P = subs(P);
dP = subs(dP);
ddP = subs(ddP);

```

```

ea = subs(ea);
dea = subs(dea);
ddea = subs(ddea);

X = [P;ea];
dX = [dP;dea];

% Matriz de rotacao que representa a orientacao da plataforma movel em
% relacao a plataforma fixa na convencao de angulos de euler Z-Y-X

fe = zeros(3,1);
ne = zeros(3,1);

RBA = zeros(3,3,N);
tau = zeros(6,N);
taune = zeros(6,N);
RCA = zeros(3,6,6,N);

dv1J = zeros(6,6,N);
dv1 = zeros(3,6,N);

wt = w*to;

for i = 1:N
%     RBA(:, :, i) = [cos(ea(1,i))*cos(ea(2,i))
cos(ea(1,i))*sin(ea(2,i))*sin(ea(3,i))-sin(ea(1,i))*cos(ea(3,i))
cos(ea(1,i))*sin(ea(2,i))*cos(ea(3,i))+sin(ea(1,i))*sin(ea(3,i))];...
%
%         sin(ea(1,i))*cos(ea(2,i))
sin(ea(1,i))*sin(ea(2,i))*sin(ea(3,i))+cos(ea(1,i))*cos(ea(3,i))
sin(ea(1,i))*sin(ea(2,i))*cos(ea(3,i))-cos(ea(1,i))*sin(ea(3,i));...
%
%         -sin(ea(2,i)) cos(ea(2,i))*sin(ea(3,i))
cos(ea(2,i))*cos(ea(3,i))];
%     RBA(:, :, i) = [cos(ea(3,i))*cos(ea(2,i))
cos(ea(3,i))*sin(ea(2,i))*sin(ea(1,i))-sin(ea(3,i))*cos(ea(1,i))
cos(ea(3,i))*sin(ea(2,i))*cos(ea(1,i))+sin(ea(3,i))*sin(ea(1,i))];...
%
%         sin(ea(3,i))*cos(ea(2,i))
sin(ea(3,i))*sin(ea(2,i))*sin(ea(1,i))+cos(ea(3,i))*cos(ea(1,i))
sin(ea(3,i))*sin(ea(2,i))*cos(ea(1,i))-cos(ea(3,i))*sin(ea(1,i));...
%
%         -sin(ea(2,i)) cos(ea(2,i))*sin(ea(1,i))
cos(ea(2,i))*cos(ea(1,i))];

[L(i,:),s(:, :, i)] = Cin_Inv(P(:,i),a,b,RBA(:, :, i));

end

% for i = 1:N
%     disp(i)
%     [tau(:,i)] =
Din_Inv(L(i,:),s(:, :, i),RBA(:, :, i),dP(:,i),ddP(:,i),dea(:,i),ddea(:,i),b,e1
,e2,m1,m2,I1ii,I2ii,mp,Ipb,fe,ne);
% end

X0 = [-1.3 ; 0.2 ; 1.2 ; 0 ; 0 ; 0] ;
dX0 = [0 ; 0 ; 0 ; 0 ; 0 ; 0] ;

```

```
[t,X_new] = ode45(@ (t,X_new)
Teste_Controlo(t,X_new,N,a,b,I1ii,I2ii,m1,m2,e1,e2,mp,Ipb), [0 2.0944],
[X0, dx0],odeset('OutputFcn','odeplot','OutputSel',[3]));
```

```
global e;
l1 = L(:,1);
l2 = L(:,2);
l3 = L(:,3);
l4 = L(:,4);
l5 = L(:,5);
l6 = L(:,6);
```

```
t1 = tau(1,:);
t2 = tau(2,:);
t3 = tau(3,:);
t4 = tau(4,:);
t5 = tau(5,:);
t6 = tau(6,:);
```

```
tne1 = taune(1,:);
tne2 = taune(2,:);
tne3 = taune(3,:);
tne4 = taune(4,:);
tne5 = taune(5,:);
tne6 = taune(6,:);
```

```
figure(1)
plot(wt,t1,wt,t2,wt,t3,wt,t4,wt,t5,wt,t6)
title('Força em cada atuador (N) vs. Tempo Adimensional')
xlabel('Tempo Adimensional (\omega t)')
ylabel('Força em cada atuador (N)')
legend('\tau_1','\tau_2','\tau_3','\tau_4','\tau_5','\tau_6')
grid on
```

