



Universidade Federal da Bahia  
Instituto de Matemática / Escola Politécnica

Programa de Pós-Graduação em Mecatrônica

ANÁLISE DA EFICIÊNCIA COMPUTACIONAL PARA SOLUÇÃO  
DO PROBLEMA DA CINEMÁTICA INVERSA DE ROBÔS  
ANTROPOMÔRFICOS UTILIZANDO A TEORIA DE BASES DE  
GRÖBNER.

Sérgio Ricardo Xavier da Silva

TESE DE DOUTORADO

Salvador

13 de julho de 2020

SÉRGIO RICARDO XAVIER DA SILVA

ANÁLISE DA EFICIÊNCIA COMPUTACIONAL PARA SOLUÇÃO  
DO PROBLEMA DA CINEMÁTICA INVERSA DE ROBÔS  
ANTROPOMÓRFICOS UTILIZANDO A TEORIA DE BASES DE  
GRÖBNER.

Esta Tese de Doutorado foi apresentada ao Programa de Pós-Graduação em Mecatrônica da Universidade Federal da Bahia, como requisito parcial para obtenção do grau de Doutor em Mecatrônica.

Orientador: Prof. Dr. Leizer Schnitman.

Coorientador: Prof. Dr. Vitalino Cesca Filho.

Salvador

13 de julho de 2020

- 
- S586 Silva, Sérgio Ricardo Xavier da Silva.  
Análise da eficiência computacional para solução do problema da cinemática inversa de robôs antropomórficos utilizando a teoria de bases de Gröbner/ Sérgio Ricardo Xavier da Silva. – Salvador, 2020.  
262 f.: il. color.
- Orientador: Prof. Dr. Leizer Schnitman.  
Coorientador: Prof. Dr. Vitalino Cesca Filho.  
Tese (doutorado) – Universidade Federal da Bahia. Escola Politécnica, 2020.
1. Manipuladores robóticos seriais. 2. Cinemática. 3. Gröbner, bases de. 4. Método de Paul. I. Schnitman, Leizer. II. Cesca Filho, Vitalino. III. Universidade Federal da Bahia. IV. Título.

CDD: 629.89

---

## TERMO DE APROVAÇÃO

SÉRGIO RICARDO XAVIER DA SILVA

### ANÁLISE DA EFICIÊNCIA COMPUTACIONAL PARA SOLUÇÃO DO PROBLEMA DA CINEMÁTICA INVERSA DE ROBÔS ANTROPOMÓRFICOS UTILIZANDO A TEORIA DE BASES DE GRÖBNER.

Esta Tese de Doutorado foi julgada adequada à obtenção do título de Doutor em Mecatrônica e aprovada em sua forma final pelo Programa de Pós-Graduação em Mecatrônica da Universidade Federal da Bahia.

Salvador, 13 de julho de 2020



---

Prof. Dr. Leizer Schnitman  
Orientador - Universidade Federal da Bahia




---

Prof. Dr. Vitalino Cesca Filho  
Coorientador - Universidade Federal do Pampa



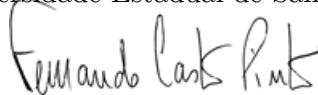
---

Prof. Dr. Alexandre Queiroz Bracarense  
Universidade Federal de Minas Gerais



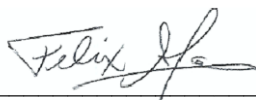
---

Prof. Dr. Fermin de la Caridad Garcia Velasco  
Universidade Estadual de Santa Cruz



---

Prof. Dr. Fernando Augusto de Noronha Castro Pinto  
Universidade Federal do Rio de Janeiro



---

Prof. Dr. Felix Mas Milian  
Universidade Estadual de Santa Cruz

*Ao meu filho, Serginho, que ilumina a minha vida.*

*À minha esposa, Gracielle, pelo seu amor incondicional.*

## AGRADECIMENTOS

À minha amada esposa, Gracielle Mendonça Lins da Silva, que sempre com sua compreensão, incentivo, amor, carinho, estímulo e paciência, foi fundamental para o cumprimento desta jornada.

Ao meu filho, Sérgio Ricardo Xavier da Silva Júnior, que é para mim motivo de orgulho e felicidade.

Aos meus pais, Adilson Carlos da Silva (*in memoriam*) e Hildete Alves Xavier da Silva, por me terem me incentivado e ajudado em todo o meu processo de formação.

Ao meu orientador, Prof. Dr. Leizer Schnitman, pela paciência, ensinamento e contribuição no desenvolvimento deste trabalho.

Ao meu coorientador, Prof. Dr. Vitalino Cesca Filho, pelo incentivo, dedicação e tempo dispensados no criterioso labor de acompanhamento e orientação deste trabalho.

Aos professores, colegas e funcionários do Programa de Pós-Graduação em Mecatrônica da Universidade Federal da Bahia, pelo apoio na realização deste curso.

Ao Departamento de Ciências Exatas e da Terra (DCET), da Universidade do Estado da Bahia (UNEB), pelo apoio concedido durante a realização do Doutorado.

Ao Centro de Armazenamento de Dados e Computação Avançada, da Universidade Estadual de Santa Cruz (UESC).

Não deixo de agradecer também ao imprescindível apoio da CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior).

A todos que direta ou indiretamente me apoiaram para a realização deste trabalho.

*“Saber muito não lhe torna inteligente. A inteligência se traduz na forma que você recolhe, julga, maneja e, sobretudo, onde e como aplica esta informação.”*

*Carl Sagan.*

## RESUMO

O algoritmo de Denavit-Hartenberg é um método utilizado por décadas para solucionar um dos problemas clássicos na cinemática de manipuladores robóticos: o problema cinemático inverso. Quando esse método é utilizado, existe a necessidade de algoritmos adicionais para resolver o problema, como por exemplo, o método de Paul. A Teoria de Bases de Gröbner para solução da cinemática inversa, como método suplementar ao algoritmo de Denavit-Hartenberg, será apresentada neste trabalho. Para favorecer um melhor entendimento do leitor com cada método, os robôs manipuladores Stäubli TS20, um robô do tipo SCARA, e o Unimation PUMA 560, um manipulador antropomórfico com seis juntas rotativas, serão utilizados como estudos de caso aplicando o método de Paul e o método proposto nesta tese, em que os dados de eficiência computacional serão utilizados para comparação. O objetivo principal é analisar a eficiência computacional na solução do problema da cinemática inversa de robôs manipuladores antropomórficos utilizando os dois métodos. Com cada abordagem, o problema da cinemática inversa para os dois robôs seriais será resolvido. Ao comparar cada método, esta pesquisa demonstrará que o método, com base na Teoria de Bases de Gröbner, é computacionalmente mais eficiente para a solução do problema cinemático inverso de robôs antropomórficos.

Palavras-chave: Manipuladores Robóticos Seriais, Cinemática Inversa, Bases de Gröbner, Método de Paul, Eficiência Computacional.



## ABSTRACT

The Denavit-Hartenberg algorithm is a method used for decades to solve one of the classic problems in the kinematics of robotic manipulators, the inverse kinematic problem. When this method is used, there is a need for additional algorithms to solve the problem, such as the Paul method. The Gröbner Bases Theory for the solution of inverse kinematics, as a supplementary method to the Denavit-Hartenberg algorithm, will be presented in this work. To familiarize the reader with each method, the manipulator robots Stäubli TS20, a SCARA type robot, and Unimation PUMA 560, an anthropomorphic manipulator with six rotating joints, will be used as case studies applying the Paul method and the method proposed in this work, where the computational efficiency data will be used for comparison. The main goal of this work is to analyze the computational efficiency in solving the problem of the inverse kinematics of anthropomorphic manipulating robots using both methods. With each approach, the problem of inverse kinematics for the two serial robots will be solved. When comparing each method, this work will demonstrate that the method using Gröbner Bases Theory is more computationally efficient for the solution of the inverse kinematic problem of anthropomorphic robots.

Key-words: Serial Manipulators, Inverse Kinematics, Gröbner Bases Theory, Paul Method, Computational Efficiency.

## LISTA DE FIGURAS

Figura 1: Cinemática direta <i>versus</i> cinemática inversa.....	14
Figura 2: Sistemas de referência deslocados no mesmo espaço tridimensional .....	22
Figura 3: Representação dos parâmetros da notação de Denavit-Hartenberg.....	24
Figura 4: Robô SCARA Stäubli TS20 .....	65
Figura 5: Juntas do robô SCARA Stäubli TS20.....	66
Figura 6: Dimensões do robô SCARA Stäubli TS20.....	67
Figura 7: Sistemas de coordenadas para o robô SCARA Stäubli TS20.....	68
Figura 8: Solução da cinemática direta pelo método geométrico.....	70
Figura 9: Representação dos pontos inatingíveis do SCARA Stäubli TS20 com base nas equações da cinemática inversa.....	78
Figura 10: Determinação de $\theta_3$ pelo método geométrico .....	79
Figura 11: Determinação de $\theta_1$ pelo método geométrico .....	81
Figura 12: Volume de trabalho do robô SCARA Stäubli TS20.....	88
Figura 13: Robô SCARA Stäubli TS20 com elos $L_1$ e $L_3$ colineares .....	89
Figura 14: Ponto no interior do volume de trabalho do robô SCARA Stäubli TS20 .	90
Figura 15: Ponto fora do volume de trabalho do robô SCARA Stäubli TS20.....	93
Figura 16: Resumo das soluções para o robô SCARA Stäubli TS20.....	94
Figura 17: Modelo algébrico do SCARA Stäubli TS20 (vista superior) .....	98
Figura 18: Pontos atingíveis com duas configurações de juntas.....	102
Figura 19: Volume de trabalho do manipulador TS20 por meio da Teoria de Bases de Gröbner .....	103
Figura 20: Configurações distintas para $x = 0$ e $y > 0$ .....	108
Figura 21: Configurações distintas para $x = 0$ e $y < 0$ .....	108
Figura 22: Modelo geométrico do SCARA Stäubli TS40 .....	110
Figura 23: SCARA Stäubli TS40.....	111

Figura 24: Pontos singulares de fronteira para o SCARA Stäubli TS40 .....	112
Figura 25: Sistemas de referência do Puma 560.....	117
Figura 26: Localização dos sistemas de referência intermediários {P}, {Q} e {R} ..	118
Figura 27: Estrutura em árvore das oito configurações cinemáticas inversas do PUMA 560.....	137

## LISTA DE TABELAS

Tabela 1: Parâmetros cinemáticos para juntas rotacionais e prismáticas .....	24
Tabela 2: Parâmetros D-H do robô SCARA Stäubli TS20 .....	68
Tabela 3: Custo computacional das operações mais comuns.....	96
Tabela 4: Parâmetros cinemáticos do Puma 560 .....	119
Tabela 5: Número de operações para solução da cinemática direta do PUMA 560..	121
Tabela 6: Número de operações para solução da cinemática inversa do PUMA 560, ângulo a ângulo, por meio do método de Paul.....	138
Tabela 7: Configurações de juntas do Puma 560 (em radianos) obtidas com base no método de Paul.....	140
Tabela 8: Configurações de juntas (em radianos) do Puma 560 obtidas por meio da Teoria de Bases de Gröbner.....	145
Tabela 9: Configurações de juntas (em graus) do Puma 560 obtidas por meio da Teoria de Bases de Gröbner.....	146
Tabela 10: Dados de desempenho computacional para solução da cinemática inversa do manipulador Puma 560 por meio dos dois métodos .....	147
Tabela 11: Número de operações para solução da cinemática inversa do PUMA 560 por meio do método de Paul – 2 <sup>o</sup> Caso.....	149
Tabela 12: Configurações de juntas do Puma 560 (em radianos) obtidas por meio do método de Paul – 2 <sup>o</sup> Caso .....	151
Tabela 13: Configurações de juntas (em radianos) do Puma 560 obtidas por meio da Teoria de Bases de Gröbner – 2 <sup>o</sup> Caso .....	153
Tabela 14: Configurações de juntas (em graus) do Puma 560 obtidas por meio da Teoria de Bases de Gröbner – 2 <sup>o</sup> Caso.....	154
Tabela 15: Dados de desempenho computacional para solução da cinemática inversa do manipulador Puma 560 por meio dos dois métodos – 2 <sup>o</sup> Caso.....	155

Tabela 16: Comparação do desempenho computacional para solução da cinemática inversa do manipulador Puma 560 utilizando ordens lexicográficas distintas para os dois casos estudados .....	156
Tabela 17: Parâmetros cinemáticos de um manipular 6R com pulso não esférico....	158
Tabela 18: Configurações de juntas (em radianos) de um manipular 6R com pulso não esférico.....	162
Tabela 19: Configurações de juntas (em graus) de um manipular 6R com pulso não esférico.....	162
Tabela 20: Dados de desempenho computacional para um manipular 6R com pulso não esférico.....	163

# SUMÁRIO

1	INTRODUÇÃO .....	13
2	ALGORITMO DE DENAVIT-HARTENBERG .....	21
2.1	CINEMÁTICA DIRETA.....	21
2.2	CINEMÁTICA INVERSA.....	26
2.2.1	Método de Paul.....	27
3	TEORIA DE BASES DE GRÖBNER .....	29
3.1	GRUPOS, ANÉIS, DOMÍNIOS, CORPOS E IDEAIS.....	29
3.2	ANÉIS DE POLINÔMIOS .....	32
3.3	ORDENAÇÃO MONOMIAL.....	33
3.4	ALGORITMO DA DIVISÃO .....	36
3.4.1	Algoritmo da Divisão em $k[x]$ .....	37
3.4.2	Algoritmo da divisão em $k[x_1, \dots, x_n]$ .....	39
3.5	ALGORITMO DE BUCHBERGER.....	52
4	CINEMÁTICA INVERSA DO STÄUBLI TS20 POR MEIO DO MÉTODO DE PAUL.....	65
4.1	CINEMÁTICA DIRETA DO SCARA STÄUBLI TS20.....	67
4.2	CINEMÁTICA INVERSA DO SCARA STÄUBLI TS20.....	71
4.3	CONFIGURAÇÕES SINGULARES .....	84
4.4	EXEMPLOS GENÉRICOS .....	89
4.4.1	Análise dos resultados obtidos.....	94
5	CINEMÁTICA INVERSA DO STÄUBLI TS20 POR MEIO DA TEORIA DE BASES DE GRÖBNER .....	97
5.1	MODELO ALGÉBRICO PARA O ROBÔ SCARA STÄUBLI TS20.....	97
5.2	DETERMINAÇÃO DE UMA BASE DE GRÖBNER .....	100
5.3	CONFIGURAÇÕES GENÉRICAS .....	105
5.3.1	Análise dos resultados obtidos.....	114

6	CINEMÁTICA INVERSA DO PUMA 560 POR MEIO DO MÉTODO DE PAUL.....	116
6.1	CINEMÁTICA DIRETA DO PUMA 560 .....	117
6.2	CINEMÁTICA INVERSA DO PUMA 560 .....	121
7	CINEMÁTICA INVERSA DO PUMA 560 POR MEIO DA TEORIA DE BASES DE GRÖBNER .....	142
7.1	CINEMÁTICA INVERSA DO PUMA 560 (1º CASO).....	143
7.2	CINEMÁTICA INVERSA DO PUMA 560 (2º CASO).....	148
7.3	CASO ESPECIAL: MANIPULADOR 6R COM PULSO NÃO ESFÉRICO .....	158
8	CONCLUSÕES .....	165
	REFERÊNCIAS .....	168
	APÊNDICE A .....	173
	APÊNDICE B .....	184
	APÊNDICE C .....	195
	APÊNDICE D .....	198
	APÊNDICE E.....	206
	APÊNDICE F.....	208
	APÊNDICE G .....	209
	APÊNDICE H .....	217
	APÊNDICE I.....	219
	APÊNDICE J.....	221
	APÊNDICE K .....	228
	APÊNDICE L.....	242
	APÊNDICE M .....	256

# 1 INTRODUÇÃO

Para Ullrich (1987), robôs manipuladores são equipamentos multifuncionais e reprogramáveis, projetados para movimentar materiais, peças, ferramentas ou dispositivos especializados por meio de movimentos variáveis e programados, para a execução de diversas tarefas.

Ainda, segundo o mesmo autor, os robôs podem ser equipados com sensores para sentir ou perceber, por exemplo, calor, pressão e objetos. São capazes também de reagir ao seu ambiente de trabalho, operar outras máquinas e comunicar-se quando ocorrem problemas em seu funcionamento.

De acordo com Craig (2017), um manipulador robótico serial é um sistema de corpos, denominados elos, conectados por meio de juntas. Existem dois tipos de juntas: a rotativa, que permite a rotação de um elo sobre a articulação do elo anterior; e uma junta prismática, que possibilita um movimento de translação entre as articulações. Todos esses conceitos são válidos e utilizados até os dias atuais.

O sistema supramencionado forma uma cadeia cinemática que contém, conectada ao elo final, uma ferramenta ou garra: denominada efetuador. A cinemática de manipuladores robóticos discorre sobre os movimentos do efetuador em relação aos das juntas.

Consoante o autor referido acima, a localização do efetuador final de um manipulador robótico é determinada por intermédio do controle adequado das suas variáveis articulares. Por consequência, os valores do conjunto de variáveis de junta de um robô determinam o posicionamento de seu efetuador final no sistema de coordenadas.



De acordo com Sciavicco e Siciliano (1996), um grau de liberdade, GDL ou DOF (*degree of freedom*), é o número de parâmetros independentes que são necessários para se definir a posição de um corpo no espaço, em qualquer instante.

Dois problemas clássicos na cinemática de manipuladores robóticos são: a cinemática direta e a inversa. Na cinemática direta, a posição e orientação do efetuador são calculados com base nas variáveis das juntas do robô. Na cinemática inversa, os movimentos das juntas individuais são calculados utilizando a posição e orientação do efetuador final do manipulador robótico. A Figura 1 ilustra a distinção entre cinemática direta e inversa.

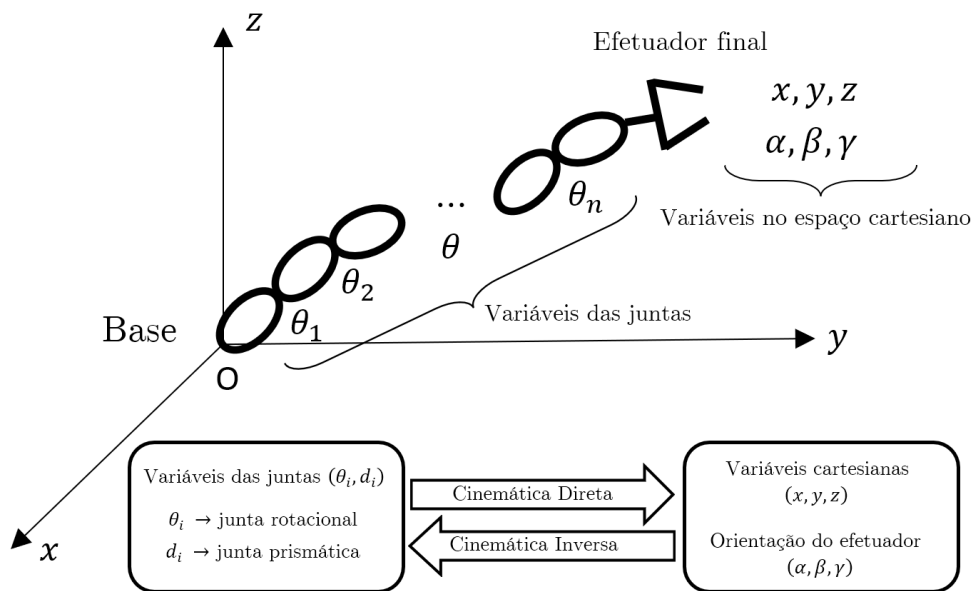


Figura 1: Cinemática direta *versus* cinemática inversa

A solução da cinemática direta independe da geometria do robô. O mesmo não ocorre com o problema cinemático inverso, uma vez que o procedimento para obtenção das equações depende exclusivamente da configuração geométrica do manipulador robótico. Essa limitação torna este problema demasiadamente mais complexo, devido à dificuldade em seguir uma sequência única de passos que facilitem a resolução. Dessa forma, cada manipulador robótico converte-se em um problema exclusivo.

---

Para casos específicos, o problema da cinemática inversa pode ser resolvido utilizando álgebra matricial, aplicações geométricas ou procedimentos iterativos, visto que depende da geometria de cada manipulador robótico. Nesse sentido, todas essas estratégias de solução são divididas em duas categorias amplas: soluções de forma fechada (métodos algébrico e geométrico) e soluções de forma aberta, também conhecidas como soluções numéricas (métodos iterativos).

Conforme Daya, Khawandi e Akoum (2010), um dos eminentes problemas na cinemática e controle de robôs consiste na determinação da solução da cinemática inversa. À medida que a complexidade do manipulador robótico aumenta, a obtenção da cinemática inversa se torna difícil e, computacionalmente, cara. Métodos tradicionais, como os geométricos, iterativos e algébricos, são inadequados, pois a estrutura geométrica do manipulador é mais complexa.

De acordo com Craig (2017), as soluções numéricas normalmente são muito mais lentas do que suas correspondentes de forma fechada, exatamente por conta de sua natureza iterativa.

Os movimentos do efetuador final são definidos no volume de trabalho do sistema robótico, o qual pode ser descrito convenientemente por diferentes tipos de sistemas de coordenadas, tais como: cartesiano, polar, esférico ou cilíndrico. Por outro lado, o espaço das juntas representa o local em que o vetor das variáveis de junta é definido. Segundo o mesmo autor, a essência do problema da cinemática de manipuladores é a coordenação dos movimentos individuais das juntas no espaço das juntas e o movimento do efetuador no volume de trabalho.

Jacques Denavit e Richard Hartenberg introduziram uma convenção com o intuito de padronizar as coordenadas de sistemas de referências para ligações espaciais (DENAVID e HARTENBERG, 1955). Após uma década, esses autores apresentaram o algoritmo para solução da cinemática de sistemas articulados (HARTENBERG e

DENAVIT, 1965). A demonstração da relevância desse algoritmo para a análise cinemática de sistemas robóticos foi apresentada por Paul (1981), após dezesseis anos.

O método criado por Denavit e Hartenberg ainda é a técnica mais utilizada para resolver o problema da cinemática direta por diversas razões. Em primeiro lugar, em intervalos específicos, pode-se descrever o ângulo de rotação (se a junta for rotativa) ou translação (caso a junta seja prismática) de cada conjunto junta/elo de um robô manipulador. Em segundo lugar, nesses intervalos de movimento, certos parâmetros, os quais serão apresentados na Seção 2.1, podem ser utilizados em uma matriz para descrever o movimento da junta  $i$  em termos da junta  $i-1$ . Desta forma, cada conjunto junta/elo pode ser descrito por uma matriz. Então, o produto de todas essas matrizes descreve a posição e a orientação do efetuador final do robô no espaço em relação a um sistema de coordenadas.

Segundo Sciavicco e Siciliano (1996), o método de Denavit-Hartenberg é fácil de ser utilizado, no entanto, para a solução da cinemática inversa, é possível se deparar com o seguinte problema: a necessidade de métodos e algoritmos adicionais, que normalmente requerem maiores esforços computacionais.

Levando em consideração o problema citado no parágrafo anterior, um método alternativo suplementar que pode solucioná-lo é utilizado: a Teoria de Bases de Gröbner. Desenvolvido por Bruno Buchberger em sua tese de doutorado, consiste em um algoritmo algébrico que pode ser aplicado a um dado conjunto (normalmente infinito) de polinômios não nulos, produzindo um conjunto finito de geradores, de tal modo que seja possível identificar, com base nesses geradores, quando um certo polinômio pertence ao conjunto original dado (BUCHBERGER, 1965).

Ao encontrar soluções para esse conjunto de geradores, encontram-se também soluções para todo o conjunto. Com o auxílio de um *software* de computação algébrica, pode-se calcular todas essas soluções, incluindo os casos não tão facilmente resolvidos utilizando álgebra matricial ou procedimentos iterativos, devido à complexidade da

geometria de alguns robôs manipuladores, determinando, assim, todas as soluções possíveis.

Com o objetivo de familiarizar o leitor com cada método, os robôs manipuladores Stäubli TS20, um robô do tipo SCARA com quatro graus de liberdade, e o Unimation PUMA 560, um manipulador antropomórfico com seis juntas rotativas, serão utilizados como estudos de caso aplicando o método de Paul com base na matriz de Denavit-Hartenberg e o método das bases de Gröbner para solução do problema da cinemática inversa.

Kendricks (2007), em sua tese de doutorado, comparou o algoritmo de Denavit e Hartenberg com o método das bases de Gröbner para a solução do problema da cinemática inversa do robô Fanuc A-510, um manipulador do tipo SCARA com três graus de liberdade. A autora comprova matematicamente que é possível determinar todas as configurações singulares de um manipulador SCARA fundamentando-se em uma base de Gröbner, sem a necessidade de utilização de um método complementar, como o Jacobiano por exemplo. Porém, devido à simplicidade geométrica do manipulador analisado, as equações produzidas com uma base de Gröbner, comparadas às equações determinadas pelo método de Paul com base em uma matriz obtida pelo algoritmo de Denavit-Hartenberg, fornecem as mesmas soluções, conforme esperado para as duas aplicações, porém com o mesmo custo computacional.

O presente trabalho pretende ir além, uma vez que aborda uma verificação da aplicabilidade do método das bases de Gröbner na solução do problema da cinemática inversa de robôs manipuladores seriais do tipo SCARA. Além disso, realiza uma análise da eficiência computacional para a solução da cinemática inversa de um manipulador robótico antropomórfico. Para tal, utiliza-se a Teoria de Bases de Gröbner, comparando com os resultados obtidos pelo método de Paul, todos baseados na matriz obtida pelo algoritmo de Denavit-Hartenberg. Desse modo, é demonstrado que o método com base

---

na Teoria de Bases de Gröbner é mais eficiente computacionalmente para esta família de manipuladores.

Segundo Craig (2017), para o controle de trajetórias de manipuladores robóticos, faz-se necessário o cálculo da cinemática inversa em intervalos mínimos de tempo, o que justifica a importância de se utilizar um método computacionalmente eficiente para a solução do problema da cinemática inversa de manipuladores robóticos.

Ni e Wu (2018) empregaram quatérnios como operadores espaciais e a Teoria de Bases de Gröbner para a solução da cinemática inversa de um robô genérico 6R. Com base no algoritmo apresentado pelos autores, 16 soluções distintas são determinadas, porém nesse conjunto apenas 4 soluções são reais. Como esse método não é comparado com nenhum outro, a sua eficiência computacional não pôde ser confirmada.

Diferentemente de Wang, Hang e Yang (2006), que também apresentaram uma solução para a cinemática inversa de um manipulador serial 6R utilizando a Teoria de Bases de Gröbner, uma base com um número reduzido de polinômios será determinada neste trabalho, uma vez que esses autores, ao utilizarem o método de solução da cinemática de mecanismos proposta por Duffy e Crane (1980), produziram uma base de Gröbner com 72 polinômios.

Guzmán-Giménez, *et al.* (2020) apresentaram um procedimento sistemático para solução do problema da cinemática inversa de sistemas robóticos de cadeia aberta não redundantes utilizando a Teoria de Bases de Gröbner após solução da cinemática direta com base no algoritmo de Denavit-Hartenberg. Porém valer ressaltar que os autores se restringiram às três primeiras juntas do robô antropomórfico Unimation PUMA 560, desconsiderando as três juntas responsáveis pela orientação do efetuador do manipulador.

O objetivo principal do trabalho aqui proposto é analisar a eficiência computacional na solução do problema da cinemática inversa de robôs manipuladores antropomórficos ao utilizar a Teoria de Bases de Gröbner.

Os dados de desempenho computacional obtidos neste trabalho visam comprovar que essa escolha reduzirá o custo computacional para a solução da cinemática inversa de manipuladores robóticos antropomórficos.

O presente trabalho está organizado da seguinte maneira: o Capítulo 2 é dedicado ao algoritmo de Denavit-Hartenberg para solução do problema cinemático direto e ao método de Paul para obtenção da solução da cinemática inversa de manipuladores robóticos. Em seguida, no Capítulo 3, é feita uma descrição da Teoria de Bases de Gröbner, em que um algoritmo para determinação dessas bases é apresentado para a solução do problema da cinemática inversa dos manipuladores estudados neste trabalho.

No Capítulo 4, o problema da cinemática inversa do robô SCARA Stäubli TS20 é solucionado utilizando o método de Paul com base no algoritmo de Denavit-Hartenberg e o Jacobiano para determinação de configurações não singulares para essa família de manipuladores robóticos. Em seguida, no Capítulo 5, a cinemática inversa desse manipulador é resolvida utilizando a Teoria de Bases de Gröbner. Ainda nesse capítulo, é realizada uma análise do método utilizado neste trabalho, comparando com os resultados obtidos pelo método de Paul.