```
figure(2)
plot(wt,tne1,wt,tne2,wt,tne3,wt,tne4,wt,tne5,wt,tne6)
title('Força em cada atuador (N) vs. Tempo Adimensional')
xlabel('Tempo Adimensional (\omega t)')
ylabel('Força em cada atuador (N)')
legend('\tau_1','\tau_2','\tau_3','\tau_4','\tau_5','\tau_6')
grid on
```

```
figure(3)
plot(w*t,X_new(:,1),wt,P(1,:))
grid on
title('Desempenho de rastreo para translação no eixo x (m) vs. Tempo Adimensional')
xlabel('Tempo Adimensional (\omega t)')
ylabel('Desempenho de rastreo para translação no eixo x (m)')
legend('Trajetória executada', 'Trajetória de referência')
```

```
figure(4)
plot(w*t,X_new(:,2),wt,P(2,:))
grid on
```

```

title('Desempenho de rastreo para translação no eixo y (m) vs. Tempo
Adimensional')
xlabel('Tempo Adimensional (\omega t)')
ylabel('Desempenho de rastreo para translação no eixo y (m)')
legend('Trajetória executada', 'Trajetória de referência')

```

```

figure(5)
plot(w*t,X_new(:,3),wt,P(3,:))
grid on
title('Desempenho de rastreo para translação no eixo z (m) vs. Tempo
Adimensional')
xlabel('Tempo Adimensional (\omega t)')
ylabel('Desempenho de rastreo para translação no eixo z (m)')
legend('Trajetória executada', 'Trajetória de referência')

```

```

figure(6)
plot(w*t,X_new(:,4),wt,ea(1,:))
grid on
title('Desempenho de rastreo para rotação no eixo x (rad) vs. Tempo
Adimensional')
xlabel('Tempo Adimensional (\omega t)')
ylabel('Desempenho de rastreo para rotação no eixo x (rad)')
legend('Trajetória executada', 'Trajetória de referência')

```

```

figure(7)
plot(w*t,X_new(:,5),wt,ea(2,:))
grid on
title('Desempenho de rastreo para rotação no eixo y (rad) vs. Tempo
Adimensional')
xlabel('Tempo Adimensional (\omega t)')
ylabel('Desempenho de rastreo para rotação no eixo y (rad)')
legend('Trajetória executada', 'Trajetória de referência')

```

```

figure(8)
plot(w*t,X_new(:,6),wt,ea(3,:))
grid on
title('Desempenho de rastreo para rotação no eixo z (rad) vs. Tempo
Adimensional')
xlabel('Tempo Adimensional (\omega t)')
ylabel('Desempenho de rastreo para rotação no eixo z (rad)')
legend('Trajetória executada', 'Trajetória de referência')

```

```

figure(8)
plot(wt,l1,wt,l2,wt,l3,wt,l4,wt,l5,wt,l6)
grid on
title('Comprimento de cada atuador (m) vs. Tempo Adimensional')
xlabel('Tempo Adimensional (\omega t)')
ylabel('Comprimento de cada atuador (m)')
legend('l1','l2','l3','l4','l5','l6')

```

```

figure(9)
plot(wt,P(1,:))
grid on
title('Translação no eixo X da trajetória de referência')
xlabel('Tempo Adimensional (\omega t)')
ylabel('Eixo X(m)')

```

```

figure(10)
plot(to,P(2,:))
grid on

```

```

title('Translação no eixo Y da trajetória de referência')
xlabel('Tempo Adimensional (\omega t)')
ylabel('Eixo Y(m)')

figure(11)
plot(to,P(3,:))
grid on
title('Translação no eixo Z da trajetória de referência')
xlabel('Tempo Adimensional (\omega t)')
ylabel('Eixo Z (m)')

figure(12)
plot(wt,ea(1,:))
grid on
title('Rotação no eixo X da trajetória de referência')
xlabel('Tempo Adimensional (\omega t)')
ylabel('Eixo X (rad)')

figure(13)
plot(to,ea(2,:))
grid on
title('Rotação no eixo Y da trajetória de referência')
xlabel('Tempo Adimensional (\omega t)')
ylabel('Eixo Y (rad)')

figure(13)
plot(to,ea(3,:))
grid on
title('Rotação no eixo Z da trajetória de referência')
xlabel('Tempo Adimensional (\omega t)')
ylabel('Eixo Z (rad)')

```

A.2 CÁLCULO DA CINEMÁTICA INVERSA