No Capítulo 6, o problema da cinemática inversa do robô antropomórfico Unimation PUMA 560 é solucionado utilizando o método de Paul com base no algoritmo de Denavit-Hartenberg. A solução desse mesmo problema utilizando a Teoria de Bases de Gröbner é apresentada no Capítulo 7. Nele, é feita uma análise da eficiência computacional para solução da cinemática inversa desse manipulador utilizando uma nova configuração de juntas, comparando com os dados de desempenho computacional para a obtenção de todas as configurações ao utilizar o método de Paul. Ainda nesse capítulo, a cinemática inversa de um manipulador 6R com pulso não esférico também é resolvida com base na Teoria de Bases de Gröbner, com o objetivo de validar essa teoria como método para obtenção de todas as configurações de juntas para orientar e

posicionar o efetuador final de um manipulador robótico antropomórfico independentemente do número de parâmetros cinemáticos não nulos.

Por fim, o Capítulo 8 traz as conclusões do trabalho, apresentando as considerações finais e perspectivas futuras.

## 2 Algoritmo de Denavit-Hartenberg

Este capítulo é dedicado ao problema cinemático direto e ao método de Paul para obtenção da solução da cinemática inversa de manipuladores robóticos com base no algoritmo de Denavit-Hartenberg.

### 2.1 CINEMÁTICA DIRETA

A cinemática é a ciência que estuda o movimento de um corpo desconsiderando as forças que o causam. Nessa ciência são estudadas a posição e suas duas derivadas de ordem mais elevada em relação ao tempo.

Para uma cadeia cinemática com  $n$  juntas e  $n - 1$  elos, a cada junta é atribuído um sistema de referência  $(x, y, z)$ . Pode-se alinhar cada sistema de referência por meio da realização de uma série de rotações e translações. Desse modo, cada junta pode ser representada por uma matriz de transformação homogênea que descreva a rotação específica ou translação necessária para alinhar a junta  $i - 1$  com a junta  $i$ , podendo ser a combinação das duas. O produto dessas matrizes determina a posição final da  $n$ -ésima junta, que representa o efetuador final do manipulador robótico.

A matriz de transformação homogênea pode ser descrita como:

$$\begin{bmatrix} n_x & o_x & a_x & x \\ n_y & o_y & a_y & y \\ n_z & o_z & a_z & z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (1)$$

A matriz (1) descreve os vetores de direção para os eixos  $x_2$ ,  $y_2$  e  $z_2$  de um sistema de referência  $\{2\}$  em termos dos vetores de direção para os eixos  $x_1$ ,  $y_1$  e  $z_1$  de um sistema de referência  $\{1\}$ . Os dois sistemas estão representados na Figura 2, em



que o vetor normal  $\vec{n} = (n_x, n_y, n_z)$  representa o eixo  $x$ , o vetor de orientação  $\vec{o} = (o_x, o_y, o_z)$  representa o eixo  $y$ , e o vetor de aproximação  $\vec{a} = (a_x, a_y, a_z)$  representa o eixo  $z$ .

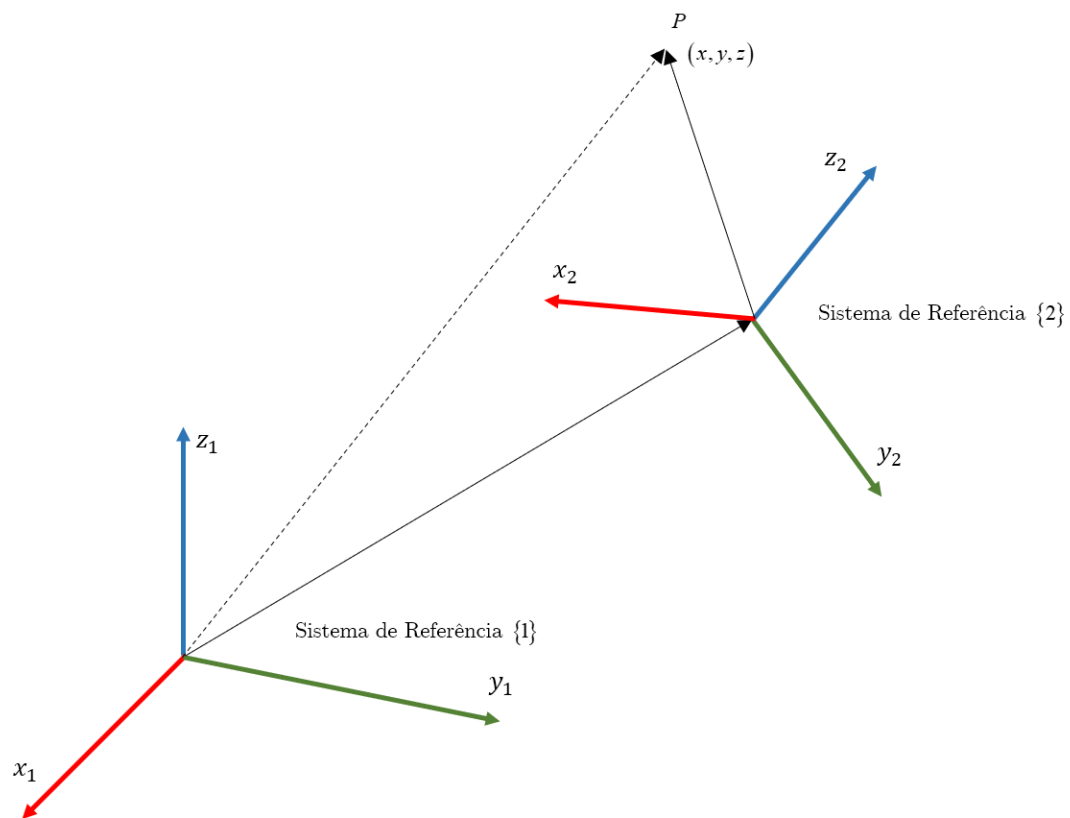


Figura 2: Sistemas de referência deslocados no mesmo espaço tridimensional

Para a primeira coluna da matriz,  $n_x$ ,  $n_y$ ,  $n_z$  são as componentes do vetor que define o eixo  $x$  no sistema de referência  $\{2\}$  em termos dos três vetores dos eixos do sistema de referência  $\{1\}$ . Para a segunda coluna,  $o_x$ ,  $o_y$ ,  $o_z$  são as componentes do vetor que define o eixo  $y$  no sistema de referência  $\{2\}$  em termos dos três vetores dos eixos do sistema de referência  $\{1\}$ .

Para a terceira coluna,  $a_x$ ,  $a_y$ ,  $a_z$  são as componentes do vetor que define o eixo  $z$  no sistema de referência  $\{2\}$  em termos dos três vetores dos eixos do sistema de

referência  $\{1\}$ . A quarta coluna é a posição no sistema de referência  $\{2\}$  em relação ao sistema de referência  $\{1\}$ .

Um sistema de coordenadas cartesiano deve ser atribuído a cada conjunto junta/elo de um robô manipulador, de modo que, a cada elo ligado, é atribuído um eixo de coordenadas específico. Denavit e Hartenberg (1955) propuseram uma notação sistemática para atribuir um sistema de coordenadas ortonormal para cada elo em uma cadeia cinemática aberta. Cada sistema de coordenadas é determinado com base nas seguintes regras:

- i. o eixo  $z_{k-1}$  se encontra ao longo do eixo de movimento (rotação ou translação) da junta  $k$ ;
- ii. o eixo  $x_k$  é perpendicular ao eixo  $z_{k-1}$ , com o seu sentido positivo na direção do eixo  $z_k$ ;
- iii. o eixo  $y_k$  é escolhido de forma que os três eixos formem um sistema ortogonal dextrogiro<sup>1</sup> (LEE, 1982).

Uma vez que esses sistemas de coordenadas fixados aos elos são atribuídos, transformações entre sistemas de coordenadas adjacentes podem ser representadas por uma matriz de transformação de coordenadas homogêneas.

Os quatro parâmetros  $a_k$ ,  $d_k$ ,  $\alpha_k$ ,  $\theta_k$  ilustrados na Figura 3 representam:  $a_k$ , a distância ao longo de  $x_k$ , da origem  $O_k$  à interseção dos eixos  $x_k$  e  $z_{k-1}$ ;  $d_k$ , a distância ao longo de  $z_{k-1}$ , de  $O_{k-1}$  à interseção dos eixos  $x_k$  e  $z_{k-1}$ ;  $\alpha_k$ , o ângulo do eixo  $z_{k-1}$  para o eixo  $z_k$ , medido em torno de  $x_k$ ; e  $\theta_k$ , o ângulo do eixo  $x_{k-1}$  para o eixo  $x_k$ , medido em torno de  $z_{k-1}$ , (sinal obtido pela regra da mão direita).

---

<sup>1</sup> O termo "dextrogiro" se refere à rotação para a direita. Resumidamente, um sistema dextrogiro é aquele que cumpre a regra da mão direita.

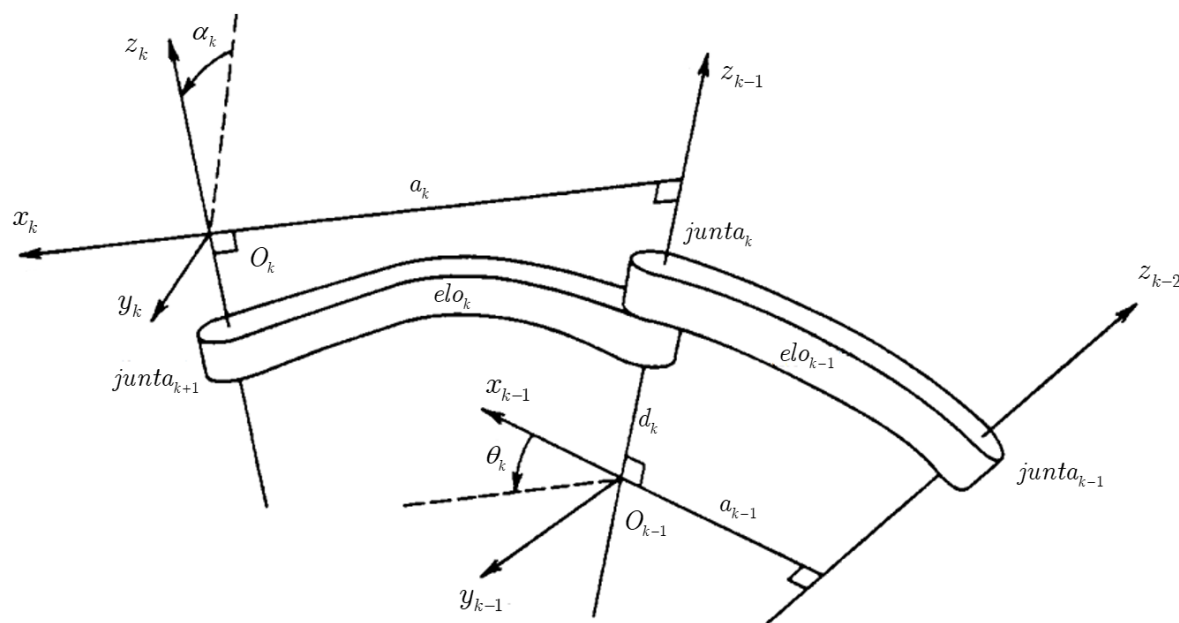


Figura 3: Representação dos parâmetros da notação de Denavit-Hartenberg  
 Fonte: Adaptado de (SPONG e VIDYASAGAR, 2008)

Caso os parâmetros cinemáticos fossem estáticos, ou constantes, a transformação homogênea também seria constante, fazendo com que o robô permanecesse inerte. Dois dos quatro parâmetros utilizados como referência possuem relação direta com a componente das variáveis das juntas. Para juntas rotacionais, a variável de junta é o ângulo  $\theta_k$ ; enquanto que a variável de junta é a distância entre os elos  $d_k$ , caso seja prismática. Esses parâmetros são obtidos com base no algoritmo proposto por Denavit e Hartenberg e são apresentados na Tabela 1.

Tabela 1: Parâmetros cinemáticos para juntas rotacionais e prismáticas

Parâmetro	Símbolo	Junta rotacional	Junta prismática
Ângulo de junta	$\theta_k$	Variável	Fixo
Deslocamento de junta	$d_k$	Fixo	Variável
Comprimento do elo	$a_k$	Fixo	Fixo
Ângulo de torção do elo	$\alpha_k$	Fixo	Fixo

De acordo com Tsai (1999), na representação de Denavit-Hartenberg original, associa-se o eixo de rotação para junta rotativa ou o eixo de translação para junta prismática ao eixo  $z$  do sistema coordenado fixado na junta, e cada matriz é representada pelo produto de quatro transformações básicas envolvendo rotações e translações, como pode ser observado na Equação (2). Utiliza-se  $s_{\theta_k}$  para  $\sin(\theta_k)$  e  $c_{\theta_k}$  para  $\cos(\theta_k)$ , analogamente para  $\alpha_k$ .

$${}^{k-1}A_k = Rot_{z_k, \theta_k} \cdot Transl_{z_k, d_k} \cdot Transl_{x_k, a_k} \cdot Rot_{x_k, \alpha_k}$$

$$= \begin{bmatrix} c_{\theta_k} & -s_{\theta_k} & 0 & 0 \\ s_{\theta_k} & c_{\theta_k} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_k \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_k \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{\alpha_k} & -s_{\alpha_k} & 0 \\ 0 & s_{\alpha_k} & c_{\alpha_k} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

O termo  $Rot_{z_k, \theta_k}$  representa a rotação  $\theta_k$  em torno do eixo  $z_k$ ;  $Transl_{z_k, d_k}$ , a translação  $d_k$  ao longo do eixo  $z_k$ ;  $Transl_{x_k, a_k}$ , a translação  $a_k$  ao longo do eixo  $x_k$ ; e por fim,  $Rot_{x_k, \alpha_k}$ , representa a rotação  $\alpha_k$  em torno do eixo  $x_k$ .

Esta operação resulta na matriz representada pela Equação (3), sendo que por meio dela, é realizado o mapeamento de coordenadas entre os elos do robô manipulador.

$${}^{k-1}A_k = \begin{bmatrix} c_{\theta_k} & -s_{\theta_k} c_{\alpha_k} & s_{\theta_k} s_{\alpha_k} & a_k c_{\theta_k} \\ s_{\theta_k} & c_{\theta_k} c_{\alpha_k} & -c_{\theta_k} s_{\alpha_k} & a_k s_{\theta_k} \\ 0 & s_{\alpha_k} & c_{\alpha_k} & d_k \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

Multiplicando todas as matrizes de transformação obtidas pelo algoritmo de Denavit-Hartenberg, obtém-se a matriz de transformação homogênea que parte da base à extremidade do efetuador final, ou seja, variando  $k$  de 0 a  $n$ . Essa matriz resultante pode ser considerada como a solução do problema da cinemática direta, como podemos observar na Equação (4), na qual  $q$  representa as variáveis de junta ( $\theta_k$  para rotativa e  $d_k$  para prismática);  $R(q)$ , a matriz de rotação; e  $p(q)$ , o vetor posição.

$${}^{\text{base}}T(q)_{\text{efetuador final}} = \begin{bmatrix} R(q) & p(q) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

Resumidamente, para um manipulador com  $n$  graus de liberdade, determina-se  $T$ , que é uma função de todas as variáveis de junta.

$$T = A_{01} \cdot A_{12} \cdot A_{23} \cdot \dots \cdot A_{(n-1)n} = \prod_{i=1}^n A_{0i} = A_{0n} \quad (5)$$

Onde cada  $A_{ij}$  é uma matriz D-H (Denavit-Hartenberg), definida em (3).

Dessa forma, quando  $T$  é multiplicada pelo vetor  $[0 \ 0 \ 0 \ 1]^T$ , a matriz transforma a origem do sistema de coordenadas nas coordenadas do efetuador final.

## 2.2 CINEMÁTICA INVERSA

De acordo com Craig (2017), o problema inverso tem sido o foco da análise cinemática de robôs manipuladores. Para determinar as combinações necessárias para posicionar e orientar o efetuador final de um robô manipulador em um ponto específico no espaço, deve-se considerar a posição atual e calcular os movimentos associados com cada variável comum.

Segundo Barrientos (2007), a solução da cinemática direta independe da configuração do robô, o mesmo não ocorre com o problema cinemático inverso, sendo que o procedimento para obtenção das equações depende exclusivamente da configuração do manipulador.

A cinemática inversa é um dos tópicos de estudo de maior importância na Robótica. O entendimento da cinemática inversa fornece uma análise dos movimentos do robô relacionados ao espaço geométrico no qual ele se insere.

A resolução do problema da cinemática inversa ficaria mais apropriada se fosse encontrada uma única solução, ou seja, com a determinação de uma relação matemática

explícita dos valores de junta ( $q_k$ ) em função da posição e orientação  $(x, y, z, \alpha, \beta, \gamma)$  do efetuador final, onde  $k$  indica o número da junta, isto é:  $q_k = f_k(x, y, z, \alpha, \beta, \gamma)$ .

Podem ser adotadas diversas técnicas para solucionar o problema da cinemática inversa de um manipulador robótico, por exemplo: método de Paul, simbólico, análise da posição zero, vetores *screw*, Newton-Raphson, pseudoinversa, integração das velocidades, entre outros.

Exclusivamente para alguns manipuladores robóticos que possuem seis graus de liberdade e quando suas três últimas juntas possuem eixos que se intersectam em um único ponto, é possível utilizar o método do desacoplamento proposto por Pieper (1968), que permite separar o problema em duas etapas independentes: uma relacionada aos três primeiros graus mais próximos à base; e outra aos três mais próximos do efetuador final.

O método de Paul para solução do problema cinemático inverso de manipuladores robóticos, citado anteriormente, e que será comparado com o método da Teoria de Bases de Gröbner, será apresentado na Subseção 2.2.1 deste trabalho.

### 2.2.1 Método de Paul

Este método, apresentado por Paul (1981), é muito utilizado em robôs industriais e, com base nele, as coordenadas das juntas são determinadas igualando-se os elementos correspondentes das matrizes  $T$  e  $A_{0n}$  representadas em (5). Busca-se, assim, igualdades que, após algumas transformações, resultem na coordenada de junta procurada.

Caso não se consiga obter as relações desejadas pela Equação (5), o que normalmente acontece, pesquisa-se entre matrizes que relacionem sistemas

intermediários. Matematicamente isso corresponde a realizar multiplicações à esquerda e/ou à direita em ambos os membros da igualdade (5) pelas inversas das matrizes  $A_{i-1,i}$ , uma de cada vez, de modo a formar novos conjuntos de equações.

Para um robô com seis graus de liberdade, por exemplo, pode-se obter as seguintes equações matriciais dentro de um universo maior de combinações:

$$\begin{aligned}
T &= A_{01} \cdot A_{12} \cdot A_{23} \cdot A_{34} \cdot A_{45} \cdot A_{56}, \\
A_{01}^{-1} \cdot T &= A_{12} \cdot A_{23} \cdot A_{34} \cdot A_{45} \cdot A_{56}, \\
A_{12}^{-1} \cdot A_{01}^{-1} \cdot T &= A_{23} \cdot A_{34} \cdot A_{45} \cdot A_{56}, \\
A_{23}^{-1} \cdot A_{12}^{-1} \cdot A_{01}^{-1} \cdot T &= A_{34} \cdot A_{45} \cdot A_{56}, \\
A_{34}^{-1} \cdot A_{23}^{-1} \cdot A_{12}^{-1} \cdot A_{01}^{-1} \cdot T &= A_{45} \cdot A_{56}, \\
A_{45}^{-1} \cdot A_{34}^{-1} \cdot A_{23}^{-1} \cdot A_{12}^{-1} \cdot A_{01}^{-1} \cdot T &= A_{56}, \\
&\vdots \\
A_{12}^{-1} \cdot A_{01}^{-1} \cdot T \cdot A_{56}^{-1} &= A_{23} \cdot A_{34} \cdot A_{45}.
\end{aligned} \tag{6}$$

Por inspeção, observa-se os elementos correspondentes em cada equação matricial, buscando-se aqueles que possam fornecer uma expressão que possibilite a solução para a variável de junta a ser encontrada. Esse procedimento se repete para outros elementos desta ou das outras equações matriciais, até que se chegue a todas as variáveis de juntas.

Este trabalho visa comprovar que, para a solução do problema cinemático inverso de manipuladores antropomórficos, o método da Teoria de Bases de Gröbner, que será apresentado no Capítulo 3, é computacionalmente mais eficiente que o método de Paul.

# 3 TEORIA DE BASES DE GRÖBNER

A Teoria de Bases de Gröbner é aplicada em áreas distintas da Matemática. Por exemplo, Robbiano (1998), utiliza bases de Gröbner para resolver problemas de Estatística. Wang (1998), mostra como as bases de Gröbner podem ser utilizadas para solucionar problemas geométricos complexos. Cox, Little e O'Shea (1992), têm dedicado seus trabalhos para as aplicações da Teoria de Bases de Gröbner em Geometria, Teoria dos Grafos e Robótica.

A Teoria de Bases de Gröbner permite que seja utilizado um algoritmo para resolver sistemas de equações polinomiais. Mais formalmente, este algoritmo, chamado algoritmo de Buchberger (BOSE, 1995), aplica técnicas de álgebra computacional para ideais polinomiais específicos, produzindo uma base de Gröbner que pode ser usada para encontrar soluções comuns para um conjunto de diferentes polinômios não nulos.

Resumidamente, uma base de Gröbner para um sistema de equações polinomiais é um novo sistema com as mesmas soluções, porém mais fácil de ser resolvido.

Alguns conceitos iniciais sobre grupos, anéis, domínios, corpos e ideais, que serão utilizados ao longo do texto, serão apresentados na Seção 3.1.

## 3.1 GRUPOS, ANÉIS, DOMÍNIOS, CORPOS E IDEAIS

Um grupo  $(G, +)$  é um conjunto não vazio, no qual permite uma operação binária, denotada por  $+$ , chamada de adição, que satisfaz as seguintes condições:

(A1) Se  $a$  e  $b$  pertencem a  $G$ , então  $a + b$  também pertence a  $G$ . Propriedade denominada “Fechamento sob adição”.

(A2) A adição é associativa, isto é:  $\forall a, b, c \in G, (a + b) + c = a + (b + c)$ .



(A3)  $G$  possui um elemento neutro com respeito à adição, denotado por  $0$  e chamado de zero, isto é:  $\exists 0 \in G$  tal que  $a + 0 = a$  e  $0 + a = a$ ,  $\forall a \in G$ .

(A4) Cada elemento de  $G$  possui um inverso com respeito à adição, isto é: para cada  $a \in G, \exists -a \in G$  tal que  $a + (-a) = 0$  e  $(-a) + a = 0$ .

Um grupo abeliano  $(G, +)$  satisfaz de (A1) a (A4) e ainda mais a seguinte condição:

(A5) A adição é comutativa, isto é:  $\forall a, b \in G, a + b = b + a$ .

Um anel  $(A, +, \cdot)$ , munido de uma operação denotada por  $+$ , chamada de adição, e de uma operação denotada por  $\cdot$ , chamada de multiplicação, é um grupo abeliano com relação à operação  $+$ , ou seja, satisfaz as condições (A1) a (A5), sendo que as operações satisfazem ainda as seguintes condições:

(M1) Se  $a$  e  $b$  pertencem a  $A$ , então  $a \cdot b$  também pertence a  $A$ . Propriedade denominada “Fechamento sob multiplicação”.

(M2)  $\cdot$  é associativa, ou seja,  $(a \cdot b) \cdot c = a \cdot (b \cdot c)$  para quaisquer  $a, b, c \in A$ .

(M3)  $\cdot$  é distributiva relativamente a  $+$ , ou seja,  $a \cdot (b + c) = a \cdot b + a \cdot c$  e  $(b + c) \cdot a = b \cdot a + c \cdot a$  para quaisquer  $a, b, c \in A$ .

Utiliza-se simplesmente a letra  $A$  para designar um anel arbitrário  $(A, +, \cdot)$ . Um anel  $A$  diz-se comutativo se  $\cdot$  é comutativa, ou seja, se a seguinte condição for atendida:

(M4)  $a \cdot b = b \cdot a$ ,  $\forall a, b \in A$ .

Chama-se anel com unidade, ou anel unitário, se a operação  $\cdot$  possui um elemento neutro, chamado identidade. Dessa forma, a seguinte condição é atendida:

(M5)  $A$  possui um elemento neutro, diferente de zero, com respeito à multiplicação, denotado por  $1$  e chamado de um, isto é:  $\exists 1 \in A - \{0\}$ , tal que  $1 \cdot a = a$  e  $a \cdot 1 = a, \forall a \in A$ .

Um domínio de integridade é um anel comutativo com unidade sem divisores de zero, ou seja, onde a lei do cancelamento para o produto é válida. Dessa forma, a seguinte condição é atendida:

(M6) Se  $a, b \in A$  e  $a \cdot b = 0$ , então  $a = 0$  ou  $b = 0$ .

Seja  $(A, +, \cdot)$  um anel, um subconjunto não vazio  $B \subseteq A$  é definido como um subanel de  $A$  quando  $(B, +, \cdot)$  é um anel, em que  $+$  e  $\cdot$  são as mesmas operações de  $A$ .

Um domínio de integridade também é chamado de anel de integridade ou simplesmente domínio.

Um corpo  $k$  é um anel comutativo com unidade em que todo elemento diferente de zero possui inverso. Dessa forma, a seguinte condição é atendida:

(M7) Se  $a \in A$  e  $a \neq 0$ , existe um elemento  $a^{-1} \in A$  tal que  $a \cdot a^{-1} = a^{-1} \cdot a = 1$ .

A partir deste ponto, um anel comutativo com unidade será chamado simplesmente de anel. O anel principal que será tratado nesta tese é o de polinômios em várias variáveis, que será definido na Seção 3.2.

Seja  $A$  um anel, dizemos que um subconjunto não vazio  $I \subseteq A$  é um ideal, se:

- i.  $0 \in I$ ;
- ii.  $a + b \in I, \forall a, b \in I$ ;
- iii.  $ca \in I, \forall a \in I, \forall c \in A$ .

O conceito de ideal é fundamental no estudo da Geometria Algébrica. Essa área combina técnicas de Álgebra Abstrata, especialmente de Álgebra Comutativa, com a linguagem e os problemas da Geometria. Os principais objetos da Geometria Algébrica são as variedades algébricas, isto é, objetos geométricos definidos por polinômios.

### 3.2 ANÉIS DE POLINÔMIOS

Seja  $A$  um anel comutativo e seja  $A[x]$ , o conjunto de todos os polinômios na variável  $x$  com coeficientes em  $A$ :

$$A[x] = \{f(x) = a_0 + a_1x + \cdots + a_nx^n \mid n \geq 0, a_i \in A, \forall i\}. \quad (7)$$

Em  $A[x]$  são definidas duas operações  $+$  e  $\cdot$  por: se  $f(x) = a_0 + a_1x + \cdots + a_nx^n$  e  $g(x) = b_0 + b_1x + \cdots + b_mx^m \in A[x]$ , então:

$$f(x) + g(x) = (a_0 + b_0) + (a_1 + b_1)x + \cdots + (a_k + b_k)x^k, k = \max\{n, m\} \quad (8)$$

e

$$f(x) \cdot g(x) = c_0 + c_1x + \cdots + c_{n+m}x^{n+m}, \quad (9)$$

onde  $c_i = a_ib_0 + a_{i-1}b_1 + \cdots + a_0b_i$ .

Dessa forma, tem-se que  $(A[x], +, \cdot)$  é um anel de polinômios em uma variável com coeficientes no anel  $A$ .

Algumas definições e propriedades merecem destaque:

- i. O elemento neutro de  $A[x]$  é o polinômio nulo, ou seja,  $0 = 0 + 0x + 0x^2 + \cdots$ ;
- ii. Um polinômio é denominado constante se ele é da forma  $f(x) = a_0$ , com  $a_i = 0$  para todo  $i > 0$ ;
- iii.  $A$  é um subanel de  $A[x]$ . Os elementos do anel  $A$ , em  $A[x]$ , fazem o papel dos polinômios constantes.

Dado um anel de polinômios  $A[x]$  com coeficientes em  $x$ , será considerada a introdução de uma nova variável  $y$ , por exemplo. Observa-se que  $x$  é elemento de  $A[x]$  e, portanto, no novo anel, representa uma constante. Neste novo anel, representado por  $A[x][y]$ , no qual seus elementos podem ser reescritos na forma  $\sum a_{ij}x^iy^j$ , em que  $a_{ij} \in A, i, j \in \mathbb{N}_n$ , com um número finito de coeficientes não nulos  $a_{ij}$ .

Considerando, por exemplo,  $x + 1, x^2 + 2x + 3, x^3 + 1 \in A[x]$ , tem-se:  $(x + 1) + (x^2 + 2x + 3)y + (x^3 + 1)y^2 \in A[x][y]$ . Podendo escrever este polinômio na variável  $y$  como sendo:  $1 + x + 3y + 2xy + x^2y + y^2 + x^3y^2$ .

O anel  $A[x][y]$  pode ser escrito como  $A[x, y]$ . Analogamente, dado um anel comutativo com unidade  $A$ ,  $A[x, y, z]$  pode ser definido e, mais geralmente, o anel de polinômios  $A[x_1, \dots, x_n]$ , em  $n$  variáveis  $x_1, \dots, x_n$  com coeficientes em  $A$ .

O anel  $A[x_1, \dots, x_n]$  dos polinômios com  $n$  variáveis  $x_1, \dots, x_n$  pode ser definido simplesmente por iteração:  $A[x_1, \dots, x_n] = A[x_1, \dots, x_{n-1}][x_n]$ . Essa definição confirma a possibilidade de reorganização dos termos de um polinômio, colocando-se em evidência uma determinada variável. Por exemplo, o polinômio  $7x^2y^3 + x^3y + y^3 + 8x + 5y \in \mathbb{R}[x, y]$  pode ser escrito como  $(7x^2 + 1)y^3 + (x^3 + 5)y + 8x$ , um polinômio em  $y$  com coeficientes em  $\mathbb{R}[x]$ .

### 3.3 ORDENAÇÃO MONOMIAL

Como polinômios são somas de monômios, faz-se necessário ordenar os seus termos em ordem decrescente ou crescente. Nesses casos, todos os pares de monômios devem ser comparados, e uma ordem entre eles deve ser estabelecida. Isso significa que para todo par de monômios  $x^\alpha$  e  $x^\beta$ , exatamente uma das três relações de igualdade deve ocorrer:  $x^\alpha \succ x^\beta$ ,  $x^\alpha = x^\beta$  e  $x^\alpha \prec x^\beta$ , em que  $x$  representa a multivariável  $x_1, \dots, x_n$ ; e  $\alpha$  e  $\beta$ , os vetores dos expoentes. Relações de ordem em que cada par de elementos é comparável por uma destas relações são denominadas ordens lineares ou totais.

A partir deste ponto, a ordem lexicográfica (*lex*) será definida.

Sejam  $\alpha = (\alpha_1, \dots, \alpha_n)$  e  $\beta = (\beta_1, \dots, \beta_n) \in \mathbb{Z}_{\geq 0}^n$ , pode-se afirmar que  $\alpha \succ_{lex} \beta$  se, no vetor diferença  $\alpha - \beta \in \mathbb{Z}_{\geq 0}^n$ , a coordenada não nula mais à esquerda é positiva. Escreve-se  $x^\alpha \succ_{lex} x^\beta$ , ou simplesmente  $x^\alpha \succ x^\beta$ , se  $\alpha \succ_{lex} \beta$ .

Por exemplo:  $(3, 0, 0) \succ_{lex} (2, 3, 4)$ , uma vez que  $\alpha - \beta = (1, -3, -4)$ ;  $(2, 3, 4) \succ_{lex} (2, 3, 1)$ , já que  $\alpha - \beta = (0, 0, 3)$ . As variáveis  $x_1, \dots, x_n$  são ordenadas da maneira mais usual pela ordem lexicográfica  $(1, 0, \dots, 0) \succ_{lex} (0, 1, \dots, 0) \succ_{lex} \dots \succ_{lex} (0, 0, \dots, 1)$ , e conseqüentemente  $x_1 \succ_{lex} x_2 \succ_{lex} \dots \succ_{lex} x_n$ .

Tomando como exemplo:  $f = -5z^2 - 8x^3 + xy^2z + 3x^2z^2 \in \mathbb{R}[x, y, z]$ , com relação à ordem lexicográfica  $x \succ_{lex} y \succ_{lex} z$ , tem-se:  $f = -8x^3 + 3x^2z^2 + xy^2z - 5z^2$ . Observa-se que os monômios de  $f$  estão escritos em ordem decrescente.

Existem muitas ordens lexicográficas de acordo com a forma como as variáveis são ordenadas. No caso de  $n$  variáveis, existem  $n!$  diferentes ordens lexicográficas. Em uma ordem lexicográfica, uma variável domina qualquer monômio envolvendo apenas variáveis menores, independente do grau total. Por exemplo, para a ordem lexicográfica  $x \succ y \succ z$ , tem-se  $x^2 \succ y^{12}z^8$ .

Em alguns casos, opta-se por considerar o grau total de um monômio e ordenar os monômios de graus maiores primeiro. Neste caso específico, utiliza-se a ordem lexicográfica graduada (*grlex*) que será definida a partir deste ponto.

Sejam  $\alpha = (\alpha_1, \dots, \alpha_n)$  e  $\beta = (\beta_1, \dots, \beta_n) \in \mathbb{Z}_{\geq 0}^n$ ,  $|\alpha| = \sum_i \alpha_i$  e  $|\beta| = \sum_i \beta_i$ .

Constata-se que  $\alpha \succ_{grlex} \beta$  se,  $|\alpha| \succ |\beta|$ , ou  $|\alpha| = |\beta|$  e  $\alpha \succ_{lex} \beta$ . Portanto, a ordem lexicográfica graduada ordena os monômios inicialmente pelo grau total e, quando são iguais, pela ordem lexicográfica. Por exemplo:  $(3, 3, 4) \succ_{grlex} (5, 5, 2)$ , uma vez que  $|(3, 3, 4)| = 10 > 9 = |(5, 5, 2)|$ ;  $(3, 1, 4) \succ_{grlex} (1, 2, 5)$ , já que  $|(3, 1, 4)| = 8 = |(1, 2, 5)|$  e  $(3, 1, 4) \succ_{lex} (1, 2, 5)$ .

Retomando o exemplo anterior:  $f = -5z^2 - 8x^3 + xy^2z + 3x^2z^2 \in \mathbb{R}[x, y, z]$ , com relação à ordem lexicográfica graduada  $x \succ_{grlex} y \succ_{grlex} z$ , tem-se:  $f = 3x^2z^2 + xy^2z - 8x^3 - 5z^2$ , com os monômios escritos em ordem decrescente.

Uma outra ordem é dada pela ordem lexicográfica graduada reversa (*grevlex*), que é a mais eficiente das ordens em muitas operações algébricas referentes às bases de Gröbner.

Sejam  $\alpha = (\alpha_1, \dots, \alpha_n)$  e  $\beta = (\beta_1, \dots, \beta_n) \in \mathbb{Z}_{\geq 0}^n$  dizemos que  $\alpha \succ_{grevlex} \beta$  se,  $|\alpha| \succ |\beta|$ , ou  $|\alpha| = |\beta|$  e em  $\alpha - \beta \in \mathbb{Z}^n$  a primeira coordenada não nula, iniciando à direita, é negativa. Por exemplo:  $(1, 3, 2) \succ_{grevlex} (4, 0, 1)$ , um vez que  $|(1, 3, 2)| = 6 > 5 = |(4, 0, 1)|$ ;  $(3, 4, 1, 2) \succ_{grevlex} (4, 2, 2, 2)$ , já que  $|(3, 4, 1, 2)| = 10 = |(4, 2, 2, 2)|$  e  $\alpha - \beta = (-1, 2, -1, 0)$ .

E, finalmente, retomando mais uma vez o exemplo utilizado anteriormente:  $f = -5z^2 - 8x^3 + xy^2z + 3x^2z^2 \in \mathbb{R}[x, y, z]$ , com relação à ordem lexicográfica graduada reversa  $x \succ_{grevlex} y \succ_{grevlex} z$ , tem-se:  $f = xy^2z + 3x^2z^2 - 8x^3 - 5z^2$ , com os monômios escritos em ordem decrescente.

A partir deste ponto, tomando  $k[x_1, \dots, x_n]$  ordenado com relação a uma ordem previamente fixada, então, para todo  $f \in k[x_1, \dots, x_n]$ , com  $f \neq 0$ , tem-se:

$$f = \sum_{j=1}^n a_j x^{\alpha_j} = a_1 x^{\alpha_1} + a_2 x^{\alpha_2} + \dots + a_n x^{\alpha_n}, \quad (10)$$

onde os  $a_j$  pertencem a  $k$ , e os  $x^{\alpha_j}$  são ordenados de modo que  $x^{\alpha_1} \succ x^{\alpha_2} \succ \dots \succ x^{\alpha_n}$ .

Define-se:

- i. O grau de  $f$  é  $\text{multigrav}(f) = \max\{\alpha \in \mathbb{Z}_{\geq 0}^n \mid a_\alpha \neq 0\} = \alpha_1$ , onde o máximo é tomado com relação a  $\succ$ ;
- ii. O monômio líder de  $f$  é definido como  $ml(f) = x^{\alpha_1} = x^{\text{multigrav}(f)}$ ;
- iii. O coeficiente líder de  $f$  é definido como  $cl(f) = a_1$ ;

- iv. O termo líder de  $f$  é definido como  $tl(f) = a_1x^{\alpha_1} = cl(f) \cdot ml(f)$ ;
- v.  $f$  é dito mônico se  $cl(f) = 1$ .

Baseando-se nessas informações, de acordo com Boas e Marcos (2013), pode-se afirmar que sejam  $f, g \in k[x_1, \dots, x_n]$  polinômios não nulos, então:

$$multigrav(f \cdot g) = multigrav(f) + multigrav(g); \quad \text{e se } f + g \neq 0, \quad \text{então}$$

$$multigrav(f + g) \leq \max\{multigrav(f), multigrav(g)\}.$$

Segundo os mesmos autores, se  $multigrav(f) \neq multigrav(g)$ , então

$$multigrav(f + g) = \max\{multigrav(f), multigrav(g)\}.$$

Existem dois algoritmos que são as bases desta teoria: o Algoritmo da Divisão e o Algoritmo de Buchberger. O Algoritmo da Divisão decide a pertinência de um polinômio em um ideal de polinômios, dados o polinômio e uma Base de Gröbner para o ideal em questão. Já o Algoritmo de Buchberger fornece uma Base de Gröbner para um ideal, com base em um conjunto finito de geradores.

### 3.4 ALGORITMO DA DIVISÃO

A divisão de polinômios é essencial no estudo de ideais em anéis de polinômios. Faz-se necessário a utilização do algoritmo da divisão em conjunto com bases de Gröbner para verificar se um dado polinômio pertence a um ideal.

Os algoritmos da divisão em anéis polinomiais com uma e com várias variáveis serão definidos nas Subseções 3.4.1 e 3.4.2, respectivamente.

3.4.1 Algoritmo da Divisão em  $k[x]$ 

Em anéis polinomiais em uma única variável, o algoritmo da divisão de Euclides é utilizado para dividir polinômios, ou seja, dados dois polinômios  $f(x), g(x) \in k[x]$ , com  $g(x) \neq 0$ . Então, existem polinômios únicos  $a(x), r(x) \in k[x]$  tais que  $f(x) = a(x) \cdot g(x) + r(x)$ , com  $r(x) = 0$  ou  $\text{grau}(r) < \text{grau}(g)$ .

Dessa forma, como todo ideal  $I \subset k[x]$  é gerado por um único gerador, ou seja, é principal, se  $I = \langle g \rangle$ , tem-se  $f(x) \in I$  se  $r(x) = 0$ . Se  $f(x) = a(x) \cdot g(x) + r(x)$ , com  $r(x) \neq 0$  e  $\text{grau}(r) < \text{grau}(g)$ , pode-se afirmar que  $f(x) \notin I$  (COX, LITTLE e O'SHEA, 1992).

O algoritmo da divisão de Euclides pode ser descrito como:

```

entrada  $f(x) \in k[x]$  e  $g(x) \in k[x] \setminus \{0\}$ 
início
   $a \leftarrow 0$ 
   $r \leftarrow f$ 
  enquanto  $r \neq 0$  e  $\text{grau}(r) \geq \text{grau}(g)$  faça
     $a \leftarrow a + \frac{tl(r)}{tl(g)}$ 
     $r \leftarrow r - \frac{tl(r)}{tl(g)}g$ 
  fim enquanto
forneça  $a, r$ 
fim

```

Tomando como exemplo a divisão de  $f(x) = x^4 + 2x^3 + 3x^2 + x + 4$  por  $g(x) = x^2 + 2x + 2$ , nota-se as seguintes etapas:



- i. Inicialmente, atribui-se  $f$  a  $r$ , o divisor  $g$  é listado na chave e o quociente  $a$  abaixo da chave. Dessa forma, apresenta-se o seguinte esquema:

$$\begin{array}{l|l} r : x^4 + 2x^3 + 3x^2 + x + 4 & g : x^2 + 2x + 2 \\ \hline & a : 0 \end{array}$$

- ii. Como  $r \neq 0$  e  $\text{grau}(r) \geq \text{grau}(g)$ ,  $a = a + \frac{tl(r)}{tl(g)} = x^2$  e

$$r = r - \frac{tl(r)}{tl(g)}g = x^2 + x + 4. \text{ Desse modo, tem-se:}$$

$$\begin{array}{l|l} x^4 + 2x^3 + 3x^2 + x + 4 & g : x^2 + 2x + 2 \\ -(x^4 + 2x^3 + 2x^2) & a : x^2 \\ \hline r : x^2 + x + 4 & \end{array}$$

- iii. Como  $r \neq 0$  e  $\text{grau}(r) \geq \text{grau}(g)$ ,  $a = a + \frac{tl(r)}{tl(g)} = x^2 + 1$  e

$$r = r - \frac{tl(r)}{tl(g)}g = 5x + 2. \text{ Nesse sentido, consta-se:}$$

$$\begin{array}{l|l} x^4 + 2x^3 + 3x^2 + x + 4 & g : x^2 + 2x + 2 \\ -x^4 - 2x^3 - 2x^2 & a : x^2 + 1 \\ \hline x^2 + x + 4 & \\ -(x^2 + 2x + 2) & \\ \hline r : 5x + 2 & \end{array}$$

- iv. Com base nesse último passo, observa-se que a condição ( $r \neq 0$  e  $\text{grau}(r) \geq \text{grau}(g)$ ) não é verdadeira, uma vez que  $\text{grau}(r) < \text{grau}(g)$ . Com isso, o algoritmo da divisão é concluído.

Assim, percebe-se que:  $f(x) = a(x) \cdot g(x) + r(x)$  com  $\text{grau}(r) < \text{grau}(g)$ , ou seja,  $(x^4 + 2x^3 + 3x^2 + x + 4) = (x^2 + 1) \cdot (x^2 + 2x + 2) + (5x + 2)$ .

### 3.4.2 Algoritmo da divisão em $k[x_1, \dots, x_n]$

Fixada uma ordem monomial  $\succ$  e dado  $g \in k[x_1, \dots, x_n]$  não nulo, para qualquer polinômio  $f \in k[x_1, \dots, x_n]$  existem  $a, r \in k[x_1, \dots, x_n]$  unicamente determinados pelas condições  $f = ag + r$  com  $r(x) \neq 0$  e  $\text{grau}(r) < \text{grau}(g)$ .

Para a divisão em anéis polinomiais com várias variáveis, se faz necessário a utilização do algoritmo a seguir.

**entrada**  $f \in k[x_1, \dots, x_n]$  e  $g \in k[x_1, \dots, x_n] \setminus \{0\}$

**início**

$a \leftarrow 0$

$r \leftarrow 0$

$p \leftarrow f$

**enquanto**  $p \neq 0$  **faça**

**se**  $ml(g)$  divide  $ml(p)$  **então**

$a \leftarrow a + \frac{tl(p)}{tl(g)}$

$p \leftarrow p - \frac{tl(p)}{tl(g)}g$

**senão**

$r \leftarrow r + tl(p)$

$p \leftarrow p - tl(p)$

**fim enquanto**

**forneça**  $a, r$

**fim**

Tomando como exemplo os polinômios  $f = x^4 + x^3y + xy^4 + y^3$  e  $g = x^2 + y^3$ ,  $f$  será dividido por  $g$  segundo a ordem lexicográfica  $x \succ y$ . Em seguida, pela ordem lexicográfica graduada  $x \succ_{grlex} y$ .

Efetuada a divisão de  $f$  por  $g$ , utilizando a ordem lexicográfica  $x \succ y$ , nota-se as seguintes etapas:

- i. Inicialmente, atribui-se  $f$  a  $p$ , o divisor  $g$  é listado na chave e o quociente  $a$  e o resto  $r$  abaixo da chave. Dessa forma, apresenta-se o seguinte esquema.

$$\begin{array}{l|l}
 p : x^4 + x^3y + xy^4 + y^3 & g : x^2 + y^3 \\
 \hline
 & a : 0 \\
 & r : 0
 \end{array}$$

- ii. Observa-se que  $ml(g) = x^2$  divide  $ml(p) = x^4$ . Desse modo,  $a = a + \frac{tl(p)}{tl(g)} = x^2$  e

$$p = p - \frac{tl(p)}{tl(g)}g = x^3y - x^2y^3 + xy^4 + y^3. \text{ Dessa maneira, tem-se:}$$

$$\begin{array}{l|l}
 p : x^4 + x^3y + xy^4 + y^3 & g : x^2 + y^3 \\
 -(x^4 + x^2y^3) & \hline
 \hline
 x^3y - x^2y^3 + xy^4 + y^3 & a : x^2 \\
 & r : 0
 \end{array}$$

iii. Pode-se notar que  $ml(g) = x^2$  divide  $ml(p) = x^3y$ . Dessa forma,

$$a = a + \frac{tl(p)}{tl(g)} = xy \text{ e } p = p - \frac{tl(p)}{tl(g)}g = -x^2y^3 + y^3. \text{ Assim sendo, consta-se:}$$

$x^4 + x^3y + xy^4 + y^3$	$g : x^2 + y^3$
$-(x^4 + x^2y^3)$	$a : x^2 + xy$
$p : x^3y - x^2y^3 + xy^4 + y^3$	$r : 0$
$-(x^3y + xy^4)$	
$-x^2y^3 + y^3$	

iv. Percebe-se que  $ml(g) = x^2$  divide  $ml(p) = -x^2y^3$ . Dessa forma,

$$a = a + \frac{tl(p)}{tl(g)} = -y^3 \text{ e } p = p - \frac{tl(p)}{tl(g)}g = y^6 + y^3. \text{ Logo, tem-se:}$$

$x^4 + x^3y + xy^4 + y^3$	$g : x^2 + y^3$
$-(x^4 + x^2y^3)$	$a : x^2 + xy - y^3$
$x^3y - x^2y^3 + xy^4 + y^3$	$r : 0$
$-(x^3y + xy^4)$	
$p : -x^2y^3 + y^3$	
$-(-x^2y^3 - y^6)$	
$y^6 + y^3$	

- v. Como  $ml(g) = x^2$  não divide  $ml(p) = y^6$ . Assim,  $r = r + tl(p) = y^6$  e  $p = p - tl(p) = y^3$ . Logo, nota-se:

$$\begin{array}{r|l}
 x^4 + x^3y + xy^4 + y^3 & g : x^2 + y^3 \\
 \hline
 -(x^4 + x^2y^3) & a : x^2 + xy - y^3 \\
 \hline
 x^3y - x^2y^3 + xy^4 + y^3 & r : y^6 \\
 \hline
 -(x^3y + xy^4) & \\
 \hline
 -x^2y^3 + y^3 & \\
 \hline
 -(-x^2y^3 - y^6) & \\
 \hline
 p : \cancel{y^6} + y^3 & 
 \end{array}$$

- vi. Finalmente, como  $ml(g) = x^2$  não divide  $ml(p) = y^3$ . Assim sendo,  $r = r + tl(p) = y^6 + y^3$  e  $p = p - tl(p) = 0$ . Logo, tem-se:

$$\begin{array}{r|l}
 x^4 + x^3y + xy^4 + y^3 & g : x^2 + y^3 \\
 \hline
 -(x^4 + x^2y^3) & a : x^2 + xy - y^3 \\
 \hline
 x^3y - x^2y^3 + xy^4 + y^3 & r : y^6 + y^3 \\
 \hline
 -(x^3y + xy^4) & \\
 \hline
 -x^2y^3 + y^3 & \\
 \hline
 -(-x^2y^3 - y^6) & \\
 \hline
 p : \cancel{y^6} & 
 \end{array}$$

- vii. Com  $p = 0$ , o algoritmo da divisão é concluído. Assim sendo, nota-se que:

$$x^4 + x^3y + xy^4 + y^3 = (x^2 + xy - y^3) \cdot (x^2 + y^3) + y^6 + y^3.$$

Agora, efetuando a divisão de  $f$  por  $g$ , utilizando a ordem lexicográfica graduada  $x \succ_{grlex} y$ , tem-se as seguintes etapas:

- i. Inicialmente, atribui-se  $f$  a  $p$ , o divisor  $g$  é listado na chave e o quociente  $a$  e o resto  $r$  abaixo da chave.

$$\begin{array}{l|l}
 p : xy^4 + x^4 + x^3y + y^3 & g : y^3 + x^2 \\
 \hline
 & a : 0 \\
 & r : 0
 \end{array}$$

- ii. Observa-se que  $ml(g) = y^3$  divide  $ml(p) = xy^4$ . Dessa forma,  $a = a + \frac{tl(p)}{tl(g)} = xy$

e  $p = p - \frac{tl(p)}{tl(g)}g = x^4 + y^3$ . Por conseguinte, tem-se:

$$\begin{array}{l|l}
 p : xy^4 + x^4 + x^3y + y^3 & g : y^3 + x^2 \\
 -(xy^4 + x^3y) & a : xy \\
 \hline
 x^4 + y^3 & r : 0
 \end{array}$$

- iii. Pode-se notar que  $ml(g) = y^3$  não divide  $ml(p) = x^4$ . Assim sendo,  $r = r + tl(p) = x^4$  e  $p = p - tl(p) = y^3$ . Logo, tem-se:

$$\begin{array}{l|l}
 xy^4 + x^4 + x^3y + y^3 & g : y^3 + x^2 \\
 -(xy^4 + x^3y) & a : xy \\
 \hline
 p : x^4 + y^3 & r : x^4
 \end{array}$$

iv. Continuando, percebe-se que  $ml(g) = y^3$  divide  $ml(p) = y^3$ . Dessa forma,

$$a = a + \frac{tl(p)}{tl(g)} = xy + 1 \text{ e } p = p - \frac{tl(p)}{tl(g)}g = -x^2. \text{ Desse modo, tem-se:}$$

$xy^4 + x^4 + x^3y + y^3$	$g : y^3 + x^2$
$-(xy^4 + x^3y)$	$a : xy + 1$
$y^3$	$r : x^4$
$-(y^3 + x^2)$	
$p : -x^2$	

v. Finalmente, como  $ml(g) = y^3$  não divide  $ml(p) = x^2$ . Portanto,

$$r = r + tl(p) = x^4 - x^2 \text{ e } p = p - tl(p) = 0. \text{ Logo, tem-se:}$$

$xy^4 + x^4 + x^3y + y^3$	$g : y^3 + x^2$
$-(xy^4 + x^3y)$	$a : xy + 1$
$y^3$	$r : x^4 - x^2$
$-(y^3 + x^2)$	
$p : \cancel{x^2}$	

vi. Com  $p = 0$ , o algoritmo da divisão é concluído. Dessarte, tem-se que:

$$xy^4 + x^4 + x^3y + y^3 = (xy + 1) \cdot (y^3 + x^2) + x^4 - x^2.$$

Nos dois exemplos anteriores, pode-se observar que os resultados obtidos após a divisão de  $f$  por  $g$  são totalmente distintos, devido à alteração realizada na ordem monomial.

A partir deste ponto, um método que estrutura a divisão de um polinômio por vários outros polinômios ao mesmo tempo será apresentado: o algoritmo da pseudodivisão.

Inicialmente, fixa-se uma ordem monomial em  $\mathbb{Z}_{\geq 0}^n$  para um polinômio  $f$  e um conjunto ordenado  $(g_1, \dots, g_s)$  de polinômios em  $k[x_1, \dots, x_n]$ . O algoritmo da pseudodivisão é utilizado para determinação dos polinômios  $a_1, \dots, a_s$  e  $r$  em  $k[x_1, \dots, x_n]$  tal que  $f = a_1g_1 + \dots + a_sg_s + r$ , onde  $r = 0$  ou nenhum monômio em  $r$  é divisível por qualquer termo líder dos polinômios pertencentes ao conjunto  $(g_1, \dots, g_s)$ .

Este algoritmo pode ser descrito como:

```

entrada  $f \in k[x_1, \dots, x_n]$  e  $g_1, \dots, g_s \in k[x_1, \dots, x_n] \setminus \{0\}$ 
início
   $a_1 \leftarrow 0$ 
   $a_2 \leftarrow 0$ 
   $\vdots$ 
   $a_s \leftarrow 0$ 
   $r \leftarrow 0$ 
   $p \leftarrow f$ 
enquanto  $p \neq 0$  faça
  se  $\exists i \in \{1, \dots, s\}$  tal que  $tl(g_i)$  divide  $tl(p)$  então
    escolha o menor índice  $i$  com essa propriedade
    
$$a_i \leftarrow a_i + \frac{tl(p)}{tl(g_i)}$$

    
$$p \leftarrow p - \frac{tl(p)}{tl(g_i)} f$$

  senão
     $r \leftarrow r + tl(p)$ 
     $p \leftarrow p - tl(p)$ 
fim enquanto
forneça  $a_1, \dots, a_s$ 
fim

```



Analisando o algoritmo, nota-se que a variável  $p$  representa o dividendo intermediário a cada etapa, a variável  $r$  representa o resto e as variáveis  $a_1, \dots, a_s$  são os quocientes.

Tomando como exemplo os polinômios  $f = x^2y + xy^2 + y^2$ ,  $g_1 = xy - 1$  e  $g_2 = y^2 - 1$ , efetua-se a divisão de  $f$  por  $g_1$  e  $g_2$ , utilizando a ordem lexicográfica  $x \succ y$ , tem-se as seguintes etapas:

- i. Inicialmente, atribui-se  $f$  a  $p$ , os divisores  $g_1$  e  $g_2$  são listados na chave e os quocientes  $a_1$ ,  $a_2$  e o resto  $r$  abaixo da chave. Tem-se o seguinte esquema:

$$\begin{array}{l|l}
 p : x^2y + xy^2 + y^2 & g_1 : xy - 1 \\
 & g_2 : y^2 - 1 \\
 \hline
 & a_1 : 0 \\
 & a_2 : 0 \\
 & r : 0
 \end{array}$$

- ii. Observa-se que  $tl(g_1) = xy$  divide  $tl(p) = x^2y$ . Dessa forma,  $a_1 = a_1 + \frac{tl(p)}{tl(g_1)} = x$

e  $p = p - \frac{tl(p)}{tl(g_1)} f = xy^2 + x + y^2$ .

$$\begin{array}{l|l}
 p : x^2y + xy^2 + y^2 & g_1 : xy - 1 \\
 -(x^2y - x) & g_2 : y^2 - 1 \\
 \hline
 xy^2 + x + y^2 & a_1 : x \\
 & a_2 : 0 \\
 & r : 0
 \end{array}$$

iii. Verifica-se que  $tl(g_1) = xy$  divide  $tl(p) = xy^2$ . Então,  $a_1 = a_1 + \frac{tl(p)}{tl(g_1)} = x + y$  e

$$p = p - \frac{tl(p)}{tl(g_1)} f = x + y^2 + y.$$

$x^2y + xy^2 + y^2$	$g_1 : xy - 1$
$-(x^2y - x)$	$g_2 : y^2 - 1$
$p : xy^2 + x + y^2$	$a_1 : x + y$
$-(xy^2 - y)$	$a_2 : 0$
$x + y^2 + y$	$r : 0$

iv. Nota-se que  $tl(g_1) = xy$  não divide  $tl(p) = x$ , bem como  $tl(g_2) = y^2$  não divide  $tl(p) = x$ . Neste caso, move-se  $tl(p) = x$  para o resto  $r$ , ou seja,  $r = r + tl(p) = x$  e  $p = p - tl(p) = y^2 + y$ .