```

%-----%
%           UFSM - Universidade Federal de Santa Maria           %
%           Curso de Engenharia de Controle e Automação           %
%
%   Programador:                                                 %
%   Davi Klein                                                    %
%
%   Versão: 1.0                                                  30/10/2020 %
%===== %
%           Descrição do Programa                               %
%===== %
%   Arquivo de funcao responsavel pelo calculo da cinematica inversa %
%   de uma SGP 6-6p SPS.                                         %
%
%   v1.0 - Versão inicial.                                       %
%-----%

function [l,s] = Cin_Inv(P,a,b,RBA)% definicao da funcao onde P é
% o vetor posicao da
% plataforma movel
% em relacao a fixa, phi,
% theta e psi sao angulos de
% euler, em graus, que

```

```

% descrevem a rotacao da
% plataforma e a e b sao os
% vetores que armazenam a
% posicao das juntas da
% plataforma fixa e movel.

% Angulos de Euler que descrevem a orientacao da plataforma, em
radianos

% phi = deg2rad(Phi); %angulo de rotacao em torno de Z
% psi = deg2rad(Psi); %angulo de rotacao em torno de Y
% theta = deg2rad(Theta); %angulo de rotacao em torno de X

%RBA = [cos(phi)*cos(theta)*cos(psi)-sin(phi)*sin(psi) -
cos(phi)*cos(theta)*sin(psi)-sin(phi)*cos(psi) cos(phi)*sin(theta);...
% sin(phi)*cos(theta)*cos(psi)+cos(phi)*sin(psi) -
sin(phi)*cos(theta)*sin(psi)+cos(phi)*cos(psi) sin(phi)*sin(theta);...
% -sin(phi)*cos(psi) sin(theta)*sin(psi) cos(theta)];

%-----%
% Calculo da cinematica inversa %
%-----%

L = zeros(1,6); % matriz que armazena os valores de comprimento de cada
atuador
S = zeros(3,6); % matriz que armazena os valores de posição do vetor S,
o qual da a direção e sentido aos atuadores

RBAb = RBA*b; % transformacao linear que converte a matriz b em relacao
ao sistema de referencia A

% Iteracoes para calculo do comprimento das juntas e das coordenadas do
vetor posição de cada elo
for i = 1:6
    L(i) = norm(P + RBAb(:,i) - a(:,i)); % sqrt(P'*P + b(:,i)'*b(:,i) +
a(:,i)'*a(:,i) - 2*P'*a(:,i) + 2*P'*RBAb(:,i) - 2*RBAb(:,i)'*a(:,i));
    S(:,i) = (P + RBAb(:,i) - a(:,i))/L(i);
end

l = L; % retorno da funcao
s = S;

end

```

A.3 CÁLCULO DA CINEMÁTICA DIRETA

```

%-----%
% UFSM - Universidade Federal de Santa Maria %
% Curso de Engenharia de Controle e Automação %
% %
% Programador: %
% Davi Klein %

```



```

%
%
%   v1.0 - Versão inicial.
%-----%

function din_mat =
Closed_Form_Din(dx,bb,L,s,m1,m2,e1,e2,mp,Ipb,I1ii,I2ii,RBA)

b = RBA*bb;
Ip = RBA*Ipb*RBA';

%----- Análise de velocidades -----%

RCA = zeros(3,3,6); % matriz de rotação que transforma o sistema de referen
% cia C em A

cti = zeros(1,6);
sti = zeros(1,6);
spi = zeros(1,6);
cpi = zeros(1,6);

for i = 1:6
    cti(i) = s(3,i);
    sti(i) = sqrt((s(1,i)^2)+(s(2,i)^2));
    spi(i) = s(2,i)/sti(i);
    cpi(i) = s(1,i)/sti(i);