$x^2y + xy^2 + y^2$	$g_1 : xy - 1$
$-(x^2y - x)$	$g_2 : y^2 - 1$
$xy^2 + x + y^2$	$a_1 : x + y$
$-(xy^2 - y)$	$a_2 : 0$
$p : x + y^2 + y$	$r : x$

v. Observa-se, a partir desta etapa, que  $tl(g_1) = xy$  não divide  $tl(p) = y^2$ , todavia

$$tl(g_2) = y^2 \quad \text{divide} \quad tl(p) = y^2. \quad \text{Dessa forma} \quad a_2 = a_2 + \frac{tl(p)}{tl(g_2)} = 1 \quad \text{e}$$

$$p = p - \frac{tl(p)}{tl(g_2)} f = y + 1.$$

$x^2y + xy^2 + y^2$	$g_1 : xy - 1$
$-(x^2y - x)$	$g_2 : y^2 - 1$
<hr/> $xy^2 + x + y^2$	$a_1 : x + y$
$-(xy^2 - y)$	$a_2 : 1$
<hr/> $p : y^2 + y$	$r : x$
$-(y^2 - 1)$	
<hr/> $y + 1$	

vi. Nota-se que  $tl(g_1) = xy$  não divide  $tl(p) = y$ , bem como  $tl(g_2) = y^2$  não divide  $tl(p) = y$ . Neste caso, move-se  $tl(p) = y$  para o resto  $r$ , ou seja,  $r = r + tl(p) = x + y$  e  $p = p - tl(p) = 1$ .

$x^2y + xy^2 + y^2$	$g_1 : xy - 1$
$-(x^2y - x)$	$g_2 : y^2 - 1$
<hr/> $xy^2 + x + y^2$	$a_1 : x + y$
$-(xy^2 - y)$	$a_2 : 1$
<hr/> $y^2 + y$	$r : x + y$
$-(y^2 - 1)$	
<hr/> $p : y + 1$	

- vii. Novamente,  $tl(g_1) = xy$  não divide  $tl(p) = 1$ , bem como  $tl(g_2) = y^2$  não divide  $tl(p) = 1$ . Neste caso,  $tl(p) = 1$  é movido para o resto  $r$ , ou seja,  $r = r + tl(p) = x + y + 1$  e  $p = p - tl(p) = 0$ .

$$\begin{array}{r|l}
 x^2y + xy^2 + y^2 & g_1 : xy - 1 \\
 -(x^2y - x) & g_2 : y^2 - 1 \\
 \hline
 xy^2 + x + y^2 & a_1 : x + y \\
 -(xy^2 - y) & a_2 : 1 \\
 \hline
 y^2 + y & r : x + y + 1 \\
 -(y^2 - 1) & \\
 \hline
 p : \cancel{1} & 
 \end{array}$$

- viii. Como  $p = 0$ , o algoritmo da pseudodivisão é concluído. Dessarte, tem-se que:

$$x^2y + xy^2 + y^2 = (x + y) \cdot (xy - 1) + 1 \cdot (y^2 - 1) + (x + y + 1).$$

Tomando como um segundo exemplo, a divisão do polinômio  $f = x^2y - y$  por  $g_1 = x^2 - y$  e  $g_2 = xy - x$  será realizada utilizando a ordem lexicográfica graduada  $x \succ_{grlex} y$ . Em seguida, os cálculos serão refeitos invertendo a ordem de  $g_1$  e  $g_2$ . O desenvolvimento de todas as etapas destas duas divisões pode ser consultado no Apêndice A deste trabalho.

Efetuando a divisão de  $f$  por  $g_1$  e  $g_2$ , obtém-se:

$$\begin{array}{r|l}
 x^2y - y & g_1 : x^2 - y \\
 -(x^2y - y^2) & g_2 : xy - x \\
 \hline
 p : \cancel{y} & a_1 : y \\
 & a_2 : 0 \\
 & r : y^2 - y
 \end{array}$$

Dessa forma, tem-se que:  $x^2y - y = y \cdot (x^2 - y) + 0 \cdot (xy - x) + (y^2 - y)$ .

Partindo dos mesmos polinômios  $f$ ,  $g_1$  e  $g_2$ , utilizando a ordem lexicográfica graduada  $x \succ_{grlex} y$ , efetuando a divisão de  $f$  por  $g_2$  e  $g_1$ , obtém-se:

$$\begin{array}{r|l}
 x^2y - y & g_2 : xy - x \\
 -(x^2y - x^2) & g_1 : x^2 - y \\
 \hline
 p : x^2 - y & a_1 : x \\
 -(x^2 - y) & a_2 : 1 \\
 \hline
 0 & r : 0
 \end{array}$$

Desse modo, tem-se que:  $x^2y - y = x \cdot (xy - x) + 1 \cdot (x^2 - y)$ .

Nesse exemplo, pode-se observar que a ordem dos divisores interfere nos valores obtidos para  $a_1$  e  $a_2$ .

O polinômio  $f$  pertence ao ideal  $\langle g_1, g_2 \rangle$ , uma vez que, no segundo processo de divisão, obteve-se  $f = x \cdot (g_2) + 1 \cdot (g_1)$ . Porém, quando  $f$  foi dividido primeiro por  $g_1$  e, em seguida, por  $g_2$ , o resto obtido foi diferente de zero. Dessa forma, bastaria testar todas as  $s!$  possibilidades de se listar os polinômios  $g_1, \dots, g_s$  que geram o ideal  $I = \langle g_1, \dots, g_s \rangle$ . Contudo, o próximo exemplo demonstrará que independente de se obterem restos não nulos para todas as permutações possíveis de divisão, um polinômio pode pertencer ao ideal.

Tomando como um terceiro exemplo, a divisão do polinômio  $f = x^4y^2 - y^2$  por  $g_1 = x^2 - y$  e  $g_2 = xy - x$  será realizada utilizando a ordem lexicográfica graduada  $x \succ_{grlex} y$ . Em seguida, efetuando a divisão de  $f$  por  $g_1$  e  $g_2$ , obtém-se:

$$\begin{array}{r|l}
 x^4y^2 - y^2 & g_1 = x^2 - y \\
 -(x^4y^2 - x^2y^3) & g_2 = xy - x \\
 \hline
 x^2y^3 - y^2 & a_1 : x^2y^2 + y^3 \\
 -(x^2y^3 - y^4) & a_2 : 0 \\
 \hline
 p : \cancel{y^2} & r : y^4 - y^2
 \end{array}$$

Desse modo, tem-se:  $x^4y^2 - y^2 = (x^2y^2 + y^3) \cdot (x^2 - y) + 0 \cdot (xy - x) + (y^4 - y^2)$ .

Baseando-se nos mesmos polinômios  $f$ ,  $g_1$  e  $g_2$ , invertendo a ordem de  $g_1$  e  $g_2$ .

Efetuada a divisão de  $f$  por  $g_2$  e  $g_1$ , obtém-se:

$$\begin{array}{r|l}
 x^4y^2 - y^2 & g_2 = xy - x \\
 -(x^4y^2 - x^4y) & g_1 = x^2 - y \\
 \hline
 x^4y - y^2 & a_1 : x^3y + x^3 + x \\
 -(x^4y - x^4) & a_2 : x^2 + 1 \\
 \hline
 x^4 - y^2 & r : -y^2 + y \\
 -(x^4 - x^2y) & \\
 \hline
 x^2y - y^2 & \\
 -(x^2y - x^2) & \\
 \hline
 x^2 - y^2 & \\
 -(x^2 - y) & \\
 \hline
 p : \cancel{y} &
 \end{array}$$

Logo, tem-se:  $x^4y^2 - y^2 = (x^3y + x^3 + x) \cdot (xy - x) + (x^2 + 1) \cdot (x^2 - y) + y - y^2$ .

O desenvolvimento de todas as etapas destas duas divisões pode ser consultado no Apêndice A deste trabalho.

Sabe-se que se o resto da divisão de um polinômio  $f$  por um par de polinômios  $g_i, g_j \in \{g_1, \dots, g_n\}$  for nulo em alguma permutação escolhida. Então  $f$  pertence ao ideal, porém a recíproca não é verdadeira, uma vez que o polinômio  $f = x^4y^2 - y^2$  pode ser determinado com base em uma combinação contendo  $g_1 = x^2 - y$  e  $g_2 = xy - x$ , como por exemplo:  $f = (x^2y^2 + y)g_1 + (xy^2 + xy)g_2$ . Dessa forma, o polinômio  $f = x^4y^2 - y^2$  pertence ao ideal  $I = \langle x^2 - y, xy - x \rangle$ .

Na Seção 3.5, um tipo especial de geradores, conhecidos como bases de Gröbner de um ideal, será apresentado. Ele é fundamental para a solução do problema da pertinência.

### 3.5 ALGORITMO DE BUCHBERGER

O algoritmo para calcular uma base de Gröbner foi desenvolvido em 1965 por Bruno Buchberger. Esse método pode ser aplicado a um conjunto não nulo de polinômios de um dado ideal para produzir uma base de Gröbner (BUCHBERGER, 1965).

Mais formalmente, de acordo com Kendricks (2007), tem-se a seguinte definição: um conjunto de polinômios não nulos  $G = \{g_1, \dots, g_n\}$  contidos em um ideal  $I$ , é chamado de Base de Gröbner para  $I$  com respeito a uma dada ordem monomial  $\prec$ , se e somente se, para todo  $f \in I$ , com  $f \neq 0$ , existe  $i \in 1, \dots, n$ , de tal modo que  $ml(g_i)$  divide  $ml(f)$ .

O algoritmo de Buchberger constrói uma Base de Gröbner para um ideal com base em um conjunto finito de geradores, ou seja, dado um ideal  $I \subset k[x_1, \dots, x_n]$ , uma base de Gröbner pode ser efetivamente construída para  $I$ , baseando-se nesse algoritmo.

Segundo Cox, Little e O'Shea (1992), para desenvolver o Algoritmo de Buchberger, faz-se necessário entender o conceito de S-polinômio.

Tomando  $0 \neq f, g \in k[x_1, \dots, x_n]$ , e considerando  $L = mmc(ml(f), ml(g))$ , a combinação  $S(f, g) = \frac{L}{tl(f)}f - \frac{L}{tl(g)}g$  é denominada S-polinômio de  $f$  e  $g$ , em que  $(mmc)$  é o mínimo múltiplo comum entre os monômios líderes e pode ser calculado tomando o maior expoente de cada variável entre os monômios.

De acordo com o critério de Buchberger, considerando  $I \neq \{0\}$  um ideal polinomial, a base  $G = \{g_1, \dots, g_n\}$  de  $I$  é uma base de Gröbner de  $I$  se, e somente se, o resto da divisão de  $S(g_i, g_j)$  por  $G$ , listados em alguma ordem, é zero para todos os pares  $(i, j)$  com  $i \neq j$ .

Tomando como exemplo:  $f = x^3y^2 - x^2y^3 + x$ ,  $g = 3x^4y + y^2 \in \mathbb{R}[x, y]$ , com ordem lexicográfica  $x \succ y$ , encontra-se:  $S(f, g)$ . Para solucionar este problema, faz-se necessário determinar  $L = mmc(ml(f), ml(g)) = mmc(x^3y^2, x^4y) = x^4y^2$ . Logo:

$$\begin{aligned} S(f, g) &= \frac{x^4y^2}{x^3y^2}f - \frac{x^4y^2}{3x^4y}g, \\ S(f, g) &= xf - \frac{y}{3}g, \\ S(f, g) &= x(x^3y^2 - x^2y^3 + x) - \frac{y}{3}(3x^4y + y^2), \\ S(f, g) &= -x^3y^3 + x^2 - \frac{y^3}{3}. \end{aligned} \tag{11}$$

Dessa forma, um S-polinômio  $S(f, g)$  é designado para produzir o cancelamento de termos dominantes (ADAMS e LOUSTAUNAU, 1994). Além disso, de acordo com Cox, Little e O'Shea (1992), todo cancelamento de termos dominantes entre polinômios de mesmo *multigrau* resulta deste tipo de cancelamento.



Para verificar se uma base  $G = \{g_1, \dots, g_n\}$  para  $I$  é uma base de Gröbner, em que  $I$  é um ideal polinomial, faz-se necessário que, para todo par  $i \neq j$ , o resto da divisão de  $S(g_i, g_j)$  por  $G$ , listado em alguma ordem, seja igual a zero.

O algoritmo de Buchberger recebe como entrada um conjunto finito  $F = \{f_1, \dots, f_n\}$  de geradores de um ideal  $I$  de  $k[x_1, \dots, x_n]$  e fornece um conjunto  $G \supseteq F$ , que é uma Base de Gröbner em  $I$ .

Este algoritmo pode ser descrito como:

```

entrada  $F \subset k[x_1, \dots, x_n]$  e uma ordem monomial
início
   $G \leftarrow F$ 
   $P \leftarrow \{(g_i, g_j) \mid g_i, g_j \in G \text{ e } g_i \neq g_j\}$ 
  enquanto  $P \neq \emptyset$  faça
    escolha  $(g_i, g_j) \in P$ 
    remova  $(g_i, g_j)$  de  $P$ 
     $r \leftarrow$  resto da divisão de  $S(g_i, g_j)$  por  $(G)$ 
    se  $r \neq 0$ 
      então
         $G \leftarrow G \cup \{r\}$ 
         $P \leftarrow (g, r)$  para cada  $g \in G \setminus \{r\}$ 
    fim se
  fim enquanto
forneça  $G$ 
fim

```

Tomando como exemplo, será gerada uma base de Gröbner para o ideal  $\langle g_1 = x^2y - 1, g_2 = xy^2 - x \rangle$  sob a ordem lexicográfica graduada  $x \succ_{grlex} y$  pelo algoritmo de Buchberger. As seguintes etapas do algoritmo são seguidas:

- i. Atribui-se  $G = \{g_1 = x^2y - 1, g_2 = xy^2 - x\}$  e  $P = \{(g_1, g_2)\} \neq \emptyset$ .

- ii. Seleccionando  $(g_1, g_2)$ , tem-se que  $P = \emptyset$ . Dessa forma, sabendo que  $L = mmc(ml(g_1), ml(g_2)) = mmc(x^2y, xy^2) = x^2y^2$ , determina-se inicialmente o S-polinômio  $S\{g_1, g_2\}$ :

$$\begin{aligned} S\{g_1, g_2\} &= \frac{x^2y^2}{x^2y} g_1 - \frac{x^2y^2}{xy^2} g_2, \\ S\{g_1, g_2\} &= \frac{x^2y^2}{x^2y} (x^2y - 1) - \frac{x^2y^2}{xy^2} (xy^2 - x), \\ S\{g_1, g_2\} &= y(x^2y - 1) - x(xy^2 - x), \\ S\{g_1, g_2\} &= x^2y^2 - y - x^2y^2 + x^2, \\ S\{g_1, g_2\} &= x^2 - y. \end{aligned} \tag{12}$$

$S\{g_1, g_2\} = x^2 - y$  é dividido pelo par de polinômios  $g_1$  e  $g_2$ . Desse modo, tem-se a seguinte pseudodivisão:

$$\begin{array}{l|l} x^2 - y & g_1 = x^2y - 1 \\ -y & g_2 = xy^2 - x \\ \hline 0 & a_1 : 0 \\ & a_2 : 0 \\ & r : x^2 - y \end{array}$$

Como  $g_3 = r = x^2 - y \neq 0$ , então  $G = \{g_1 = x^2y - 1, g_2 = xy^2 - x, g_3 = x^2 - y\}$  e  $P = \{(g_1, g_3), (g_2, g_3)\} \neq \emptyset$ .

- iii. Seleccionando  $(g_2, g_3)$ , tem-se que  $P = \{(g_1, g_3)\} \neq \emptyset$ . Assim, sabendo que  $L = mmc(ml(g_2), ml(g_3)) = mmc(xy^2, x^2) = x^2y^2$ , determina-se inicialmente o S-polinômio  $S\{g_2, g_3\}$ :

$$\begin{aligned}
 S\{g_2, g_3\} &= \frac{x^2 y^2}{xy^2} g_2 - \frac{x^2 y^2}{x^2} g_3, \\
 S\{g_2, g_3\} &= \frac{x^2 y^2}{xy^2} (xy^2 - x) - \frac{x^2 y^2}{x^2} (x^2 - y), \\
 S\{g_2, g_3\} &= x(xy^2 - x) - y^2(x^2 - y), \\
 S\{g_2, g_3\} &= x^2 y^2 - x^2 - x^2 y^2 + y^3, \\
 S\{g_2, g_3\} &= y^3 - x^2.
 \end{aligned} \tag{13}$$

$S\{g_2, g_3\} = y^3 - x^2$  é dividido pelos polinômios  $g_1$ ,  $g_2$  e  $g_3$ . Dessa forma, tem-se a seguinte pseudodivisão:

$$\begin{array}{r|l}
 & g_1 = x^2 y - 1 \\
 & g_2 = xy^2 - x \\
 y^3 - x^2 & g_3 = x^2 - y \\
 \hline
 -x^2 & a_1 : 0 \\
 \hline
 -(-x^2 + y) & a_2 : 0 \\
 -y & a_3 : -1 \\
 0 & r : y^3 - y
 \end{array}$$

Como  $g_4 = r = y^3 - y \neq 0$ , então  $G = \{g_1, g_2, g_3, g_4 = y^3 - y\}$  e

$$P = \{(g_1, g_3), (g_1, g_4), (g_2, g_4), (g_3, g_4)\} \neq \emptyset.$$

- iv. Selecionando  $(g_3, g_4)$ , tem-se que  $P = \{(g_1, g_3), (g_1, g_4), (g_2, g_4)\} \neq \emptyset$ . Desse modo, sabendo que  $L = mmc(ml(g_3), ml(g_4)) = mmc(x^2, y^3) = x^2 y^3$ , determina-se inicialmente o S-polinômio  $S\{g_3, g_4\}$ :

$$\begin{aligned}
 S\{g_3, g_4\} &= \frac{x^2 y^3}{x^2} g_3 - \frac{x^2 y^3}{y^3} g_4, \\
 S\{g_3, g_4\} &= \frac{x^2 y^3}{x^2} (x^2 - y) - \frac{x^2 y^3}{y^3} (y^3 - y), \\
 S\{g_3, g_4\} &= y^3 (x^2 - y) - x^2 (y^3 - y), \\
 S\{g_3, g_4\} &= y^3 x^2 - y^4 - x^2 y^3 + x^2 y, \\
 S\{g_3, g_4\} &= x^2 y - y^4.
 \end{aligned} \tag{14}$$

$S\{g_3, g_4\} = x^2 y - y^4$  é dividido pelos polinômios  $g_1, g_2, g_3$  e  $g_4$ . Assim, tem-se a seguinte pseudodivisão:

$$\begin{array}{r|l}
 & g_1 = x^2 y - 1 \\
 & g_2 = x y^2 - x \\
 & g_3 = x^2 - y \\
 x^2 y - y^4 & g_4 = y^3 - y \\
 \hline
 -(x^2 y - 1) & a_1 : 1 \\
 \hline
 -y^4 + 1 & a_2 : 0 \\
 -(-y^4 + y^2) & a_3 : 0 \\
 \hline
 -y^2 + 1 & a_4 : -y \\
 1 & r : -y^2 + 1 \\
 0 &
 \end{array}$$

Como  $g_5 = r = -y^2 + 1 \neq 0$ , então  $G = \{g_1, g_2, g_3, g_4, g_5 = -y^2 + 1\}$  e

$$P = \{(g_1, g_3), (g_1, g_4), (g_1, g_5), (g_2, g_4), (g_2, g_5), (g_3, g_5), (g_4, g_5)\} \neq \emptyset.$$

v. Selecionando  $(g_4, g_5)$ , tem-se que

$$P = \{(g_1, g_3), (g_1, g_4), (g_1, g_5), (g_2, g_4), (g_2, g_5), (g_3, g_5)\} \neq \emptyset.$$

Dessa forma, sabendo que  $L = mmc(ml(g_4), ml(g_5)) = mmc(y^3, y^2) = y^3$ , determina-se inicialmente o

S-polinômio  $S\{g_4, g_5\}$ :

$$\begin{aligned}
S\{g_4, g_5\} &= \frac{y^3}{y^3} g_4 - \frac{y^3}{-y^2} g_5, \\
S\{g_4, g_5\} &= \frac{y^3}{y^3} (y^3 - y) - \frac{y^3}{-y^2} (-y^2 + 1), \\
S\{g_4, g_5\} &= (y^3 - y) + y(-y^2 + 1), \\
S\{g_4, g_5\} &= y^3 - y - y^3 + y, \\
S\{g_4, g_5\} &= 0.
\end{aligned} \tag{15}$$

Como  $S\{g_4, g_5\} = 0$ , não é necessário dividi-lo pelos polinômios  $g_1, g_2, g_3, g_4$  e  $g_5$ . Desse modo, tem-se:  $a_1 = 0, a_2 = 0, a_3 = 0, a_4 = 0, a_5 = 0$  e  $r = 0$ , então  $G = \{g_1, g_2, g_3, g_4, g_5\}$  e  $P = \{(g_1, g_3), (g_1, g_4), (g_1, g_5), (g_2, g_4), (g_2, g_5), (g_3, g_5)\} \neq \emptyset$ .

vi. Selecionando  $(g_3, g_5)$ , tem-se que  $P = \{(g_1, g_3), (g_1, g_4), (g_1, g_5), (g_2, g_4), (g_2, g_5)\} \neq \emptyset$ . Assim, sabendo que  $L = mmc(ml(g_3), ml(g_5)) = mmc(x^2, y^2) = x^2 y^2$ , determina-se inicialmente o S-polinômio  $S\{g_3, g_5\}$ :

$$\begin{aligned}
S\{g_3, g_5\} &= \frac{x^2 y^2}{x^2} g_3 - \frac{x^2 y^2}{-y^2} g_5, \\
S\{g_3, g_5\} &= \frac{x^2 y^2}{x^2} (x^2 - y) - \frac{x^2 y^2}{-y^2} (-y^2 + 1), \\
S\{g_3, g_5\} &= y^2 (x^2 - y) + x^2 (-y^2 + 1), \\
S\{g_3, g_5\} &= x^2 y^2 - y^3 - x^2 y^2 + x^2, \\
S\{g_3, g_5\} &= x^2 - y^3.
\end{aligned} \tag{16}$$

$S\{g_3, g_5\} = x^2 - y^3$  é dividido pelos polinômios  $g_1, g_2, g_3, g_4$  e  $g_5$ . Dessa forma, tem-se a seguinte pseudodivisão:

$$\begin{array}{r|l}
 & g_1 = x^2y - 1 \\
 & g_2 = xy^2 - x \\
 & g_3 = x^2 - y \\
 & g_4 = y^3 - y \\
 x^2 - y^3 & g_5 = -y^2 + 1 \\
 \hline
 -(x^2 - y) & a_1 : 0 \\
 \hline
 -y^3 + y & a_2 : 0 \\
 -(-y^3 + y) & a_3 : 1 \\
 \hline
 0 & a_4 : -1 \\
 & a_5 : 0 \\
 & r : 0
 \end{array}$$

Como  $r = 0$ , então  $G = \{g_1, g_2, g_3, g_4, g_5\}$  e  $P = \{(g_1, g_3), (g_1, g_4), (g_1, g_5), (g_2, g_4), (g_2, g_5)\} \neq \emptyset$ .

- vii. Selecionando  $(g_2, g_5)$ , tem-se que  $P = \{(g_1, g_3), (g_1, g_4), (g_1, g_5), (g_2, g_4)\} \neq \emptyset$ .  
 Dessa forma, sabendo que  $L = mmc(ml(g_2), ml(g_5)) = mmc(xy^2, y^2) = xy^2$ , determina-se inicialmente o S-polinômio  $S\{g_2, g_5\}$ :

$$\begin{aligned}
S\{g_2, g_5\} &= \frac{xy^2}{xy^2} g_2 - \frac{xy^2}{-y^2} g_5, \\
S\{g_2, g_5\} &= \frac{xy^2}{xy^2} (xy^2 - x) - \frac{xy^2}{-y^2} (-y^2 + 1), \\
S\{g_2, g_5\} &= (xy^2 - x) + x(-y^2 + 1), \\
S\{g_2, g_5\} &= xy^2 - x - xy^2 + x, \\
S\{g_2, g_5\} &= 0.
\end{aligned} \tag{17}$$

Como  $S\{g_2, g_5\} = 0$ , não é necessário dividi-lo pelos polinômios  $g_1, g_2, g_3, g_4$  e  $g_5$ . Por conseguinte, tem-se:  $a_1 = 0, a_2 = 0, a_3 = 0, a_4 = 0, a_5 = 0$  e  $r = 0$ .

Então  $G = \{g_1, g_2, g_3, g_4, g_5\}$  e  $P = \{(g_1, g_3), (g_1, g_4), (g_1, g_5), (g_2, g_4)\} \neq \emptyset$ .

- viii. Selecionando  $(g_2, g_4)$ , tem-se que  $P = \{(g_1, g_3), (g_1, g_4), (g_1, g_5)\} \neq \emptyset$ . Portanto, sabendo que  $L = mmc(ml(g_2), ml(g_4)) = mmc(xy^2, y^3) = xy^3$ , determina-se inicialmente o S-polinômio  $S\{g_2, g_4\}$ :

$$\begin{aligned}
S\{g_2, g_4\} &= \frac{xy^3}{xy^2} g_2 - \frac{xy^3}{y^3} g_4, \\
S\{g_2, g_4\} &= \frac{xy^3}{xy^2} (xy^2 - x) - \frac{xy^3}{y^3} (y^3 - y), \\
S\{g_2, g_4\} &= y(xy^2 - x) - x(y^3 - y), \\
S\{g_2, g_4\} &= xy^3 - xy - xy^3 + xy, \\
S\{g_2, g_4\} &= 0.
\end{aligned} \tag{18}$$

Como  $S\{g_2, g_4\} = 0$ , não é necessário dividi-lo pelos polinômios  $g_1, g_2, g_3, g_4$  e  $g_5$ . Dessarte, tem-se:  $a_1 = 0, a_2 = 0, a_3 = 0, a_4 = 0, a_5 = 0$  e  $r = 0$ . Então  $G = \{g_1, g_2, g_3, g_4, g_5\}$  e  $P = \{(g_1, g_3), (g_1, g_4), (g_1, g_5)\} \neq \emptyset$ .

- ix. Selecionando  $(g_1, g_5)$ , tem-se que  $P = \{(g_1, g_3), (g_1, g_4)\} \neq \emptyset$ . Dessa forma, sabendo que  $L = mmc(ml(g_1), ml(g_5)) = mmc(x^2y, y^2) = x^2y^2$ , determina-se inicialmente o S-polinômio  $S\{g_1, g_5\}$ :

$$\begin{aligned}
 S\{g_1, g_5\} &= \frac{x^2y^2}{x^2y} g_1 - \frac{x^2y^2}{-y^2} g_5, \\
 S\{g_1, g_5\} &= \frac{x^2y^2}{x^2y} (x^2y - 1) - \frac{x^2y^2}{-y^2} (-y^2 + 1), \\
 S\{g_1, g_5\} &= y(x^2y - 1) + x^2(-y^2 + 1), \\
 S\{g_1, g_5\} &= x^2y^2 - y - x^2y^2 + x^2, \\
 S\{g_1, g_5\} &= x^2 - y.
 \end{aligned} \tag{19}$$

$S\{g_1, g_5\} = x^2 - y$  é dividido pelos polinômios  $g_1, g_2, g_3, g_4$  e  $g_5$ . Por conseguinte, tem-se a seguinte pseudodivisão:

	$g_1 = x^2y - 1$
	$g_2 = xy^2 - x$
	$g_3 = x^2 - y$
	$g_4 = y^3 - y$
$x^2 - y$	$g_5 = -y^2 + 1$
$-(x^2 - y)$	$a_1 : 0$
<hr style="width: 100%;"/>	$a_2 : 0$
$0$	$a_3 : 1$
	$a_4 : 0$
	$a_5 : 0$
	$r : 0$

Como  $r = 0$ , então  $G = \{g_1, g_2, g_3, g_4, g_5\}$  e  $P = \{(g_1, g_3), (g_1, g_4)\} \neq \emptyset$ .

- x. Selecionando  $(g_1, g_4)$ , tem-se que  $P = \{(g_1, g_3)\} \neq \emptyset$ . Desse modo, sabendo que  $L = mmc(ml(g_1), ml(g_4)) = mmc(x^2y, y^3) = x^2y^3$ , determina-se inicialmente o S-polinômio  $S\{g_1, g_4\}$ :



$$\begin{aligned}
 S\{g_1, g_4\} &= \frac{x^2y^3}{x^2y}g_1 - \frac{x^2y^3}{y^3}g_4, \\
 S\{g_1, g_4\} &= \frac{x^2y^3}{x^2y}(x^2y - 1) - \frac{x^2y^3}{y^3}(y^3 - y), \\
 S\{g_1, g_4\} &= y^2(x^2y - 1) - x^2(y^3 - y), \\
 S\{g_1, g_4\} &= x^2y^3 - y^2 - x^2y^3 + x^2y, \\
 S\{g_1, g_4\} &= x^2y - y^2.
 \end{aligned} \tag{20}$$

$S\{g_1, g_4\} = x^2y - y^2$  é dividido pelos polinômios  $g_1, g_2, g_3, g_4$  e  $g_5$ . Portanto, tem-se:

	$g_1 = x^2y - 1$
	$g_2 = xy^2 - x$
	$g_3 = x^2 - y$
	$g_4 = y^3 - y$
$x^2y - y^2$	$g_5 = -y^2 + 1$
$-(x^2y - 1)$	$a_1 : 1$
$-y^2 + 1$	$a_2 : 0$
$-(-y^2 + 1)$	$a_3 : 0$
$0$	$a_4 : 0$
	$a_5 : 1$
	$r : 0$

Como  $r = 0$ , então  $G = \{g_1, g_2, g_3, g_4, g_5\}$  e  $P = \{(g_1, g_3)\} \neq \emptyset$ .

- xi. Selecionando  $(g_1, g_3)$ , tem-se que  $P = \emptyset$ . Destarte, sabendo que  $L = mmc(ml(g_1), ml(g_3)) = mmc(x^2y, x^2) = x^2y$ , determina-se inicialmente o S-polinômio  $S\{g_1, g_3\}$ :

$$\begin{aligned}
 S\{g_1, g_3\} &= \frac{x^2y}{x^2y} g_1 - \frac{x^2y}{x^2} g_3, \\
 S\{g_1, g_3\} &= \frac{x^2y}{x^2y} (x^2y - 1) - \frac{x^2y}{x^2} (x^2 - y), \\
 S\{g_1, g_3\} &= (x^2y - 1) - y(x^2 - y), \\
 S\{g_1, g_3\} &= x^2y - 1 - x^2y + y^2, \\
 S\{g_1, g_3\} &= y^2 - 1.
 \end{aligned} \tag{21}$$

$S\{g_1, g_3\} = y^2 - 1$  é dividido pelos polinômios  $g_1, g_2, g_3, g_4$  e  $g_5$ . Dessa forma, tem-se a seguinte pseudodivisão:

$$\begin{array}{r|l}
 & g_1 = x^2y - 1 \\
 & g_2 = xy^2 - x \\
 & g_3 = x^2 - y \\
 & g_4 = y^3 - y \\
 y^2 - 1 & g_5 = -y^2 + 1 \\
 \hline
 -(y^2 - 1) & a_1 : 0 \\
 \hline
 0 & a_2 : 0 \\
 & a_3 : 0 \\
 & a_4 : 0 \\
 & a_5 : -1 \\
 & r : 0
 \end{array}$$

Como  $r = 0$  e  $P = \emptyset$ , então  $G = \{g_1, g_2, g_3, g_4, g_5\}$  e o algoritmo de Buchberger é concluído.

Portanto, segundo o algoritmo de Buchberger, a base de Gröbner do ideal  $\langle g_1 = x^2y - 1, g_2 = xy^2 - x \rangle$  sob a ordem lexicográfica graduada  $x \succ_{grlex} y$  será  $G = \{x^2y - 1, xy^2 - x, x^2 - y, y^3 - y, -y^2 + 1\}$ .

Com o auxílio de um *software* de computação algébrica, uma base de Gröbner pode ser calculada mais rapidamente para conjuntos maiores de polinômios não nulos.

A título de exemplificação, todas as etapas do algoritmo de Buchberger para obtenção da base de Gröbner para o ideal  $\langle g_1 = x^3 - 2xy, g_2 = x^2y - 2y^2 + x \rangle$ , sob a ordem lexicográfica graduada  $x \succ_{grlex} y$ , exemplo citado por Cox, Little e O'Shea (1992), são apresentadas no Apêndice B deste trabalho.

De acordo com Boas e Marcos (2013), para polinômios com várias variáveis, a complexidade para a determinação de uma base de Gröbner não depende somente do número de polinômios no conjunto, porém sim da ordem dos termos.

Uma base de Gröbner com respeito a uma ordem lexicográfica  $x_1 \succ \dots \succ x_n$  produz um sistema em que as variáveis são sequencialmente eliminadas, até que, por fim, a última equação é expressa somente em função da variável  $x_n$ . Isso torna mais eficiente a resolução de um sistema polinomial (COX, LITTLE e O'SHEA, 1992).

Faugère (1999) apresentou um algoritmo eficiente para calcular uma base de Gröbner: o algoritmo F4; para evitar o máximo de operações intermediárias, o algoritmo calcula sucessivas bases de Gröbner truncadas e substitui a redução polinomial clássica encontrada no algoritmo de Buchberger pela redução simultânea de vários polinômios. Esse mecanismo de redução é alcançado por meio de cálculo simbólico e pelo uso extensivo de métodos de álgebra linear esparsa.

Atualmente, o algoritmo F4 é implementado nos sistemas de computação algébrica que incluem pacotes para lidar com Bases de Gröbner, incluindo o MAPLE – *software* utilizado para gerar as bases utilizadas neste trabalho.

## 4 CINEMÁTICA INVERSA DO STÄUBLI TS20 POR MEIO DO MÉTODO DE PAUL

O robô SCARA Stäubli TS20 é fabricado pela Stäubli Robotics e é normalmente utilizado para executar tarefas de linha de montagem. A sua escolha se dá por conta de sua geometria simplificada e pelo fato dele possuir diferentes tipos de juntas: três rotativas e uma prismática (Figura 4).



Figura 4: Robô SCARA Stäubli TS20  
Fonte: <http://www.staubli.com/in/robotics/>

Este robô possui quatro graus de liberdade, sendo a primeira junta rotativa, a segunda junta prismática e a terceira e quarta juntas rotativas (Figura 5). As características de cada junta desse manipulador robótico estão descritas a seguir.

- i. O eixo  $\theta$  tem uma rotação de  $(\pm 133^\circ)$  com velocidade máxima de  $1205^\circ / s$ ;
- ii. O eixo  $Z$  tem um movimento vertical de 100mm com velocidade máxima de 645mm/s;
- iii. O eixo  $\delta$  tem uma rotação de  $(\pm 131^\circ)$  com velocidade máxima de  $1715^\circ / s$ ;
- iv. O eixo  $\varphi$  (efetuador final) tem uma rotação de  $(\pm 250^\circ)$  com velocidade máxima de  $3760^\circ / s$ .

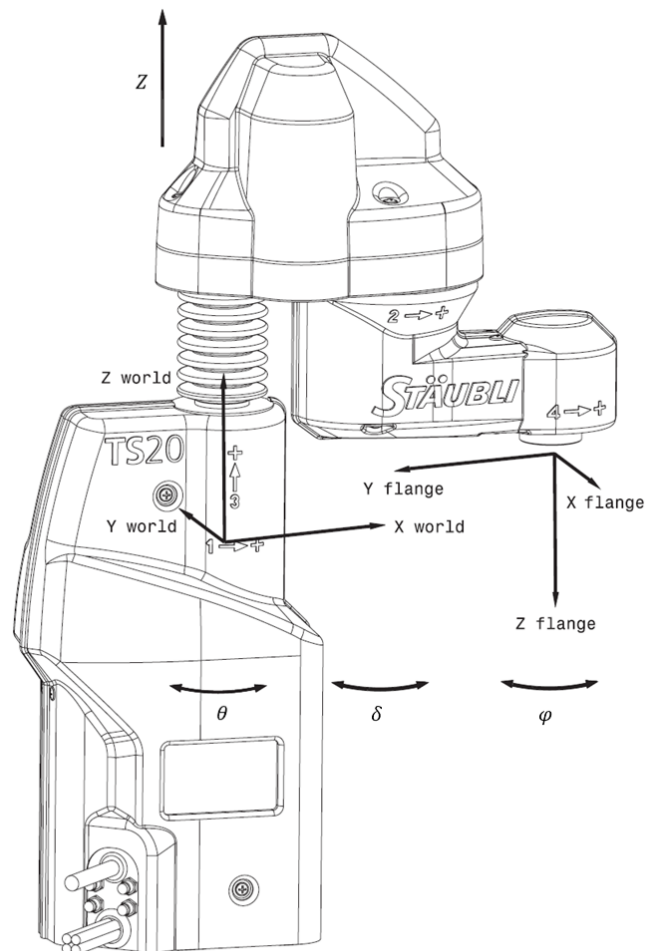


Figura 5: Juntas do robô SCARA Stäubli TS20

Fonte: <http://www.staubli.com/in/robotics/>

Uma vez que o interesse se resume ao posicionamento do robô e do respectivo efetuador no espaço, não será tratada a atuação propriamente dita do efetuador. Essa abordagem, de acordo com Kendricks (2007), elimina a necessidade de tratar o eixo  $\varphi$  em cálculos futuros.

Neste trabalho, as variáveis de junta  $\theta$ ,  $Z$  e  $\delta$  serão identificadas como  $\theta_1$ ,  $d_2$  e  $\theta_3$ , com o objetivo de padronizar com o Capítulo 2.

#### 4.1 CINEMÁTICA DIRETA DO SCARA STÄUBLI TS20

O robô SCARA Stäubli TS20 tem quatro graus de liberdade. No entanto, tal qual proposto anteriormente, o modelo cinemático a seguir incide sobre as três primeiras juntas, em que o último elo se refere ao efetuador final do robô. A Figura 6 ilustra as dimensões do robô Stäubli TS20, alguns desses valores serão utilizados nas matrizes e equações que serão construídas a partir deste ponto.

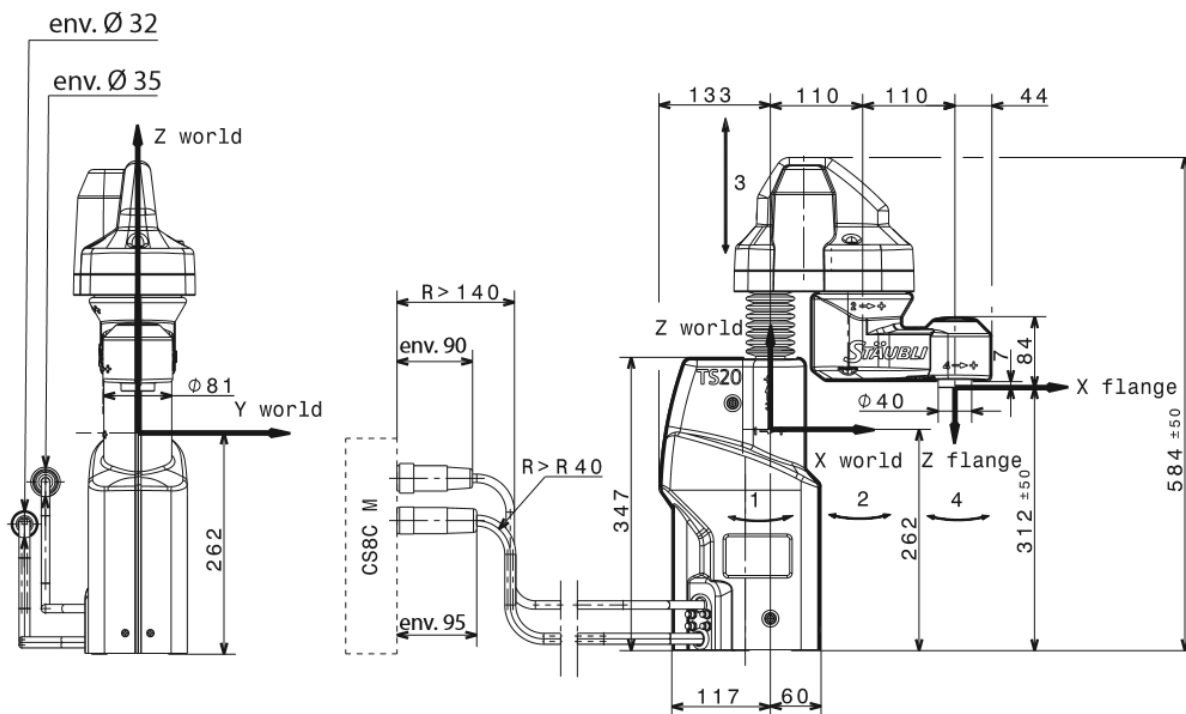


Figura 6: Dimensões do robô SCARA Stäubli TS20

Fonte: <http://www.staubli.com/in/robotics/>

Os sistemas de coordenadas em cada conjunto junta/elo são estabelecidos na Figura 7.

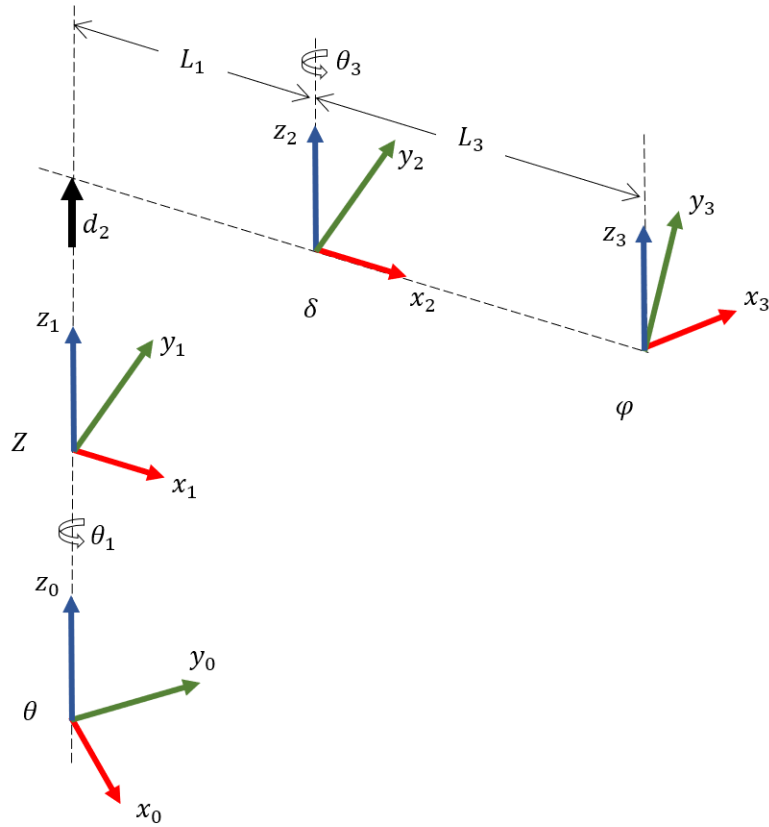


Figura 7: Sistemas de coordenadas para o robô SCARA Stäubli TS20

Em seguida, cada conjunto junta/elo é descrito em termos dos quatro parâmetros  $a_i$ ,  $d_i$ ,  $\alpha_i$  e  $\theta_i$ , apresentados na Seção 2.1. A Tabela 2 descreve os parâmetros D-H do robô SCARA Stäubli TS20.

Tabela 2: Parâmetros D-H do robô SCARA Stäubli TS20

Junta, $i$	$a_i$	$d_i$	$\alpha_i$	$\theta_i$
1	0	347 mm	0	$\theta_1$
2	110 mm	$d_2$	0	0
3	110 mm	0	0	$\theta_3$

As informações apresentadas na Tabela 2 são utilizadas para formar as matrizes de transformação  $A_{01}$ ,  $A_{12}$  e  $A_{23}$ . Utiliza-se  $s_k$  para  $\text{sen}(\theta_k)$  e  $c_k$  para  $\text{cos}(\theta_k)$ .

$$A_{01} = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 347 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (22)$$

$$A_{12} = \begin{bmatrix} 1 & 0 & 0 & 110 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (23)$$

$$A_{23} = \begin{bmatrix} c_3 & -s_3 & 0 & 110c_3 \\ s_3 & c_3 & 0 & 110s_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (24)$$

Multiplicando as matrizes de transformação  $A_{01}$ ,  $A_{12}$  e  $A_{23}$ :

$$\begin{aligned} T &= A_{01} \cdot A_{12} \cdot A_{23} = A_{01} \cdot (A_{12} \cdot A_{23}) \\ &= \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 347 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \left( \begin{bmatrix} 1 & 0 & 0 & 110 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} c_3 & -s_3 & 0 & 110c_3 \\ s_3 & c_3 & 0 & 110s_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \right) \\ &= \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 347 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} c_3 & -s_3 & 0 & 110c_3 + 110 \\ s_3 & c_3 & 0 & 110s_3 \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \end{aligned} \quad (25)$$

A cinemática direta da posição do efetuador do robô SCARA Stäubli TS20 é obtida pela matriz (26).

$$T = \begin{bmatrix} -s_1s_3 + c_1c_3 & -c_1s_3 - s_1c_3 & 0 & 110c_1c_3 - 110s_1s_3 + 110c_1 \\ s_1c_3 + c_1s_3 & -s_1s_3 + c_1c_3 & 0 & 110s_1c_3 + 110c_1s_3 + 110s_1 \\ 0 & 0 & 1 & 347 + d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (26)$$

Multiplicando a matriz  $T$  por  $[0 \ 0 \ 0 \ 1]^T$ , obtém-se:



$$P = \begin{bmatrix} 110c_1c_3 - 110s_1s_3 + 110c_1 \\ 110s_1c_3 + 110c_1s_3 + 110s_1 \\ 347 + d_2 \\ 1 \end{bmatrix}. \quad (27)$$

Os três primeiros elementos da matriz coluna  $P$  representam a posição do efetuador:  $x = 110(c_1c_3 - s_1s_3 + c_1)$ ,  $y = 110(s_1c_3 + c_1s_3 + s_1)$  e  $z = 347 + d_2$ .

De acordo com Spong e Vidyasagar (2008), para esta família de manipuladores seriais, a cinemática direta também pode ser solucionada de forma simplificada com base no método geométrico. Como se pode observar na Figura 8, os elos do robô podem ser projetados nos eixos  $x$  e  $y$  do sistema cartesiano.

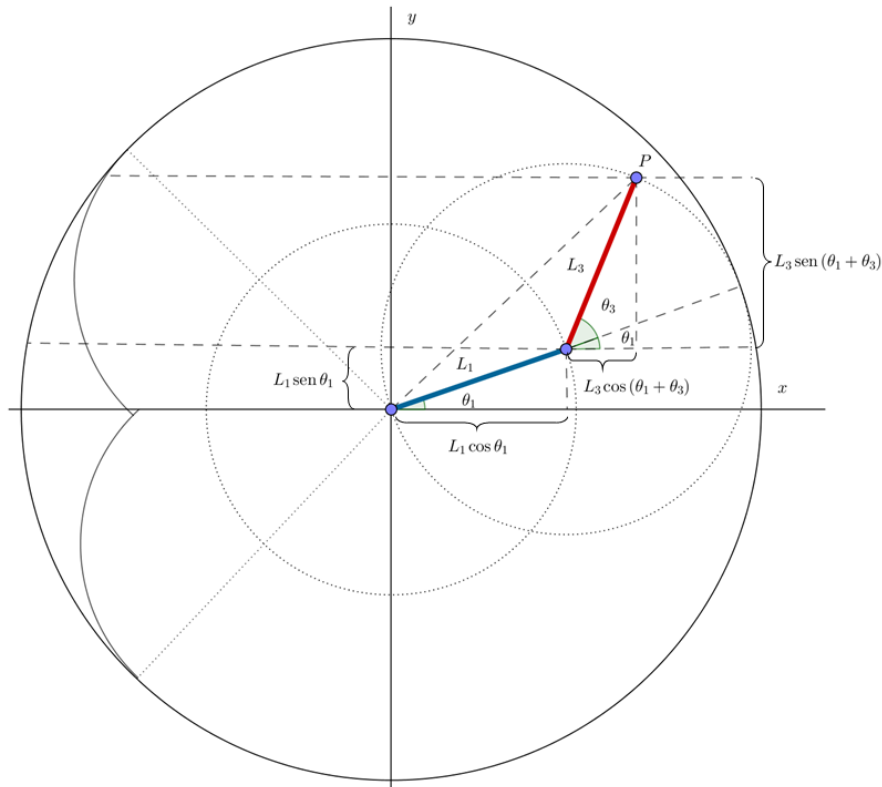


Figura 8: Solução da cinemática direta pelo método geométrico

Com base nos valores das projeções de  $L_1$  e  $L_3$ , conclui-se que o efetuador final do robô, dado pelo ponto  $P$ , é determinado por:

$$P = (L_1 c_1 + L_3 c_{13}; L_1 s_1 + L_3 s_{13}). \quad (28)$$

Utiliza-se  $s_{13}$  para  $\sin(\theta_1 + \theta_3)$  e  $c_{13}$  para  $\cos(\theta_1 + \theta_3)$ .

Considerando  $L_1 = L_3 = 110$ , tem-se:

$$P = (110c_1 + 110c_{13}; 110s_1 + 110s_{13}). \quad (29)$$

Sabendo-se que:  $c_{13}$  é igual a  $(c_1 c_3 - s_1 s_3)$  e que  $s_{13}$  é igual a  $(s_1 c_3 + s_3 c_1)$ ; constata-se que os dados obtidos na equação (29) são exatamente iguais aos obtidos nos dois primeiros elementos da última coluna da matriz (27).

Pela geometria do manipulador robótico, observa-se que a segunda junta desloca-se na direção do eixo  $z$ , as outras juntas não alteram a coordenada  $z$  do efetuador final, pois as rotações ocorrem somente no plano  $xy$  (com eixo de rotação sendo o eixo  $z$ ). Dessa forma, é possível determinar imediatamente que  $z = d_2 + 347$ .

A partir deste ponto, o problema da cinemática inversa será solucionado para o robô SCARA Stäubli TS20 utilizando o método de Paul por meio de Denavit-Hartenberg.

## 4.2 CINEMÁTICA INVERSA DO SCARA STÄUBLI TS20

Para resolver o problema da cinemática inversa do robô SCARA Stäubli TS20, determina-se inicialmente a matriz (30) obtida na Seção 4.1.

$$T = \begin{bmatrix} c_1 c_3 - s_1 s_3 & -c_1 s_3 - s_1 c_3 & 0 & 110(c_1 c_3 - s_1 s_3 + c_1) \\ c_1 s_3 + s_1 c_3 & -s_1 s_3 + c_1 c_3 & 0 & 110(s_1 c_3 + c_1 s_3 + s_1) \\ 0 & 0 & 1 & 347 + d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (30)$$

Igualando a matriz (1) à (30), obtém-se:

$$\begin{aligned}
& \begin{bmatrix} n_x & o_x & a_x & x \\ n_y & o_y & a_y & y \\ n_z & o_z & a_z & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
& = \begin{bmatrix} c_1 c_3 - s_1 s_3 & -c_1 s_3 - s_1 c_3 & 0 & 110(c_1 c_3 - s_1 s_3 + c_1) \\ c_1 s_3 + s_1 c_3 & -s_1 s_3 + c_1 c_3 & 0 & 110(s_1 c_3 + c_1 s_3 + s_1) \\ 0 & 0 & 1 & 347 + d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}.
\end{aligned} \tag{31}$$

Pretende-se encontrar os valores de  $\theta_1$ ,  $d_2$  e  $\theta_3$  que satisfaçam essas equações na quarta coluna representando  $x$ ,  $y$  e  $z$ . Ou seja, busca-se determinar os valores das juntas que permitirão o efetuador final alcançar o ponto descrito no espaço por:

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} 110(c_1 c_3 - s_1 s_3 + c_1) \\ 110(s_1 c_3 + c_1 s_3 + s_1) \\ 347 + d_2 \\ 1 \end{bmatrix}. \tag{32}$$

Para resolver o problema da cinemática inversa, utiliza-se o método de Paul (1981). Baseando-se neste algoritmo complementar, cada membro da equação  $T = A_{01} \cdot A_{12} \cdot A_{23}$ , apresentada na Seção 4.1, é multiplicado à esquerda por  $A_{01}^{-1}$  de modo que:

$$A_{01}^{-1} \cdot T = A_{01}^{-1} \cdot A_{01} \cdot A_{12} \cdot A_{23} = A_{12} \cdot A_{23}, \tag{33}$$

$$\begin{bmatrix} c_1 & s_1 & 0 & 0 \\ -s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & -347 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} n_x & o_x & a_x & x \\ n_y & o_y & a_y & y \\ n_z & o_z & a_z & z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_3 & -s_3 & 0 & 110c_3 + 110 \\ s_3 & c_3 & 0 & 110s_3 \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{34}$$

Efetuando a multiplicação no primeiro membro da igualdade, tem-se:

$$\begin{aligned}
& \begin{bmatrix} n_x c_1 + n_y s_1 & o_x c_1 + o_y s_1 & a_x c_1 + a_y s_1 & x c_1 + y s_1 \\ -n_x s_1 + n_y c_1 & -o_x s_1 + o_y c_1 & -a_x s_1 + a_y c_1 & -x s_1 + y c_1 \\ n_z & o_z & a_z & z - 347 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
& = \begin{bmatrix} c_3 & -s_3 & 0 & 110c_3 + 110 \\ s_3 & c_3 & 0 & 110s_3 \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}.
\end{aligned} \tag{35}$$

Esta igualdade de matrizes se traduz para igualdades entre os seus elementos.

Particularmente, para os elementos da quarta coluna tem-se:

$$x c_1 + y s_1 = 110c_3 + 110, \tag{36}$$

$$-x s_1 + y c_1 = 110s_3, \tag{37}$$

$$z - 347 = d_2. \tag{38}$$

Dessa forma,  $d_2 = z - 347$ , uma das três variáveis desconhecidas, é determinada.

No entanto, ainda é necessário determinar os ângulos  $\theta_1$  e  $\theta_3$ .

Elevando ao quadrado ambos os membros da Equação (36), obtém-se:

$$(x c_1 + y s_1)^2 = (110c_3 + 110)^2, \tag{39}$$

$$x^2 (c_1)^2 + 2x y c_1 s_1 + y^2 (s_1)^2 = 110^2 (c_3)^2 + 2(110)^2 c_3 + 110^2. \tag{40}$$

Elevando também ao quadrado ambos os membros da Equação (37), tem-se:

$$(-x s_1 + y c_1)^2 = (110s_3)^2, \tag{41}$$

$$x^2 (s_1)^2 - 2x y c_1 s_1 + y^2 (c_1)^2 = 110^2 (s_3)^2. \tag{42}$$

Em seguida, adicionando as equações (40) e (42), obtém-se a Equação (43).

$$x^2 + y^2 = 2(110)^2 (c_3 + 1) \tag{43}$$

Dessa forma:

$$c_3 = \frac{x^2 + y^2}{2(110)^2} - 1. \tag{44}$$

Destarte, o ângulo  $\theta_3$  pode ser determinado. Para isso,  $c_3$  é substituído na Equação (40), obtendo-se (45). Em seguida, adiciona-se o resultado à Equação (42) e obtém-se a Equação (46).

$$x^2 (c_1)^2 + 2xyc_1s_1 + y^2 (s_1)^2 = \left( \frac{x^2 + y^2}{220} \right)^2 \quad (45)$$

$$\begin{aligned} x^2 (c_1)^2 + 2xyc_1s_1 + y^2 (s_1)^2 + x^2 (s_1)^2 - 2xyc_1s_1 + y^2 (c_1)^2 \\ = \left( \frac{x^2 + y^2}{220} \right)^2 + 110^2 (s_3)^2 \end{aligned} \quad (46)$$

Simplificando a Equação (46), obtém-se:

$$x^2 + y^2 = \left( \frac{x^2 + y^2}{220} \right)^2 + 110^2 (s_3)^2. \quad (47)$$

Então:

$$s_3 = \pm \frac{\sqrt{x^2 + y^2 - \left( \frac{x^2 + y^2}{220} \right)^2}}{110}. \quad (48)$$

Logo, considerando  $\tan \theta_3 = \left( \frac{s_3}{c_3} \right)$ , tem-se:

$$\theta_3 = \arctan \left( \pm \frac{\sqrt{x^2 + y^2 - \left( \frac{x^2 + y^2}{220} \right)^2}}{\frac{x^2 + y^2}{2(110)^2} - 1} \right). \quad (49)$$

Dessa forma, obtém-se  $\theta_3$ :

$$\theta_3 = \arctan \left( \pm \frac{\sqrt{-(x^2 + y^2)(x^2 + y^2 - (220^2))}}{x^2 + y^2 - 2(110)^2} \right), -\frac{\pi}{2} < \theta_3 < \frac{\pi}{2}. \quad (50)$$

Uma vez determinado  $\theta_3$ ,  $\theta_1$  pode ser encontrado. Utilizando as Equações (36) e (37), trabalha-se com as incógnitas  $\theta_1$  e  $\theta_3$ .

$$xc_1 + ys_1 = 110c_3 + 110 \quad (51)$$

$$-xs_1 + yc_1 = 110s_3 \quad (52)$$

Usando eliminação, o sistema de equações é resolvido, obtendo as Equações (53) e (54).

$$s_1 = \frac{y(110c_3 + 110) - x110s_3}{x^2 + y^2} \quad (53)$$

$$c_1 = \frac{x(110c_3 + 110) + y110s_3}{x^2 + y^2} \quad (54)$$

Dessa forma, considerando  $\tan \theta_1 = \left( \frac{s_1}{c_1} \right)$ , tem-se:

$$\theta_1 = \arctan \left( \frac{y(110c_3 + 110) - x110s_3}{x(110c_3 + 110) + y110s_3} \right), -\frac{\pi}{2} < \theta_1 < \frac{\pi}{2}. \quad (55)$$

Simplificando, obtém-se:

$$\theta_1 = \arctan \left( \frac{-xs_3 + yc_3 + y}{xc_3 + ys_3 + x} \right), -\frac{\pi}{2} < \theta_1 < \frac{\pi}{2}. \quad (56)$$

Como  $s_3$  pode assumir dois valores na Equação (48),  $\theta_1$  também assume dois valores, conforme se pode observar nas Equações (57) e (58).

$$\theta_1 = \arctan \left( \frac{-x \left( \frac{\sqrt{x^2 + y^2 - \left( \frac{x^2 + y^2}{220} \right)^2}}{110} \right) + \left( \frac{x^2 + y^2}{2(110)^2} - 1 \right) y + y}{x \left( \frac{x^2 + y^2}{2(110)^2} - 1 \right) + y \left( \frac{\sqrt{x^2 + y^2 - \left( \frac{x^2 + y^2}{220} \right)^2}}{110} \right) + x} \right), -\frac{\pi}{2} < \theta_1 < \frac{\pi}{2} \quad (57)$$

$$\theta_1 = \arctan \frac{x \left( \frac{\sqrt{x^2 + y^2 - \left(\frac{x^2 + y^2}{220}\right)^2}}{110} + \left(\frac{x^2 + y^2}{2(110)^2} - 1\right) y + y \right)}{x \left(\frac{x^2 + y^2}{2(110)^2} - 1\right) - y \left( \frac{\sqrt{x^2 + y^2 - \left(\frac{x^2 + y^2}{220}\right)^2}}{110} + x \right)}, -\frac{\pi}{2} < \theta_1 < \frac{\pi}{2} \quad (58)$$

Simplificando (57) e (58), obtêm-se as Equações (59) e (60).

$$\theta_1 = \arctan \frac{y(x^2 + y^2) - x\sqrt{-(x^2 + y^2)(x^2 + y^2 - 220^2)}}{x(x^2 + y^2) + y\sqrt{-(x^2 + y^2)(x^2 + y^2 - 220^2)}}, -\frac{\pi}{2} < \theta_1 < \frac{\pi}{2} \quad (59)$$

$$\theta_1 = \arctan \frac{y(x^2 + y^2) + x\sqrt{-(x^2 + y^2)(x^2 + y^2 - 220^2)}}{x(x^2 + y^2) - y\sqrt{-(x^2 + y^2)(x^2 + y^2 - 220^2)}}, -\frac{\pi}{2} < \theta_1 < \frac{\pi}{2} \quad (60)$$

Analisando as Equações (59) e (60), pode-se observar que soluções não podem ser obtidas para  $\theta_1$  quando o denominador se anula e quando o argumento da raiz quadrada é negativo.