    RCA(:,:,i) = [cpi(i)*cti(i) -spi(i) cpi(i)*sti(i);...
                  spi(i)*cti(i) cpi(i) spi(i)*sti(i);...
                  -sti(i) 0 cti(i)];
end

vbi = zeros(3,6); % vetor que armazena os valores de velocidade de cada
junta Bi em relação a A
vbii = zeros(3,6); % vetor que armazena os valores de velocidade de cada
junta Bi em relação a C
wbii = zeros(3,6); % vetor que armazena os valores de velocidade angular de
cada junta Bi em relação a C
vp = dx(1:3); % velocidade linear do ponto P (velocidade linear da
plataforma movel)
wp = dx(4:6); % velocidade angular do ponto P (velocidade angular da
plataforma movel)
wbi = zeros(3,6);

for i = 1:6
    vbi(:,i) = vp + (cross(wp,b(:,i)));
    vbii(:,i) = RCA(:,:,i)'*vbi(:,i);
end

for i = 1:6
    wbii(:,i) = ((1/L(i))*[-vbii(2,i); vbii(1,i); 0]);
    wbi(:,i) = RCA(:,:,i)*wbii(:,i);
end

%----- Definições das matrizes Jacobianas dos -----%
%----- elos e da plataforma -----%

```

```

Jbi = zeros(3,6,6);
Jbii = zeros(3,6,6);
din_mat.Jp = zeros(6,6);

for i = 1:6
    Jbi(:,:,i) = [1 0 0 0 b(3,i) -b(2,i); ...
                  0 1 0 -b(3,i) 0 b(1,i); ...
                  0 0 1 b(2,i) -b(1,i) 0];
    Jbii(:,:,i) = RCA(:,:,i)'*Jbi(:,:,i);

    din_mat.Jp(i,:) = Jbii(3,:,i);
end

% ----- Closed form dynamics of the limbs ----- %
mce = zeros(1,6);
mco = zeros(1,6);
mge = zeros(1,6);
Ieq = I1ii + I2ii;
s_hat = zeros(3,6);
sx = zeros(3,3,6);
s_hatx = zeros(3,3,6);
dL = zeros(1,6);
g = [0;0;-9.807];

M1 = zeros(6,6,6);
C1 = zeros(6,6,6);
G1 = zeros(6,1,6);

Mp = [mp*eye(3) zeros(3); zeros(3) Ip];
Cp = [zeros(3) zeros(3); zeros(3) skws(wp)*Ip];
Gp = [-mp*g; zeros(3,1)];

Ji = zeros(3,6,6);
dJi = zeros(3,6,6);

M = zeros(3,3,6);
C = zeros(3,3,6);
G = zeros(3,1,6);

sumM = zeros(6,6);
sumC = zeros(6,6);
sumG = zeros(6,1);

for i = 1:6
    mce(i) = (1/(L(i)^2))*(m1(i)*(e1^2)+(m2(i)*(e2^2)));
    mco(i) = ((1/(L(i))))*m2(i)*e2 -
    (i/(L(i)^2))*(Ieq(1,1,i)+(L(i)^2)*mce(i));
    mge(i) = (1/L(i))*((m1(i)*e1)+m2(i)*(L(i)-e2));

    s_hat(:,i) = s(:,i)/(norm(s(:,i)));
    sx(:,:,i) = skws(s(:,i));
    s_hatx(:,:,i) = skws(s_hat(:,i));

    dL(i)=s_hat(:,i) '*vp;

    M(:,:,i) = (m2(i)*s_hat(:,i)*s_hat(:,i)')-
    (1/(L(i)^2)*Ieq(1,1,i)*(s_hatx(:,:,i))^2)-mce(i)*((s_hatx(:,:,i))^2);

```

```

    C(:,:,i) = -((2/L(i))*mco(i)*dL(i)*(s_hatx(:,:,i)^2))-
    ((1/(L(i)^2))*m2(i)*e2*s_hat(:,i)*vp'*(s_hatx(:,:,i)^2));
    G(:,:,i) = ((mge(i)*(s_hatx(:,:,i)^2))-
    (m2(i)*s_hat(:,i)*s_hat(:,i)'))*g;