O denominador claramente se anula quando  $x = 0$  e  $y = 0$ , ou seja, no ponto  $(0,0)$ . Supondo agora que  $x \neq 0$  ou  $y \neq 0$ , tem-se que o denominador se anula quando:

$$x(x^2 + y^2) = \pm y\sqrt{-(x^2 + y^2)(x^2 + y^2 - 220^2)}. \quad (61)$$

Dessa forma, tem-se:

$$\begin{aligned}
\left(x(x^2 + y^2)\right)^2 &= \left(\pm y \sqrt{-(x^2 + y^2)(x^2 + y^2 - 220^2)}\right)^2, \\
x^2(x^2 + y^2)^2 &= -y^2(x^2 + y^2)(x^2 + y^2 - 220^2), \\
x^2(x^2 + y^2) &= -y^2(x^2 + y^2 - 220^2), \\
x^4 + x^2y^2 &= -x^2y^2 - y^4 + 220^2y^2, \\
x^4 + x^2y^2 + x^2y^2 + y^4 &= 220^2y^2, \\
x^4 + 2x^2y^2 + y^4 &= 220^2y^2, \\
(x^2 + y^2)^2 &= 220^2y^2, \\
x^2 + (y - 110)^2 &= 110^2.
\end{aligned} \tag{62}$$

Portanto,  $x^2 + (y - 110)^2 = 110^2$  representa uma circunferência com centro em  $(0, 110)$  e raio 110.

Com relação ao argumento da raiz quadrada, o mesmo é negativo somente quando  $x^2 + y^2 - 220^2 > 0$ , isto é, fora da circunferência com centro em  $(0, 0)$  e raio 220.

Analisando a Equação (50), pode-se observar que soluções não podem ser obtidas para  $\theta_3$  quando o denominador se anula, ou seja,  $x^2 + y^2 - 2(110)^2 = 0$ . Dessa forma, tem-se:  $x^2 + y^2 = 2(110)^2$ , uma circunferência com centro em  $(0, 0)$  e raio  $110\sqrt{2}$ .

Analisando a Figura 9, pode-se observar os pontos inatingíveis pelo efetuador do manipulador SCARA Stäubli TS20, de acordo com as Equações (50), (59) e (60). A região descrita pela inequação  $x^2 + y^2 - 220^2 > 0$  representa uma região que é efetivamente externa ao volume de trabalho do manipulador. Dessa maneira, inatingível pelo efetuador, porém todos os pontos pertencentes às duas circunferências descritas pelas equações  $x^2 + y^2 = 220y$  e  $x^2 + y^2 = 2(110)^2$ , também pertencem ao volume de trabalho do robô e podem ser atingidos.



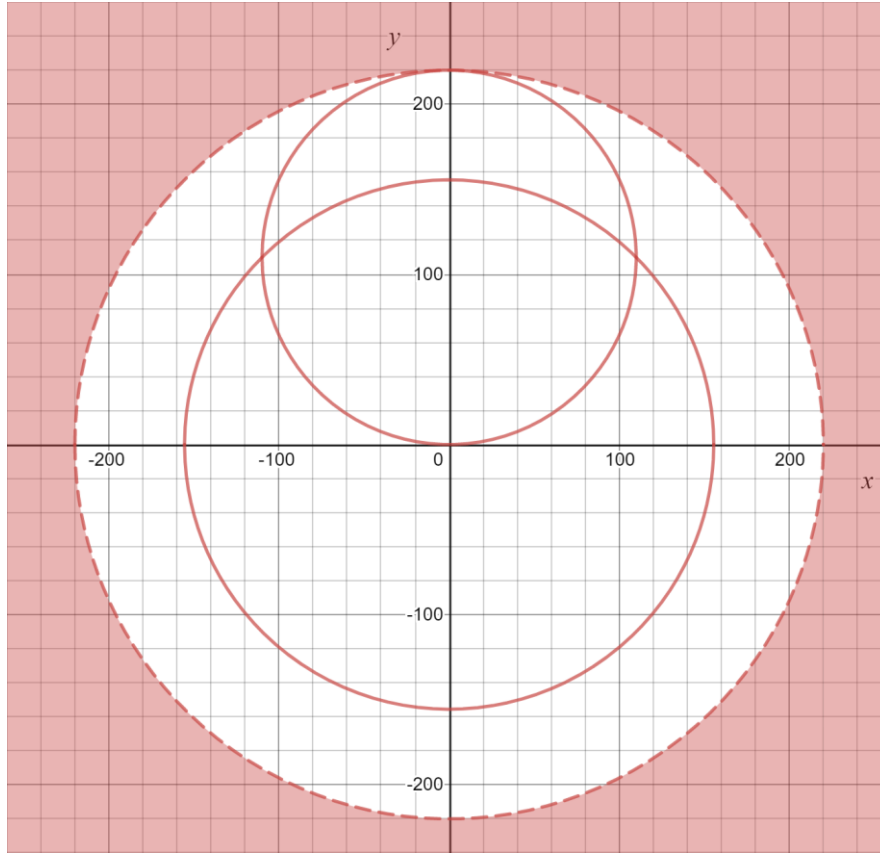


Figura 9: Representação dos pontos inatingíveis do SCARA Stäubli TS20 com base nas equações da cinemática inversa

Esta limitação observada nas Equações (50), (59) e (60) está associada aos ângulos  $\frac{\pi}{2}$  e  $-\frac{\pi}{2}$ , para os quais a tangente não está definida. Com isso, justifica-se a utilização da função  $\text{Atan2}$ . Informações mais detalhadas sobre a função  $\text{Atan2}$  são apresentadas no Apêndice C deste trabalho.

Dessa forma, tem-se:

$$\theta_1 = \text{Atan2} \left( \begin{array}{l} \left( y(x^2 + y^2) \mp x \sqrt{-(x^2 + y^2)(x^2 + y^2 - 220^2)} \right) \\ \left( x(x^2 + y^2) \pm y \sqrt{-(x^2 + y^2)(x^2 + y^2 - 220^2)} \right) \end{array} \right), \quad (63)$$

$$d_2 = z - 347, \quad (64)$$

$$\theta_3 = \text{Atan2} \left( \pm \sqrt{-(x^2 + y^2)(x^2 + y^2 - (220^2))}, (x^2 + y^2 - 2(110^2)) \right). \quad (65)$$

Dessarte, as variáveis de junta ( $\theta_1, d_2$  e  $\theta_3$ ) podem ser determinadas, o que permitirá ao efetuador final alcançar o ponto  $(x; y; z)$  no espaço.

De acordo com Spong e Vidyasagar (2008), para essa família de manipuladores seriais, a cinemática inversa também pode ser solucionada de forma simplificada com base no método geométrico.

Pela Figura 10, é possível determinar o valor de  $m$ , hipotenusa do triângulo  $ABC$ , em função de  $L_1$ ,  $L_3$  e  $\theta_3$ .

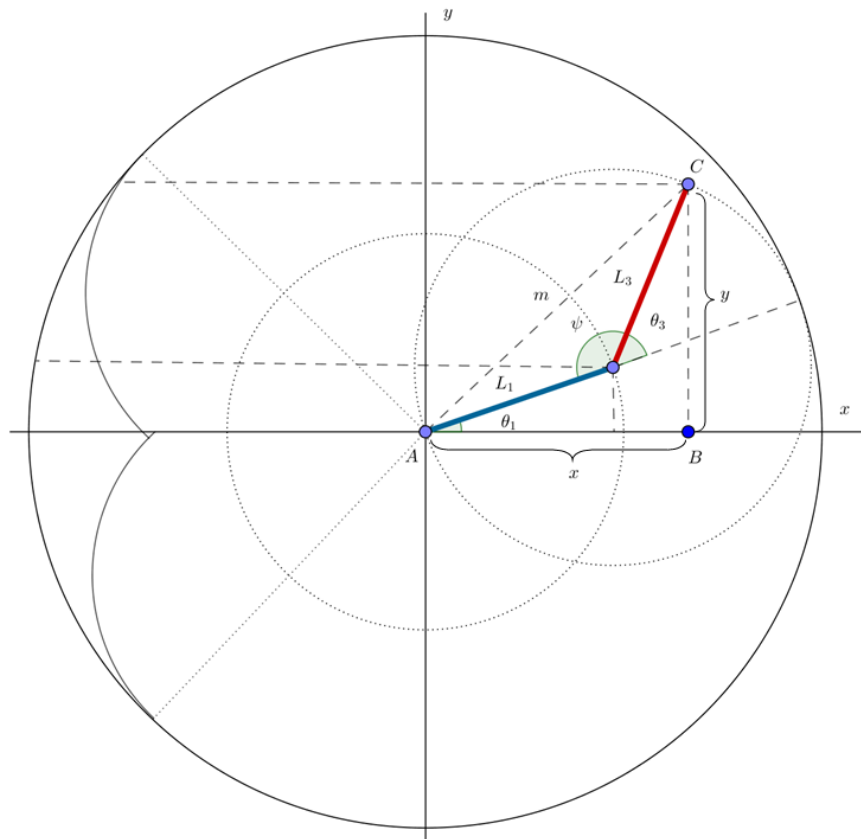


Figura 10: Determinação de  $\theta_3$  pelo método geométrico

Utilizando a lei dos cossenos, tem-se:

$$m^2 = L_1^2 + L_3^2 - 2L_1L_3c_\psi. \quad (66)$$

Como  $\psi$  e  $\theta_3$  são ângulos suplementares,  $c_\psi = -c_3$ . Desse modo, tem-se:

$$m^2 = L_1^2 + L_3^2 + 2L_1L_3(c_3). \quad (67)$$

Aplica-se o teorema de Pitágoras para o triângulo  $ABC$  (Equação (68)).

$$m^2 = x^2 + y^2 \quad (68)$$

Igualando as Equações (67) e (68), tem-se:

$$L_1^2 + L_3^2 + 2L_1L_3c_3 = x^2 + y^2. \quad (69)$$

Pode-se determinar  $\cos \theta_3$ :

$$c_3 = \frac{x^2 + y^2 - L_1^2 - L_3^2}{2L_1L_3}. \quad (70)$$

Considerando que ( $s_3^2 + c_3^2 = 1$ ), obtém-se:

$$s_3 = \pm \sqrt{1 - \left( \frac{x^2 + y^2 - L_1^2 - L_3^2}{2L_1L_3} \right)^2}. \quad (71)$$

Logo, tendo em vista  $\tan \theta_3 = \left( \frac{s_3}{c_3} \right)$ , é possível determinar  $\theta_3$  com base na

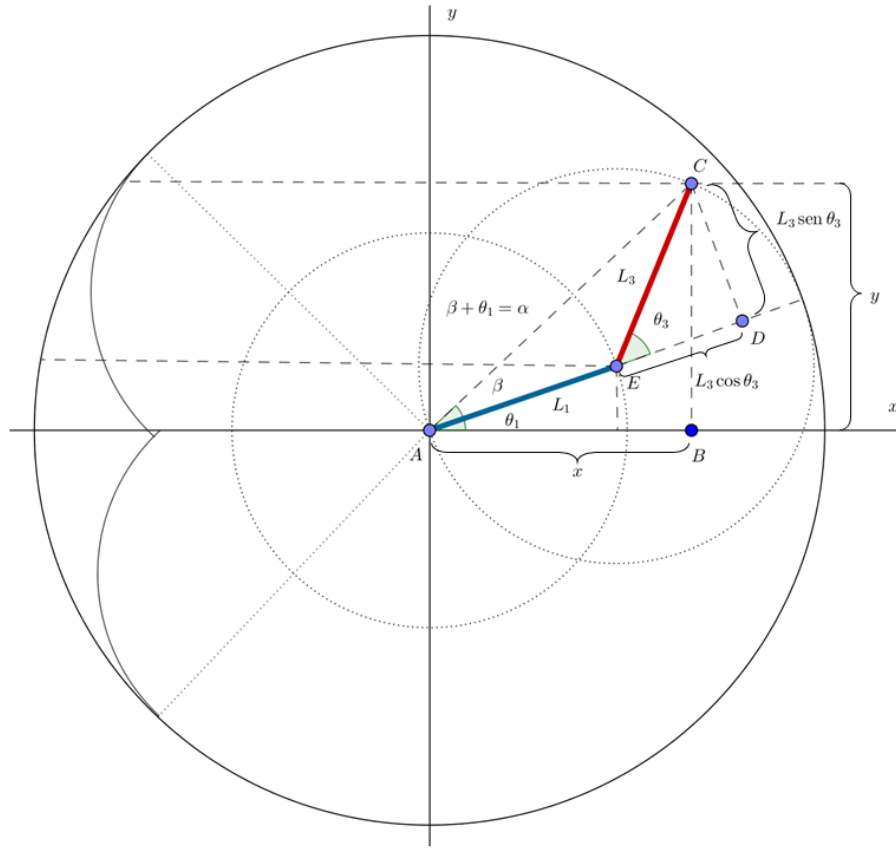
Equação (72).

$$\theta_3 = \arctan \left( \frac{\pm \sqrt{1 - \left( \frac{x^2 + y^2 - L_1^2 - L_3^2}{2L_1L_3} \right)^2}}{\frac{x^2 + y^2 - L_1^2 - L_3^2}{2L_1L_3}} \right) \quad (72)$$

Substituindo  $L_1$  e  $L_3$  por 110 e simplificando, obtém-se a Equação (73), exatamente igual à Equação (50).

$$\theta_3 = \arctan \left( \pm \frac{\sqrt{-(x^2 + y^2)(x^2 + y^2 - (220)^2)}}{x^2 + y^2 - 2(110)^2} \right) \quad (73)$$

Para determinar  $\theta_1$ , considera-se o segmento de reta  $\overline{CD}$ , perpendicular à reta que contém o comprimento do elo  $L_1$ , conforme observado na Figura 11.


 Figura 11: Determinação de  $\theta_1$  pelo método geométrico

Pode-se determinar o ângulo  $\beta$ , indicado na figura, como sendo:

$$\beta = \arctan\left(\frac{L_3 s_3}{L_1 + L_3 c_3}\right). \quad (74)$$

Considerando, de acordo com a Figura 11, a soma dos ângulos  $\beta$  e  $\theta_1$ , tem-se que:

$$\beta + \theta_1 = \arctan\left(\frac{y}{x}\right). \quad (75)$$

Dessa forma:

$$\theta_1 = (\beta + \theta_1) - \beta = \arctan\left(\frac{y}{x}\right) - \arctan\left(\frac{L_3 s_3}{L_1 + L_3 c_3}\right). \quad (76)$$

Como  $\theta_3$  pode assumir dois valores com base na Equação (71),  $\theta_1$  também assume dois valores, conforme pode-se observar nas Equações (77) e (78).

$$\theta_1 = \arctan\left(\frac{y}{x}\right) - \arctan\left(\frac{L_3 \left( \sqrt{1 - \left( \frac{x^2 + y^2 - L_1^2 - L_3^2}{2L_1L_3} \right)^2} \right)}{L_1 + L_3 \left( \frac{x^2 + y^2 - L_1^2 - L_3^2}{2L_1L_3} \right)}\right) \quad (77)$$

$$\theta_1 = \arctan\left(\frac{y}{x}\right) - \arctan\left(\frac{-L_3 \left( \sqrt{1 - \left( \frac{x^2 + y^2 - L_1^2 - L_3^2}{2L_1L_3} \right)^2} \right)}{L_1 + L_3 \left( \frac{x^2 + y^2 - L_1^2 - L_3^2}{2L_1L_3} \right)}\right) \quad (78)$$

Substituindo  $L_1$  e  $L_3$  por 110 e fazendo as devidas simplificações, determinam-se as Equações (79) e (80).

$$\theta_1 = \arctan\left(\frac{y}{x}\right) - \arctan\left(\frac{\left( \sqrt{1 - (x^2 + y^2)(x^2 + y^2 - (220^2))} \right)}{x^2 + y^2}\right) \quad (79)$$

$$\theta_1 = \arctan\left(\frac{y}{x}\right) - \arctan\left(\frac{-\left( \sqrt{1 - (x^2 + y^2)(x^2 + y^2 - (220^2))} \right)}{x^2 + y^2}\right) \quad (80)$$

Sabendo-se que

$$\tan(\alpha - \beta) = \frac{\tan \alpha + \tan \beta}{1 - \tan \alpha \cdot \tan \beta}, \quad (81)$$

e considerando  $\alpha = \theta_1 + \beta$ , de acordo com a Figura 11, tem-se que  $\theta_1 = \alpha - \beta$ . Dessa forma, pode-se determinar  $\tan \theta_1$  com base nas Equações (82) e (83).

$$\tan \theta_1 = \tan(\alpha - \beta) = \frac{\frac{y}{x} + \left( \frac{\sqrt{1 - (x^2 + y^2)(x^2 + y^2 - (220^2))}}{x^2 + y^2} \right)}{1 - \frac{y}{x} \left( \frac{\sqrt{1 - (x^2 + y^2)(x^2 + y^2 - (220^2))}}{x^2 + y^2} \right)} \quad (82)$$

$$\tan \theta_1 = \tan(\alpha - \beta) = \frac{\frac{y}{x} + \left( \frac{-\sqrt{1 - (x^2 + y^2)}(x^2 + y^2 - (220^2))}{x^2 + y^2} \right)}{1 - \frac{y}{x} \left( \frac{\sqrt{1 - (x^2 + y^2)}(x^2 + y^2 - (220^2))}{x^2 + y^2} \right)} \quad (83)$$

Com base nas Equações (82) e (83), determinam-se as Equações (84) e (85), que fornecem os valores de  $\theta_1$ , exatamente iguais às Equações (59) e (60).

$$\theta_1 = \arctan \left( \frac{y(x^2 + y^2) + x\sqrt{-(x^2 + y^2)}(x^2 + y^2 - 220^2)}{x(x^2 + y^2) - y\sqrt{-(x^2 + y^2)}(x^2 + y^2 - 220^2)} \right) \quad (84)$$

$$\theta_1 = \arctan \left( \frac{y(x^2 + y^2) - x\sqrt{-(x^2 + y^2)}(x^2 + y^2 - 220^2)}{x(x^2 + y^2) + y\sqrt{-(x^2 + y^2)}(x^2 + y^2 - 220^2)} \right) \quad (85)$$

Pela geometria do manipulador robótico, observa-se que a segunda junta desloca-se na direção do eixo  $z$ . Dessa forma, é possível determinar imediatamente o valor dessa variável de junta prismática como:  $d_2 = z - 347$ .

Devido às mesmas limitações observadas nas Equações (50), (59) e (60), deduzidas nesta seção, justifica-se novamente a utilização da função  $\text{Atan2}$ .

Assim, tem-se:

$$\theta_1 = \text{Atan2} \left( \left( \frac{y(x^2 + y^2) \pm x\sqrt{-(x^2 + y^2)}(x^2 + y^2 - 220^2)}{x(x^2 + y^2) \mp y\sqrt{-(x^2 + y^2)}(x^2 + y^2 - 220^2)} \right) \right), \quad (86)$$

$$d_2 = z - 347, \quad (87)$$

$$\theta_3 = \text{Atan2} \left( \pm\sqrt{-(x^2 + y^2)}(x^2 + y^2 - (220^2)), (x^2 + y^2 - 2(110)^2) \right). \quad (88)$$

O método geométrico, utilizado nesta seção, obviamente tem sua simplicidade reduzida à medida que a geometria do robô manipulador se torna mais complexa.

### 4.3 CONFIGURAÇÕES SINGULARES

Para esse manipulador robótico, existem duas configurações de juntas que requerem cálculos adicionais para resolver o problema cinemático inverso. Essas configurações particulares, denominadas configurações singulares, ocorrem quando os eixos das juntas se intersectam e quando são coincidentes.

A partir deste ponto, o conceito de singularidade será introduzido pelo Jacobiano, um método suplementar à matriz de Denavit-Hartenberg, que tem como função a determinação de configurações singulares.

As equações da cinemática direta definem uma função entre o espaço cartesiano (efetuador) e o espaço das posições das juntas. As relações entre velocidades são determinadas pelo Jacobiano dessa função. De acordo com Craig (2017), Jacobiano é uma forma multidimensional de derivada usualmente utilizado no planejamento e execução de trajetórias, na determinação de configurações singulares, na dedução das equações dinâmicas do movimento e na transformação de forças e torques do órgão terminal para as juntas do manipulador robótico.

O Jacobiano de um manipulador robótico é uma matriz de derivadas, e esta matriz descreve as mudanças diferenciais na localização do efetuador final causadas pelas mudanças diferenciais nas variáveis das juntas do robô. Utilizando coordenadas cartesianas, o deslocamento do efetuador final pode ser descrito pelo vetor de diferenciais  $\bar{D}$ .

Considerando  $\bar{D}_q$  como sendo um vetor de movimento diferencial especial que descreve o deslocamento do efetuador final em termos das coordenadas da junta, tem-se que:  $\bar{D}_q = [dq_1 \quad dq_2 \quad \cdots \quad dq_n]^T$ , para um manipulador com  $n$  juntas, no qual  $q$  representa a variável de junta rotativa  $\theta$  ou prismática  $d$ .

Considerando um manipulador robótico,  $\bar{D}$  e  $\bar{D}_q$  são relacionados pela equação (89):

$$\bar{D} = J\bar{D}_q. \quad (89)$$

O Jacobiano inverso estabelece a relação entre as velocidades cartesianas do efetuador final  $\bar{D}$  e as velocidades das juntas  $\bar{D}_q$ , e é dado por:

$$\bar{D}_q = J^{-1}\bar{D}. \quad (90)$$

Segundo Kendricks (2007), as propriedades do Jacobiano podem ser descritas:

- i. O número de linhas é determinado pelo número de graus de liberdade (a junta que representa o efetuador final não é contabilizada);
- ii. O número de colunas é determinado pelo número de juntas do manipulador;
- iii. Em alguns casos, o Jacobiano não é uma matriz quadrada.

Inicialmente, para determinação do Jacobiano do manipulador SCARA TS20, utiliza-se a solução cinemática direta, determinada na Seção 4.1, para descrever a posição  $(x, y, z)$  do efetuador final com base na Equação (91).

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 110c_{13} + 110c_1 \\ 110s_{13} + 110s_1 \\ 347 + d_2 \end{bmatrix} \quad (91)$$

Pode-se observar que  $z$ , que representa a altura da posição do efetuador final, depende somente de  $d_2$ ,  $x$  e  $y$ , os quais são determinados apenas por  $\theta_1$  e  $\theta_3$ . Cada configuração de junta do robô foi projetada no plano  $xy$ , ao ser examinado a partir de uma vista superior. Com essa perspectiva, o foco é restringido para  $\theta_1$  e  $\theta_3$ .

Dessa forma, com base na Equação (91), é possível utilizar a seguinte representação:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 110c_{13} + 110c_1 \\ 110s_{13} + 110s_1 \end{bmatrix}. \quad (92)$$

Para calcular o Jacobiano, a matriz é diferenciada, de tal modo que:



$$\begin{bmatrix} dx \\ dy \end{bmatrix} = J \begin{bmatrix} d\theta_1 \\ d\theta_3 \end{bmatrix}. \quad (93)$$

A obtenção das expressões diferenciais, utiliza-se o conceito de diferencial total, de forma que:

$$df = \sum_{i=1}^n \frac{\partial f}{\partial x_i} dx_i; \quad f = f(x_1, x_2, \dots, x_n). \quad (94)$$

Desse modo, tem-se:

$$\begin{cases} dx = \frac{\partial(110c_{13} + 110c_1)}{\partial\theta_1} d\theta_1 + \frac{\partial(110c_{13} + 110c_1)}{\partial\theta_3} d\theta_3 \\ dy = \frac{\partial(110s_{13} + 110s_1)}{\partial\theta_1} d\theta_1 + \frac{\partial(110s_{13} + 110s_1)}{\partial\theta_3} d\theta_3 \end{cases}. \quad (95)$$

Calculando as derivadas parciais, utilizando ainda as identidades que calculam as funções trigonométricas da soma de arcos, obtém-se:

$$\begin{cases} dx = (-110c_1s_3 - 110s_1c_3 - 110s_1) d\theta_1 + (-110c_1s_3 - 110s_1c_3) d\theta_3 \\ dy = (110c_1c_3 - 110s_1s_3 + 110c_1) d\theta_1 + (-110s_1s_3 + 110c_1c_3) d\theta_3 \end{cases}. \quad (96)$$

Assim, é possível identificar a matriz  $J$  na Equação (97) ao compará-la com a (93):

$$\begin{bmatrix} dx \\ dy \end{bmatrix} = \begin{bmatrix} -110c_1s_3 - 110s_1c_3 - 110s_1 & -110c_1s_3 - 110s_1c_3 \\ 110c_1c_3 - 110s_1s_3 + 110c_1 & -110s_1s_3 + 110c_1c_3 \end{bmatrix} \begin{bmatrix} d\theta_1 \\ d\theta_3 \end{bmatrix}. \quad (97)$$

Em seguida, o Jacobiano inverso é calculado:

$$J^{-1} = \frac{\begin{bmatrix} J_{22} & -J_{12} \\ -J_{21} & J_{11} \end{bmatrix}}{|J|}. \quad (98)$$

Em que:

$$J_{11} = -110c_1s_3 - 110s_1c_3 - 110s_1, \quad (99)$$

$$J_{12} = -110c_1s_3 - 110s_1c_3, \quad (100)$$

$$J_{21} = 110c_1c_3 - 110s_1s_3 + 110c_1, \quad (101)$$

$$J_{22} = -110s_1s_3 + 110c_1c_3. \quad (102)$$

Desse modo, determina-se o determinante do Jacobiano:

$$|J| = J_{11}J_{22} - J_{12}J_{21}, \quad (103)$$

$$|J| = (-110c_1s_3 - 110s_1c_3 - 110s_1)(-110s_1s_3 + 110c_1c_3) - (-110c_1s_3 - 110s_1c_3)(110c_1c_3 - 110s_1s_3 + 110c_1). \quad (104)$$

Após alguns cancelamentos, considerando  $(s_1^2 + c_1^2 = 1)$ , tem-se:

$$|J| = 110^2 s_1^2 s_3 + 110^2 c_1^2 s_3 = 110^2 s_3. \quad (105)$$

Assim, o Jacobiano inverso é definido quando  $s_3 \neq 0$ . Pode-se afirmar então que:

$$\begin{bmatrix} d\theta_1 \\ d\theta_3 \end{bmatrix} = \frac{1}{110^2 s_3} \begin{bmatrix} J_{22} & -J_{12} \\ -J_{21} & J_{11} \end{bmatrix} \begin{bmatrix} dx \\ dy \end{bmatrix}. \quad (106)$$

A partir deste ponto, o significado de Singularidade será explicado com maiores detalhes.

Existem dois tipos de singularidades: a singularidade interna no volume de trabalho e a singularidade no limite de volume de trabalho. Uma singularidade interna no volume de trabalho ocorre dentro do espaço de trabalho. A singularidade no limite de volume de trabalho ocorre quando o manipulador está totalmente estendido para o limite externo ou totalmente retraído para o limite interno do seu volume de trabalho, esta configuração particular será apresentada na Seção 4.4. A Figura 12 ilustra com detalhes o volume de trabalho do robô SCARA Stäubli TS20.

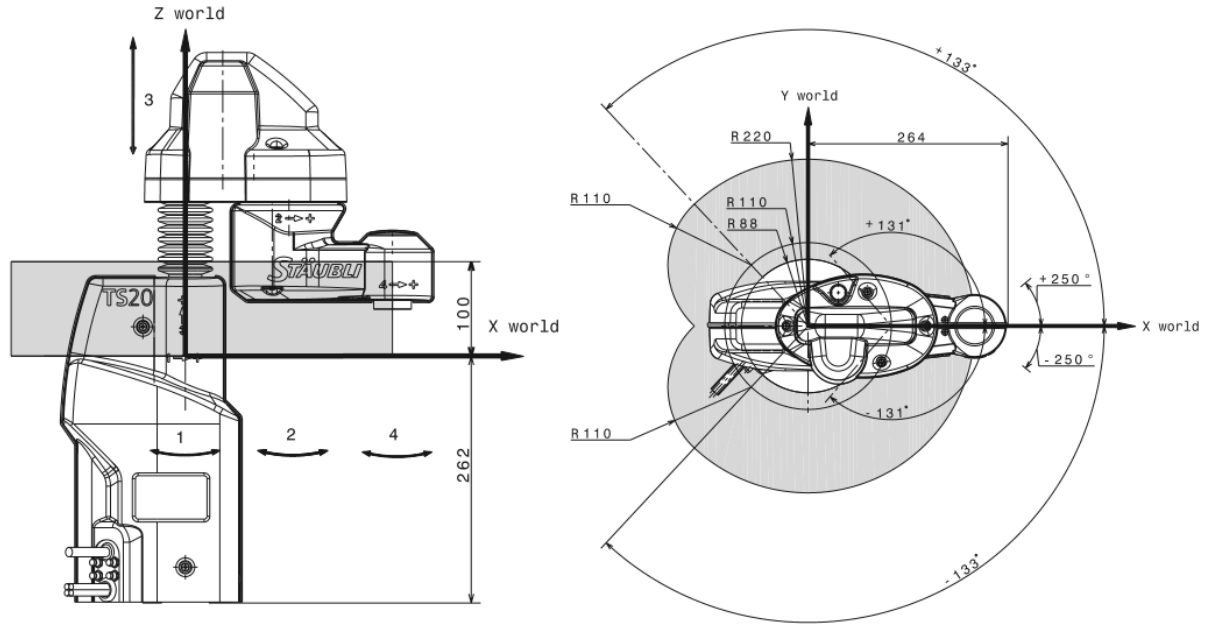


Figura 12: Volume de trabalho do robô SCARA Stäubli TS20

Fonte: <http://www.staubli.com/in/robotics/>

Segundo Kendricks (2007), em alguns pontos do volume de trabalho, um manipulador robótico perde um ou mais graus de liberdade, reduzindo a sua mobilidade em algumas direções. O ponto no espaço em que isso ocorre é chamado de singularidade interna no volume de trabalho ou singularidade no limite do volume de trabalho. O Jacobiano é utilizado para a determinação dessas configurações singulares, uma vez que no ponto de singularidade, o Jacobiano não é invertível, isto é, seu determinante é zero.

A partir deste ponto, as configurações singulares para o robô SCARA Stäubli TS20 serão determinadas.

Pela Equação (105) sabe-se que o Jacobiano não é invertível quando  $\theta_3$  é igual a zero ou  $180^\circ$ , ou seja, quando  $s_3 = 0$ . Mas  $\theta_3$  varia segundo o intervalo  $-131^\circ < \theta_3 < 131^\circ$ . Dessa forma,  $180^\circ$  não é um ângulo que fornece um ponto no interior do volume de trabalho do robô. Contudo, quando  $\theta_3 = 0$ , obtém-se uma singularidade no limite do volume de trabalho. Substituindo  $\theta_3 = 0$  em  $J$ , obtém-se:

$$\begin{bmatrix} dx \\ dy \end{bmatrix} = \begin{bmatrix} -220s_1 & -110s_1 \\ 220c_1 & 110c_1 \end{bmatrix} \begin{bmatrix} d\theta_1 \\ d\theta_3 \end{bmatrix}. \tag{107}$$

Todos os pontos pertencentes à circunferência de raio 220 com centro na origem, em que  $\theta_1$  pode variar dentro dos valores impostos pelo intervalo  $-133^\circ < \theta_1 < 133^\circ$ , são singularidades no limite do volume de trabalho.

Além disso, o Jacobiano inverso é indefinido exatamente nesses pontos. Dessa forma, existe uma única solução (isto é, uma única configuração de juntas) para que o efetuador do robô possa atingir esses pontos.

#### 4.4 EXEMPLOS GENÉRICOS

Nesta seção, as equações deduzidas são verificadas nos casos em que os elos do manipulador são colineares, quando pretende-se atingir um ponto no interior do volume de trabalho do manipular e quando um ponto a ser atingido se encontra fora do volume de trabalho.

Considerando  $L_1$  e  $L_3$  colineares, os elos  $L_1 = 110mm$  e  $L_3 = 110mm$  se encontram alinhados, como pode-se observar na Figura 13. Sendo assim:  $\theta_3 = 0^\circ$ .

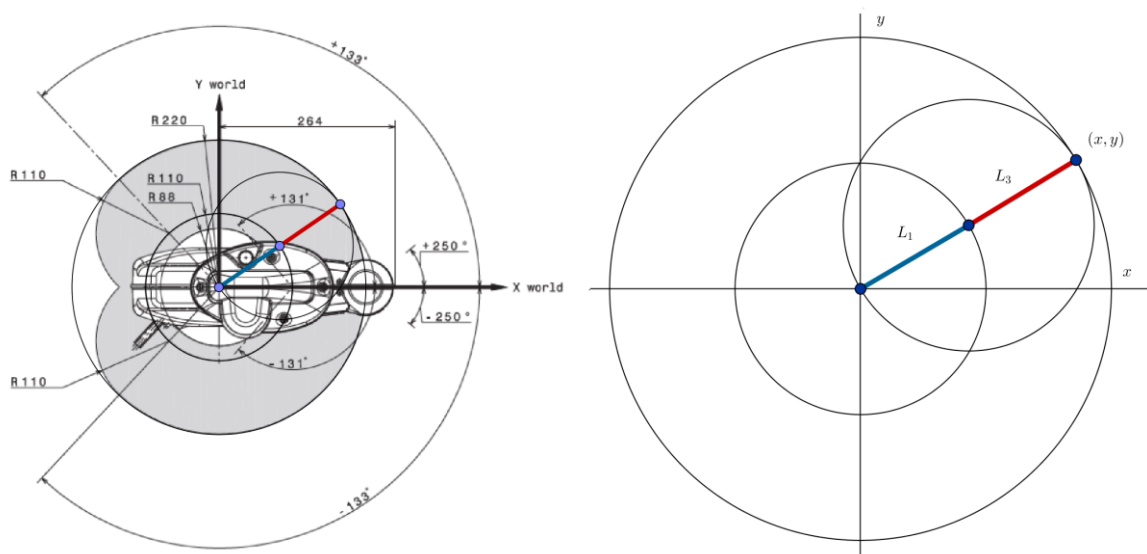


Figura 13: Robô SCARA Stäubli TS20 com elos  $L_1$  e  $L_3$  colineares

Utilizando a Equação (29), obtém-se:

$$x = 110 \cos(\theta_1 + 0) + 110 \cos \theta_1 = 220 \cos \theta_1, \quad (108)$$

$$y = 110 \sin(\theta_1 + 0) + 110 \sin \theta_1 = 220 \sin \theta_1. \quad (109)$$

Neste caso, existe uma única disposição para atingir o ponto  $(x; y)$ .

Porém, esta é uma configuração que deve ser evitada, uma vez que é considerada singular. Informações mais detalhadas sobre essas configurações foram apresentadas na Seção 4.3.

Considerando um ponto no interior do volume de trabalho, os elos  $L_1 = 110mm$  e  $L_3 = 110mm$  estão posicionados de forma que o efetuador esteja em um ponto no interior do volume de trabalho do robô, como se observa na Figura 14.

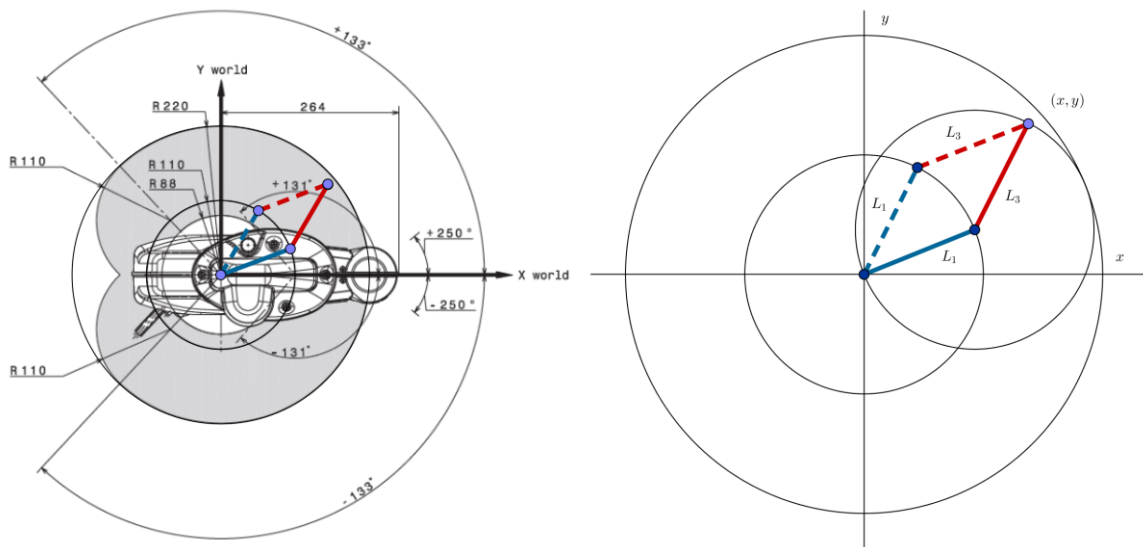


Figura 14: Ponto no interior do volume de trabalho do robô SCARA Stäubli TS20

É possível determinar  $(x; y)$ , em que  $\theta_1 = 30^\circ$  e  $\theta_3 = 30^\circ$ , usando cinemática direta.

$$x = 110 \cos(30^\circ + 30^\circ) + 110 \cos(30^\circ) = 55 + 55\sqrt{3} \quad (110)$$

$$y = 110 \sin(30^\circ + 30^\circ) + 110 \sin(30^\circ) = 55\sqrt{3} + 55 \quad (111)$$

Usando o método da cinemática inversa, esses resultados obtidos são verificados. Dada a posição final descrita por  $x$  e  $y$ , valores obtidos pelas Equações (110) e (111), pretende-se determinar  $\theta_1$  e  $\theta_3$ . Esses valores são substituídos nas Equações (44) e (48) deduzidas na Seção 4.2.

$$c_3 = \frac{(55 + 55\sqrt{3})^2 + (55\sqrt{3} + 55)^2}{24200} - 1 = \frac{\sqrt{3}}{2} \quad (112)$$

$$s_3 = \pm \frac{\sqrt{(55 + 55\sqrt{3})^2 + (55\sqrt{3} + 55)^2 - \left(110 \frac{\sqrt{3}}{2} + 110\right)^2}}{110} = \pm \frac{1}{2} \quad (113)$$

Como  $s_3$  pode assumir dois valores, obviamente  $\theta_3$  também assume dois valores, conforme as Equações (114) e (115). Dessa forma, considerando  $\tan \theta_3 = \left(\frac{s_3}{c_3}\right)$ , tem-se:

$$\theta_3 = \arctan \left( \frac{\left(\frac{1}{2}\right)}{\left(\frac{\sqrt{3}}{2}\right)} \right) = \arctan \left( \frac{\sqrt{3}}{3} \right) = 30^\circ, -\frac{\pi}{2} < \theta_1 < \frac{\pi}{2}, \quad (114)$$

$$\theta_3 = \arctan \left( \frac{\left(-\frac{1}{2}\right)}{\left(\frac{\sqrt{3}}{2}\right)} \right) = \arctan \left( -\frac{\sqrt{3}}{3} \right) = -30^\circ, -\frac{\pi}{2} < \theta_1 < \frac{\pi}{2}. \quad (115)$$

Para encontrar  $\theta_1$ , substitui-se  $s_3 = \frac{1}{2}$  e  $c_3 = \frac{\sqrt{3}}{2}$  nas Equações (53) e (54) para obter  $s_1$  e  $c_1$ .

$$s_1 = \frac{(55\sqrt{3} + 55) \left(110 \left(\frac{\sqrt{3}}{2}\right) + 110\right) - (55 + 55\sqrt{3}) 110 \left(\frac{1}{2}\right)}{(55 + 55\sqrt{3})^2 + (55\sqrt{3} + 55)^2} = \frac{1}{2} \quad (116)$$

$$c_1 = \frac{(55 + 55\sqrt{3}) \left(110 \left(\frac{\sqrt{3}}{2}\right) + 110\right) + (55\sqrt{3} + 55) 110 \left(\frac{1}{2}\right)}{(55 + 55\sqrt{3})^2 + (55\sqrt{3} + 55)^2} = \frac{\sqrt{3}}{2} \quad (117)$$

Desse modo:

$$\theta_1 = \arctan \left( \frac{\left( \frac{1}{2} \right)}{\left( \frac{\sqrt{3}}{2} \right)} \right) = 30^\circ. \quad (118)$$

Além disso, faz-se necessário determinar  $\theta_1$ , quando  $s_3 = -\frac{1}{2}$ .

$$s_1 = \frac{(55\sqrt{3} + 55) \left( 110 \left( \frac{\sqrt{3}}{2} \right) + 110 \right) - (55 + 55\sqrt{3}) 110 \left( -\frac{1}{2} \right)}{(55 + 55\sqrt{3})^2 + (55\sqrt{3} + 55)^2} = \frac{\sqrt{3}}{2} \quad (119)$$

$$c_1 = \frac{(55 + 55\sqrt{3}) \left( 110 \left( \frac{\sqrt{3}}{2} \right) + 110 \right) + (55\sqrt{3} + 55) 110 \left( -\frac{1}{2} \right)}{(55 + 55\sqrt{3})^2 + (55\sqrt{3} + 55)^2} = \frac{1}{2} \quad (120)$$

Assim:

$$\theta_1 = \arctan \left( \frac{\left( \frac{\sqrt{3}}{2} \right)}{\left( \frac{1}{2} \right)} \right) = \arctan(\sqrt{3}) = 60^\circ. \quad (121)$$

Com base na resolução de  $s_3$ , encontra-se duas soluções para  $\theta_3$  e duas soluções correspondentes para  $\theta_1$ . Nesse sentido, existem duas configurações de juntas que posicionam o efetuador final do robô em  $(x; y) = (55 + 55\sqrt{3}; 55\sqrt{3} + 55)$ . São elas:  $(\theta_3 = 30^\circ, \theta_1 = 30^\circ)$  e  $(\theta_3 = -30^\circ, \theta_1 = 60^\circ)$ .

Assim, todas as configurações de juntas que posicionam o efetuador final do robô em  $(x; y) = (55 + 55\sqrt{3}; 55\sqrt{3} + 55)$  são determinadas.

Supondo um ponto fora do volume de trabalho, como se pode observar na Figura 15, o ponto  $(x; y)$  se encontra fora dos limites impostos pelo volume de trabalho do manipulador.

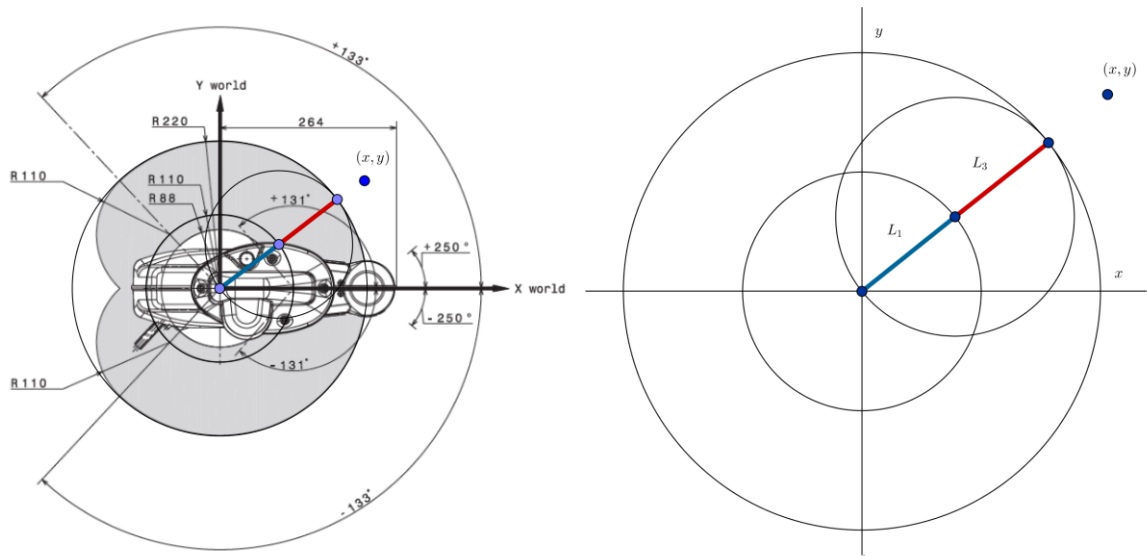


Figura 15: Ponto fora do volume de trabalho do robô SCARA Stäubli TS20

Considerando arbitrariamente o ponto  $(x; y) = (240; 240)$ , determinam-se  $\theta_1$  e  $\theta_3$ .

Utilizando a Equação (44), tem-se para  $c_3$ :

$$c_3 = \frac{240^2 + 240^2}{2(110)^2} - 1 \approx 3,7603. \quad (122)$$

Mas esta é uma indeterminação, uma vez que  $c_3 \notin [-1,1]$ . Como uma solução para  $\theta_3$  não poderá ser encontrada, é impossível também determinar uma solução para  $\theta_1$ . Dessa forma, este exemplo confirma o fato de que apenas os valores que se encontram no domínio especificado para cada variável comum têm soluções possíveis.



4.4.1 Análise dos resultados obtidos

Encontram-se  $d_2$ ,  $\theta_1$  e  $\theta_3$ , nesta ordem. Em particular,  $\theta_1$  e  $\theta_3$  são determinados em termos de  $(x; y)$ . No entanto, pelas equações utilizadas, não é possível determinar as variáveis de junta  $\theta_1$  e  $\theta_3$  que posicionem o efetuador do manipulador em determinados pontos pertencentes ao seu volume de trabalho, por conta das limitações observadas na Seção 4.2. Limitações associadas aos ângulos  $\frac{\pi}{2}$  e  $-\frac{\pi}{2}$ , para os quais a tangente não está definida. Dessa forma, justifica-se a utilização da função  $\text{Atan2}$ .

Deseja-se determinar todos os pontos  $(x; y)$ , alcançáveis dentro do volume de trabalho do robô. Ao escolher três pontos distintos no espaço,  $\theta_1$  e  $\theta_3$  são determinados, satisfazendo às condições  $-\frac{\pi}{2} < \theta_1 < \frac{\pi}{2}$  e  $-\frac{\pi}{2} < \theta_3 < \frac{\pi}{2}$ , havendo nenhuma, uma ou duas configurações que permitem o posicionamento do efetuador do robô em um determinado ponto do espaço, como se pode observar na Figura 16. Determina-se o conjunto de pontos que não produzem soluções, o conjunto que produzirá uma única solução e o conjunto de pontos que produzirão duas soluções.

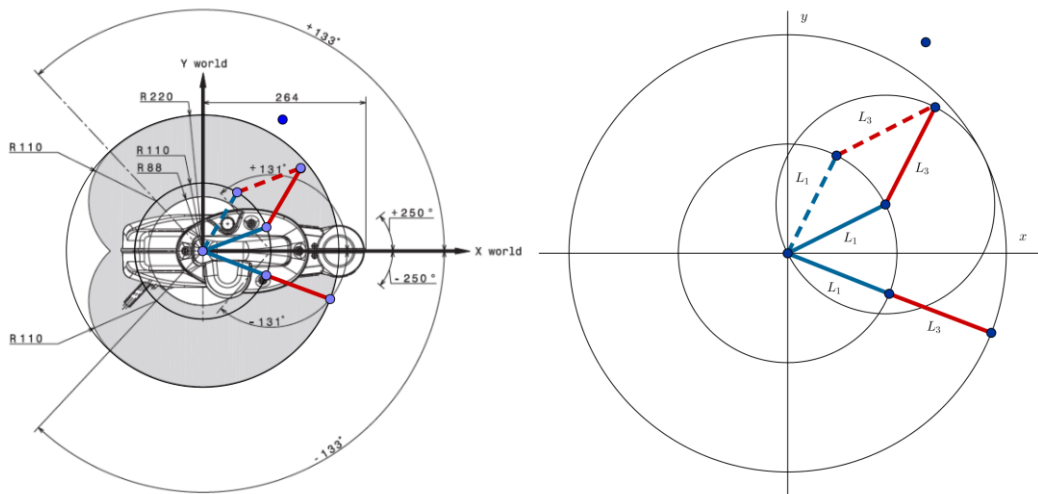


Figura 16: Resumo das soluções para o robô SCARA Stäubli TS20

Manipulando as equações no modelo cinemático para encontrar todas as soluções restantes, utilizando a Equação (44) e isolando  $\theta_3$ , obtém-se:

$$\theta_3 = \arccos\left(\frac{x^2 + y^2}{2(110)^2} - 1\right). \quad (123)$$

Kendricks (2007) desenvolveu uma análise semelhante para o manipulador Fanuc A-510, um robô do tipo SCARA com três graus de liberdade.

De acordo com Wenz e Worn (2007), são necessárias 81 operações de ponto flutuante (flop)<sup>2</sup> para a solução do problema cinemático direto de um manipulador do tipo SCARA, e até 262 operações de ponto flutuante para obtenção das duas soluções para a cinemática inversa desse mesmo manipulador.

A análise do custo computacional é feita de duas formas: em termos de tempo de execução e de número de operações de ponto flutuante realizadas. O tempo de execução é proporcional ao número de operações de ponto flutuante realizadas pelo sistema, porém o tempo total não depende apenas deste número de operações realizadas, mas também de outros fatores, como taxas de transferência de dados, memória, quantidade de processamentos por segundo que a CPU é capaz de realizar, dentre outros; isto é, depende do *hardware* utilizado.

Como a maioria dos algoritmos numéricos envolvem mais operações de ponto flutuantes do que outras operações computacionais, como por exemplo: armazenamento, testes lógicos, entre outras. Pode-se mensurar razoavelmente o custo computacional de um algoritmo numérico contando o número de operações aritméticas de ponto flutuante que esse algoritmo executa.

---

<sup>2</sup> De acordo com Ueberhuber (1997), a contagem do número de operações ponto flutuante para execução de uma determinada tarefa é o procedimento mais justo para verificação da eficiência computacional de um determinado algoritmo. Utiliza-se o termo (flop), que é um acrônimo para “operação de ponto flutuante” (em inglês *Floating Point Operation*). Usualmente o termo (flops) é utilizado como uma medida de velocidade e significa “operações de ponto flutuante por segundo”, o "s" no final do termo não denota plural, mas significa exatamente o que o acrônimo apresenta, segundo.

Dessa forma, tem-se que o número total de operações aritméticas de ponto flutuante necessárias para resolver um problema indica o custo computacional da solução. Essa informação representa um importante parâmetro para a escolha de um entre vários algoritmos para execução de uma mesma tarefa.

Os custos de execução das operações mais comuns são dados na Tabela 3.

Tabela 3: Custo computacional das operações mais comuns

Operação	Custo Computacional
Adição, Subtração e Multiplicação	1 flop
Divisão e Raiz Quadrada	4 flop
Função exponencial e Funções trigonométricas	8 flop

Fonte: (ADDISON, *et al.*, 1993, p. 8, adaptado)

Por meio da análise da geometria do robô e manipulando o modelo cinemático, mais soluções são determinadas em novas configurações geométricas. Porém só foi possível determinar essas novas soluções devido à simplicidade geométrica do robô SCARA Stäubli TS20. No entanto, se a geometria do robô é mais complexa, como a de um robô antropomórfico por exemplo, as soluções não são facilmente encontradas e, em alguns casos, não podem ser obtidas.

A cinemática inversa do manipulador SCARA Stäubli TS20 será solucionada com base na Teoria de Bases de Gröbner, no Capítulo 5.

## 5 CINEMÁTICA INVERSA DO STÄUBLI TS20 POR MEIO DA TEORIA DE BASES DE GRÖBNER

O problema da cinemática inversa em Robótica é simplesmente determinar todas as combinações de configurações de juntas que posicionarão o efetuador final do robô em um determinado ponto no espaço. Para resolver esse problema pela Teoria de Bases de Gröbner, deve-se determinar equações polinomiais que modelem o movimento do braço do robô em cada configuração de junta, de modo que o efetuador do robô possa ser posicionado em um determinado ponto no espaço.

Com todos esses elementos, deve-se encontrar um conjunto de soluções com base nos possíveis movimentos de cada junta.

### 5.1 MODELO ALGÉBRICO PARA O ROBÔ SCARA STÄUBLI TS20

Cox, Little e O'Shea (1992) desenvolveram um modelo algébrico para a solução do problema cinemático inverso de um manipulador planar genérico com dois graus de liberdade. A partir deste ponto, um modelo algébrico para o SCARA Stäubli TS20 será construído seguindo o mesmo procedimento apresentado pelos autores supramencionados.

Tomando uma visão superior do robô, pode-se projetar os movimentos de cada variável de junta no plano  $xy$ , utilizando o ponto  $(x; y)$  para representar o efetuador final no espaço. Dessa forma, a modelagem do problema pode ser realizada com equações polinomiais simples.

Para resolver o problema da cinemática inversa, estas equações polinomiais devem necessariamente descrever o comportamento dos ângulos de junta  $\theta_1$  e  $\theta_3$  em termos do ponto  $(x; y)$ , conforme pode-se observar na Figura 17.