    Ji(:,:,i) = [eye(3) -skws(b(:,i))];
    dJi(:,:,i) = [zeros(3) -skws(wp)*skws(b(:,i))+skws(b(:,i))*skws(wp)];

    Ml(:,:,i) = Ji(:,:,i)'*M(:,:,i)*Ji(:,:,i);
    Cl(:,:,i) = Ji(:,:,i)'*M(:,:,i)*dJi(:,:,i) +
    Ji(:,:,i)'*C(:,:,i)*Ji(:,:,i);
    Gl(:,:,i) = Ji(:,:,i)'*G(:,:,i);

    sumM = sumM + Ml(:,:,i);
    sumC = sumC + Cl(:,:,i);
    sumG = sumG + Gl(:,:,i);
end

din_mat.MX = Mp + sumM;
din_mat.CX = Cp + sumC;
din_mat.GX = Gp + sumG;

end

```

A.5 CÓDIGO PRINCIPAL DO CONTROLE

```

function custo =
Teste_Controlo(t,X_new,N,a,b,I1ii,I2ii,m1,m2,e1,e2,mp,Ipb)
%disp(t)

x = X_new(1:6);
dx = X_new(7:12);

w = 3.0;

syms ti;

P = [-1.5+(0.1*sin(w*ti)); 0;1.0];%[-1.5;0;1.0];
dP = diff(P);
sdP = size(dP);

if sdP(1) < 3
    dP = zeros(3,1);
end

ea = [0.1*sin(w*ti);0;1];
dea = diff(ea);
sdea = size(dea);

if sdea(1) < 3
    dea = zeros(3,1);
end

ti = t;

```

```

P = subs(P);
dP = subs(dP);

ea = subs(ea);
dea = subs(dea);

X = [P;ea];
dX = [dP;dea];

X = double(X);
dX = double(dX);

RBA = [cos(x(6))*cos(x(5)) cos(x(6))*sin(x(5))*sin(x(4)) -
sin(x(6))*cos(x(4)) cos(x(6))*sin(x(5))*cos(x(4))+sin(x(6))*sin(x(4));...
      sin(x(6))*cos(x(5))
sin(x(6))*sin(x(5))*sin(x(4))+cos(x(6))*cos(x(4))
sin(x(6))*sin(x(5))*cos(x(4))-cos(x(6))*sin(x(4));...
      -sin(x(5)) cos(x(5))*sin(x(4)) cos(x(5))*cos(x(4))];

[L,s] = Cin_Inv(x(1:3),a,b,RBA);

[din_mat] = Closed_Form_Din(dx,b,L,s,m1,m2,e1,e2,mp,Ipb,I1ii,I2ii,RBA);
global aux;
aux = aux + 1;
[din_mat]= Controle(x,dx,X,dX,din_mat);

custo = [dx; pinv(din_mat.MX)*(din_mat.F-din_mat.CX*dx-din_mat.GX)];

```

end

A.6 CÓDIGO QUE CALCULA A AÇÃO DE CONTROLE

```

%-----%
%          UFSM - Universidade Federal de Santa Maria          %
%          Curso de Engenharia de Controle e Automação         %
%
%          Programador:                                       %
%          Davi Klein                                         %
%
%          Versão: 1.0                                         %
%          30/10/2021                                         %
%===== %
%          Descrição do Programa                             %
%===== %
%          Arquivo de funcao responsavel pelo calculo do controle PD %
%          descentralizado do sistema                         %
%
%
%          v1.0 - Versão inicial.                             %
%-----%

function din_mat = Controle(x,dx,X,dX,din_mat)
Kp = 1e4*diag([1 1 1 10 10 10]);
Kv = 1e3*diag([1 1 1 1 1 1]);

```

```
din_mat.FJ = zeros(6,10);
din_mat.PID = zeros(6,1);
din_mat.FD = [0;0;0;0;0;0];
global e
global aux
e_aux = X-x;

if aux == 1
    e = e_aux;
else
    e = [e , e_aux];
end

din_mat.PID = Kp*(X-x)+Kv*(dX-dx);
din_mat.FJ = pinv(din_mat.Jp')*din_mat.PID;
din_mat.F = din_mat.FD + din_mat.PID;
end
```