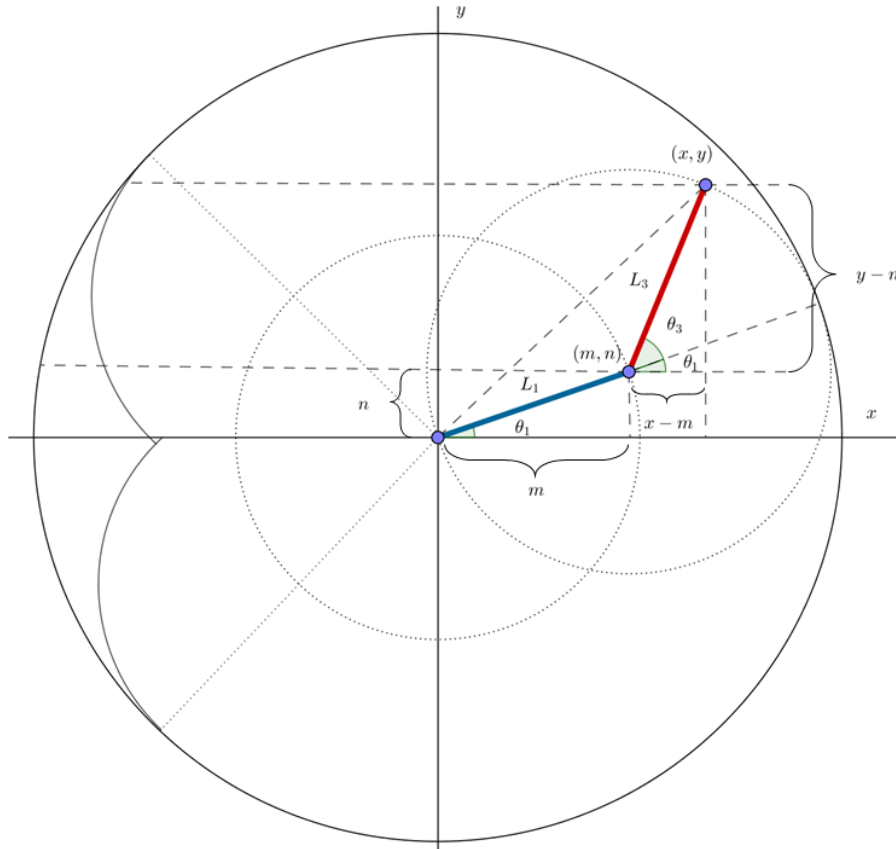


Figura 17: Modelo algébrico do SCARA Stäubli TS20 (vista superior)

Fazendo a análise geométrica do modelo e considerando  $L_1 = L_3 = 110$ , são deduzidas as Equações (124) e (125).

$$(x - m, y - n) = (110c_{13}, 110s_{13}) \quad (124)$$

$$(m, n) = (110c_1, 110s_1) \quad (125)$$

Podendo-se determinar as equações:

$$m = 110c_1, \quad (126)$$

$$n = 110s_1, \quad (127)$$

$$x - m = 110c_{13}, \quad (128)$$

$$y - n = 110s_{13}. \quad (129)$$

Fazendo as devidas substituições, obtém-se:

$$x = 110c_{13} + 110c_1, \quad (130)$$

$$y = 110s_{13} + 110s_1. \quad (131)$$

Usando as identidades de adição de funções trigonométricas, determinam-se:

$$x = 110(c_1c_3 - s_1s_3) + 110c_1, \quad (132)$$

$$y = 110(s_1c_3 + s_3c_1) + 110s_1. \quad (133)$$

Utilizando as identidades trigonométricas  $(c_1^2 + s_1^2 = 1)$  e  $(c_3^2 + s_3^2 = 1)$ , um sistema de quatro equações com quatro incógnitas, é obtido.

$$110(c_1c_3 - s_1s_3) + 110c_1 - x = 0 \quad (134)$$

$$110(s_1c_3 + s_3c_1) + 110s_1 - y = 0 \quad (135)$$

$$c_1^2 + s_1^2 - 1 = 0 \quad (136)$$

$$c_3^2 + s_3^2 - 1 = 0 \quad (137)$$

Pela geométrica do manipulador, observa-se que a segunda junta desloca-se na direção do eixo  $z$ . Desse modo, é possível determinar imediatamente o valor desta variável como:  $d_2 = z - 347$ , sendo desnecessário a inclusão desta equação no modelo geométrico do robô.

O *software* MAPLE<sup>3</sup> será utilizado para encontrar uma Base de Gröbner para este sistema de equações.

---

<sup>3</sup> Além do MAPLE, muitos outros sistemas de computação algébrica incluem pacotes para lidar com Bases de Gröbner. Entre eles: Singular, muMath, AXIOM, SageMath, Mathematica, Magma, Macaulay, CoCoA e Reduce.

## 5.2 DETERMINAÇÃO DE UMA BASE DE GRÖBNER

Tomando  $c_1 = \cos \theta_1$ ,  $s_1 = \sin \theta_1$ ,  $s_3 = \sin \theta_3$  e  $c_3 = \cos \theta_3$ , usando a ordem lexicográfica  $c_3 \succ s_3 \succ c_1 \succ s_1$ , e as equações:  $f_1 = L_3(c_1c_3 - s_1s_3) + L_1c_1 - x$ ,  $f_2 = L_3(s_1c_3 + s_3c_1) + L_1s_1 - y$ ,  $f_3 = c_1^2 + s_1^2 - 1$ ,  $f_4 = c_3^2 + s_3^2 - 1$  e considerando que  $c_3$ ,  $s_3$ ,  $c_1$ ,  $s_1$  atuam como variáveis e  $L_1$ ,  $L_3$ ,  $x$  e  $y$  como coeficientes; utiliza-se o seguinte código no MAPLE:

```
> with(Groebner);
> with(codegen, cost);
> r1:=L3*(c1*c3-s1*s3)+L1*c1-x:
> r2:=L3*(c1*s3+c3*s1)+L1*s1-y:
> r3:=c1^2+s1^2-1:
> r4:=c3^2+s3^2-1:
> GB:=Groebner[Basis]([r1,r2,r3,r4],plex(c3,s3,c1,s1));
> math_operations:=cost(GB);
> V:={additions = 1, divisions = 4, functions = 8, multiplications = 1}:
> total_cost:=eval(cost(GB), V)flop;
```

No MAPLE, o comando “cost(GB)” é utilizado para calcular o número de operações matemáticas realizadas para a obtenção da base de Gröbner; e o comando “eval(cost(GB), V)flop”, para contabilizar o número de operações de ponto flutuante. A contagem de operações é expressa como um polinômio com os nomes: adições, multiplicações, divisões, funções, atribuições, dentre outras operações.

Após execução do código, as seguintes saídas são apresentadas:

```
> GB := [(4*L1^2*x^2+4*L1^2*y^2)*s1^2+(-4*L1^3*y+4*L1*L3^2*y-4*L1*x^2*y-
4*L1*y^3)*s1+L1^4-2*L1^2*L3^2-2*L1^2*x^2+2*L1^2*y^2+L3^4-2*L3^2*x^2-
2*L3^2*y^2+x^4+2*x^2*y^2+y^4, 2*L1*c1*x+2*L1*s1*y-L1^2+L3^2-x^2-y^2,
(2*L1*x^2+2*L1*y^2)*s1+2*L3*s3*L1*x-L1^2*y+L3^2*y-x^2*y-y^3,
2*L1*L3*c3+L1^2+L3^2-x^2-y^2]
> math_operations:=30*additions+99*multiplications
> total_cost:=129*flop
```

Com a realização de 129 operações de ponto flutuante, a seguinte base de Gröbner contendo 4 polinômios é obtida:

$$g_1 = (4L_1^2x^2 + 4L_1^2y^2)s_1^2 + (-4L_1^3y + 4L_1L_3^2y - 4L_1x^2y - 4L_1y^3)s_1 + L_1^4 - 2L_1^2L_3^2 - 2L_1^2x^2 + 2L_1^2y^2 + L_3^4 - 2L_3^2x^2 - 2L_3^2y^2 + x^4 + 2x^2y^2 + y^4, \quad (138)$$

$$g_2 = 2L_1c_1x + 2L_1s_1y - L_1^2 + L_3^2 - x^2 - y^2, \quad (139)$$

$$g_3 = (2L_1x^2 + 2L_1y^2)s_1 + 2L_3s_3L_1x - L_1^2y + L_3^2y - x^2y - y^3, \quad (140)$$

$$g_4 = 2c_3L_1L_3 + L_1^2 + L_3^2 - x^2 - y^2. \quad (141)$$

Após a substituição de  $L_1$  e  $L_3$  pelos valores dos comprimentos dos elos do robô, ou seja,  $L_1 = L_3 = 110$ , a seguinte base é determinada:

$$g_1 = (48400x^2 + 48400y^2)s_1^2 + (-440x^2y - 440y^3)s_1 + x^4 + 2x^2y^2 + y^4 - 48400x^2, \quad (142)$$

$$g_2 = 220c_1x + 220s_1y - x^2 - y^2, \quad (143)$$

$$g_3 = (220x^2 + 220y^2)s_1 + 24200xs_3 - x^2y - y^3, \quad (144)$$

$$g_4 = -x^2 - y^2 + 24200c_3 + 24200. \quad (145)$$

Ao analisar o primeiro elemento na base, pode-se concluir que este elemento é um polinômio de segundo grau em termos de  $s_1$ . O discriminante da Equação (142) é dado por (146).

$$(-440x^2y - 440y^3)^2 - 4(48400x^2 + 48400y^2)(x^4 + 2x^2y^2 + y^4 - 48400x^2) \quad (146)$$

Simplificando (146), obtém-se:

$$-193600x^2(x^2 + y^2)(x^2 + y^2 - 48400). \quad (147)$$

Ao examinar o discriminante da equação obtida, haverá duas soluções reais para  $s_1$  quando ele for maior que zero, e para cada um dos valores obtidos, haverá um valor correspondente para  $c_1$ , determinado por substituição retroativa. Como  $s_3$  depende de  $s_1$ , haverá duas soluções reais para  $\theta_3$  quando  $\theta_1$  assumir dois valores distintos.

Dessa forma, conclui-se que existem dois conjuntos distintos quando:



$$-193600x^2(x^2 + y^2)(x^2 + y^2 - 48400) > 0. \quad (148)$$

Como  $193600x^2(x^2 + y^2)$  é sempre positivo, basta verificar em que situação  $-(x^2 + y^2 - 48400) > 0$ . Desse modo, tem-se:

$$\begin{aligned} -(x^2 + y^2 - 48400) &> 0, \\ -x^2 - y^2 + 48400 &> 0, \\ -x^2 - y^2 &> -48400, \\ x^2 + y^2 &< 48400, \\ x^2 + y^2 &< 220^2. \end{aligned} \quad (149)$$

Portanto, o discriminante é positivo exatamente no interior da circunferência de raio 220.

A Figura 18 indica todos os pontos atingíveis com duas configurações de junta. Pode-se observar que a região é descrita por um círculo com raio igual a 220, coincidindo com o limite do volume de trabalho do manipulador SCARA Stäubli TS20.

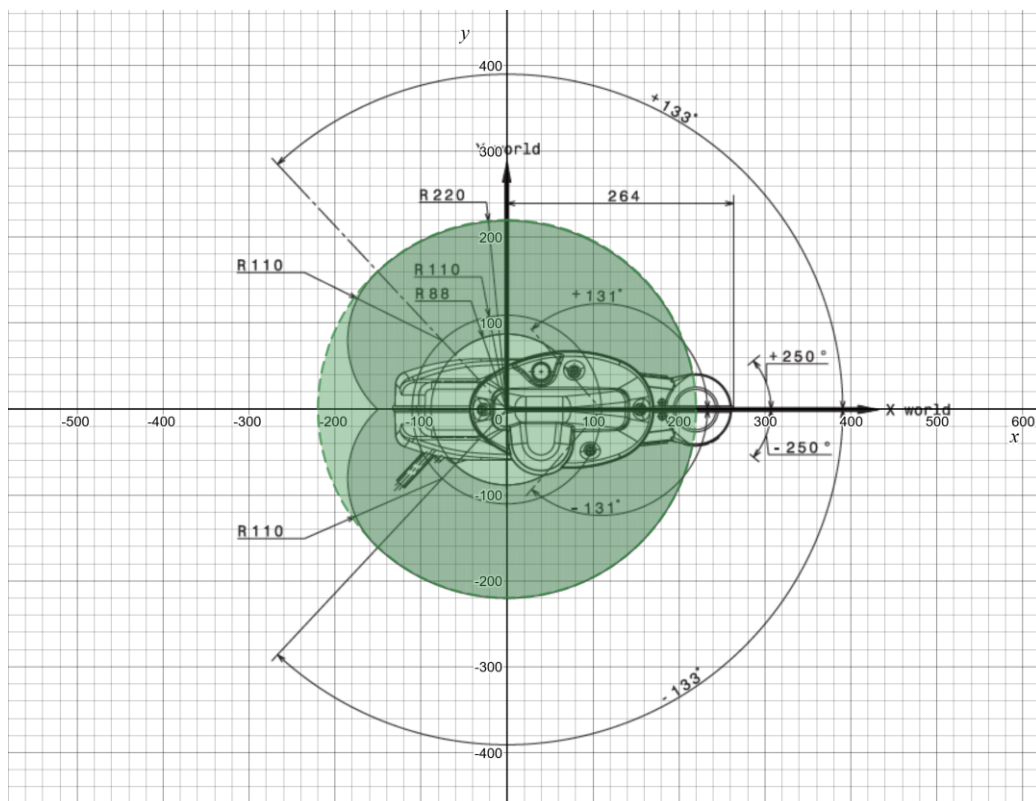


Figura 18: Pontos atingíveis com duas configurações de juntas

Igualando (147) a zero, obtém-se como gráfico uma circunferência com centro em  $(0,0)$  e raio 220, limite do volume trabalho do manipulador, exatamente igual ao gráfico gerado pela equação  $x^2 + y^2 - 220^2 = 0$ , apresentada na Seção 4.2.

Considerando os limites angulares impostos pela geometria do SCARA Stäubli TS20,  $-133^\circ \leq \theta_1 \leq 133^\circ$  e  $-131^\circ \leq \theta_3 \leq 131^\circ$ , pode-se observar na Figura 19 que a área determinada pelos polinômios da base de Gröbner coincide exatamente com o volume de trabalho do robô.

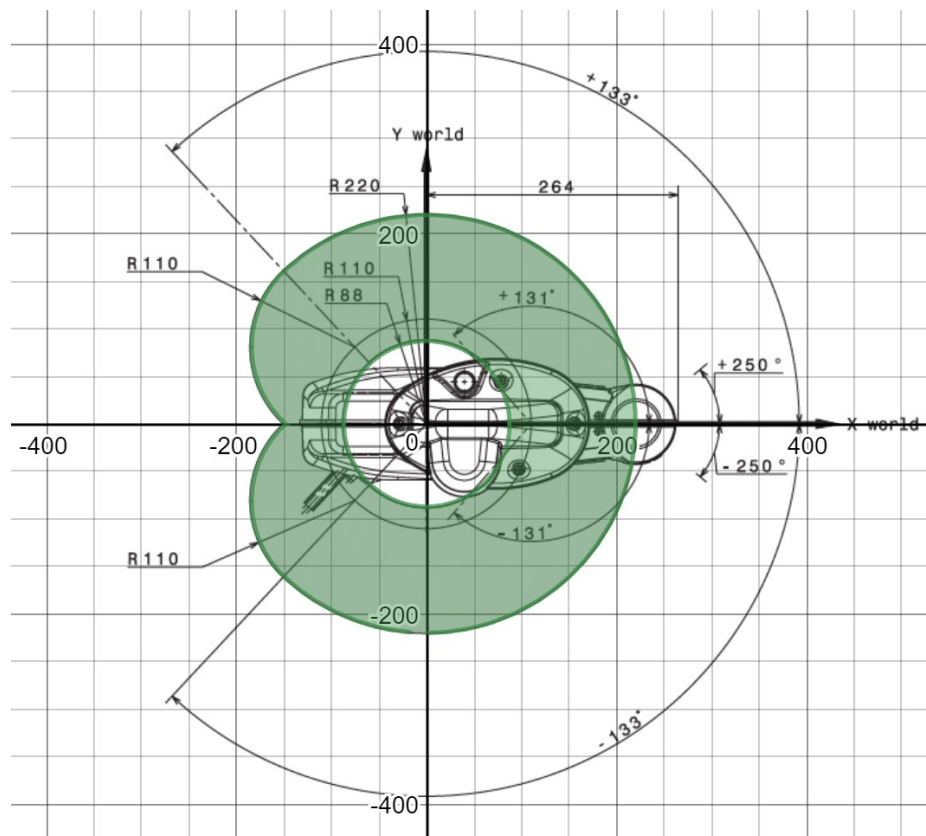


Figura 19: Volume de trabalho do manipulador TS20 por meio da Teoria de Bases de Gröbner

Retomando o caso em que é considerado um ponto no interior do volume de trabalho no manipulador SCARA, apresentado na Seção 4.4, os valores de  $\theta_1$  e  $\theta_3$  serão

determinados por intermédio da base de Gröbner produzida para o SCARA Stäubli TS20, com o objetivo de validar os polinômios gerados.

Inicialmente, com base no polinômio (142), encontram-se as raízes da Equação (150).

$$\begin{aligned} (48400x^2 + 48400y^2)s_1^2 + (-440x^2y - 440y^3)s_1 \\ + x^4 + 2x^2y^2 + y^4 - 48400x^2 = 0 \end{aligned} \quad (150)$$

São obtidas:

$$s_1' = \frac{y(x^2 + y^2) + x\sqrt{-(x^2 + y^2)(x^2 + y^2 - 220^2)}}{220(x^2 + y^2)}, \quad (151)$$

$$s_1'' = \frac{y(x^2 + y^2) - x\sqrt{-(x^2 + y^2)(x^2 + y^2 - 220^2)}}{220(x^2 + y^2)}. \quad (152)$$

Resumidamente:

$$s_1 = \frac{y(x^2 + y^2) \pm \sqrt{-(x^2 + y^2)(x^2 + y^2 - 220^2)}}{220(x^2 + y^2)}. \quad (153)$$

Os valores de  $s_1$  são determinados após as substituições de  $x$  e  $y$  por  $(55 + 55\sqrt{3})$ , de acordo com o exemplo apresentado na Seção 4.4. Obtém-se após aproximações de até cinco casas decimais:

$$s_1' \approx 0,86599, \quad (154)$$

$$s_1'' \approx 0,50011. \quad (155)$$

Dessa forma, são obtidos os valores aproximados para  $\theta_1$ :

$$\theta_1' \approx 59,99594^\circ, \quad (156)$$

$$\theta_1'' \approx 30,00728^\circ. \quad (157)$$

Para determinar  $\theta_3$ , os valores encontrados para  $s_1$  são substituídos juntamente com os valores de  $x$  e  $y$  por  $(55 + 55\sqrt{3})$  na Equação (158), por meio do polinômio (144) encontrado na base.

$$(220x^2 + 220y^2)s_1 + 24200xs_3 - x^2y - y^3 = 0 \quad (158)$$

Os valores de  $s_3$  são obtidos após aproximações de até cinco casas decimais:

$$s_3' \approx -0,49986, \quad (159)$$

$$s_3'' \approx 0,49986. \quad (160)$$

Desse modo, são obtidos os valores aproximados para  $\theta_3$ :

$$\theta_3' \approx -29,99074^\circ, \quad (161)$$

$$\theta_3'' \approx 29,99074^\circ. \quad (162)$$

Os valores encontrados são aproximadamente iguais aos valores determinados na Seção 4.4. Por conseguinte, pode-se confirmar que, por intermédio dos polinômios da base de Gröbner, gerada com base no modelo geométrico do manipulador robótico, é possível determinar os valores para  $\theta_1$  e  $\theta_3$  a fim de posicionar o efetuador em um ponto  $(x; y)$ .

### 5.3 CONFIGURAÇÕES GENÉRICAS

A partir deste ponto, serão apresentados dois novos casos: inicialmente, considerando  $x = 0$  para o robô TS20; no segundo caso, com o objetivo de ilustrar o alcance do modelo algébrico, considerar-se-á uma situação em que  $L_1 \neq L_3$ . Neste último caso, será utilizado um outro modelo de manipulador robótico do tipo SCARA, em que os comprimentos dos elos sejam distintos.

Considerando  $(x = 0)$ , a título de exemplificação, substituindo  $L_1$  e  $L_3$  por 110, usando a ordem lexicográfica  $c_3 \succ s_3 \succ c_1 \succ s_1$ , utiliza-se as equações:

$$f_1 = L_3(c_1c_3 - s_1s_3) + L_1c_1 - x, \quad f_2 = L_3(s_1c_3 + s_3c_1) + L_1s_1 - y, \quad f_3 = c_1^2 + s_1^2 - 1 \quad e$$

$f_4 = c_3^2 + s_3^2 - 1$  para gerar uma base de Gröbner; emprega-se o seguinte código no

MAPLE:

```
> with(Groebner);
> with(codegen, cost);
> r1:=L3*(c1*c3-s1*s3)+L1*c1-x:
> r2:=L3*(c1*s3+c3*s1)+L1*s1-y:
> r3:=c1^2+s1^2-1:
> r4:=c3^2+s3^2-1:
> x:=0:
> GB:=Groebner[Basis]([r1,r2,r3,r4],plex(c3,s3,c1,s1));
> math_operations:=cost(GB);
> V:={additions = 1, divisions = 4, functions = 8, multiplications = 1}:
> total_cost:=eval(cost(GB), V)flop;
```

Após execução do código, as seguintes saídas são apresentadas:

```
> GB := [2*L1*s1*y-L1^2+L3^2-y^2, 4*L1^2*c1^2*y^2+L1^4-2*L1^2*L3^2-
2*L1^2*y^2+L3^4-2*L3^2*y^2+y^4, L3*s3-c1*y, 2*L1*L3*c3+L1^2+L3^2-y^2]
> math_operations:=13*additions+41*multiplications
> total_cost:=54*flop
```

A base de Gröbner para este caso é:

$$g_1 = 2L_1s_1y - L_1^2 + L_3^2 - y^2, \quad (163)$$

$$g_2 = 4c_1^2L_1^2y^2 + L_1^4 - 2L_1^2L_3^2 - 2L_1^2y^2 + L_3^4 - 2L_3^2y^2 + y^4, \quad (164)$$

$$g_3 = -c_1y + L_3s_3, \quad (165)$$

$$g_4 = 2c_3L_1L_3 + L_1^2 + L_3^2 - y^2. \quad (166)$$

Pode-se observar que a base gerada é bem mais simples do que meramente substituir  $x = 0$  nos polinômios (138) a (141). Substituindo  $L_1$  e  $L_3$  por 110, a seguinte base é obtida:

$$g_1 = -y + 220s_1, \quad (167)$$

$$g_2 = 48400c_1^2 + y^2 - 48400, \quad (168)$$

$$g_3 = -c_1y + 110s_3, \quad (169)$$

$$g_4 = -y^2 + 24200c_3 + 24200. \quad (170)$$

Pode-se encontrar imediatamente o valor de  $y$  por meio das incógnitas  $s_1$  e  $c_3$  nos polinômios (167) e (170). O segundo elemento na base é um polinômio de segundo grau em termos de  $c_1$ , e resolvendo a equação considerando  $c_1$  como sendo a incógnita, até duas soluções são determinadas. Para cada uma delas, existe um valor correspondente para  $s_3$ , de acordo com o polinômio (169). Dessa forma, existem até duas configurações de juntas para posicionar o braço robótico no ponto  $(x; y)$  quando  $x = 0$ .

O discriminante da Equação (168) é dado por (171).

$$-193600y^2 + 9370240000 \quad (171)$$

Resolvendo a inequação (172), com o objetivo de determinar os pontos atingíveis pelo efetuador com duas configurações distintas:

$$-193600y^2 + 9370240000 > 0. \quad (172)$$

O intervalo  $-220 < y < 200$  é encontrado, confirmando que o efetuador do robô pode atingir os pontos  $(x; y)$ , com  $x = 0$ , em duas configurações distintas, dentro dos limites impostos pelo volume de trabalho do manipulador robótico.

As figuras a seguir ilustram essas configurações: a Figura 20 expõe duas configurações de juntas para  $x = 0$  e  $y > 0$ ; enquanto que na Figura 21 tem-se duas configurações para  $x = 0$  e  $y > 0$ .

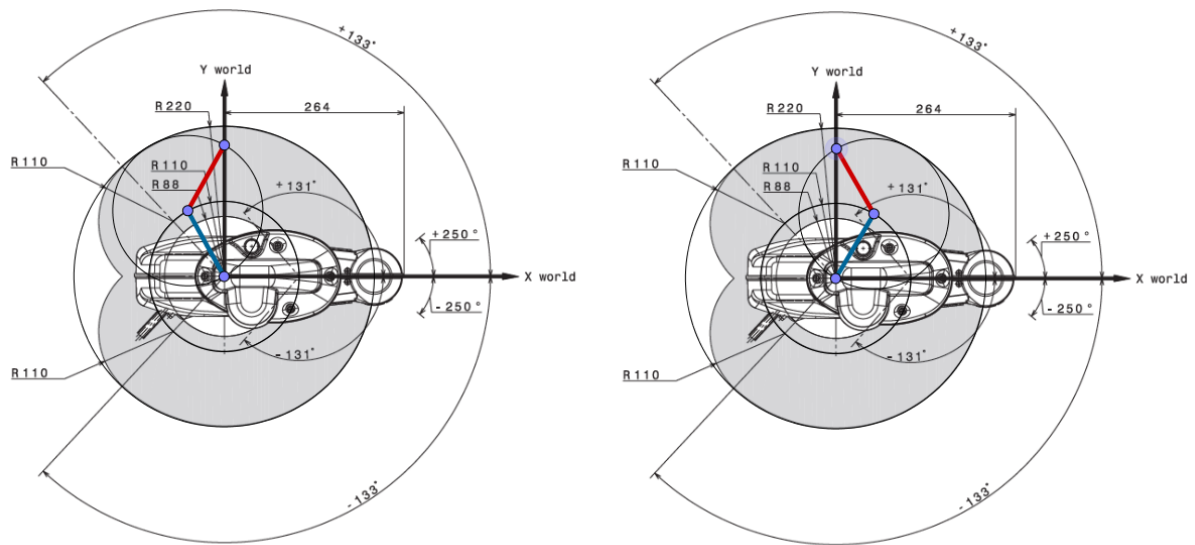


Figura 20: Configurações distintas para  $x = 0$  e  $y > 0$

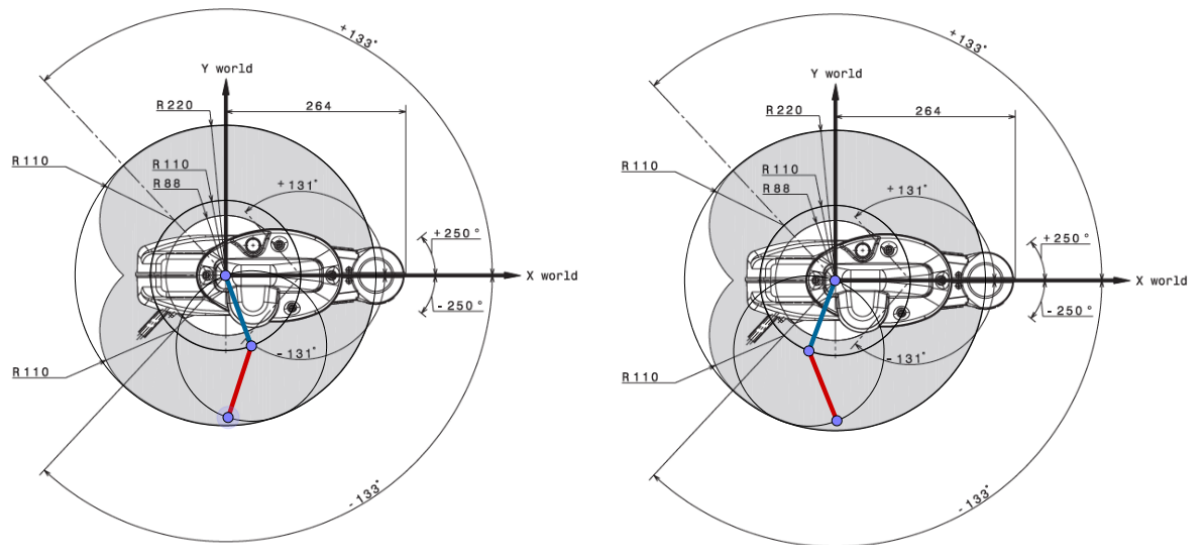


Figura 21: Configurações distintas para  $x = 0$  e  $y < 0$

Porém, o volume de trabalho do robô estudado é parte do interior da circunferência de raio 220, em virtude dos limites impostos aos ângulos das juntas  $\theta_1$  e  $\theta_3$ .

No segundo caso, com o objetivo de ilustrar o alcance do modelo algébrico, considerar-se-á agora uma situação em que  $L_1 \neq L_3$ . Neste caso, pode-se utilizar como

exemplo o modelo TS40, também produzido pela Stäubli Robotics, em que  $L_1 = 230$  e  $L_3 = 170$ .

Utiliza-se as equações:  $f_1 = 170(c_1c_3 - s_1s_3) + 230c_1 - x$ ,  $f_2 = 170(s_1c_3 + s_3c_1) + 230s_1 - y$ ,  $f_3 = c_1^2 + s_1^2 - 1$ ,  $f_4 = c_3^2 + s_3^2 - 1$ , usando a ordem lexicográfica  $c_3 \succ s_3 \succ c_1 \succ s_1$  e admitindo que  $c_3, s_3, c_1, s_1$  atuam como variáveis, obtém-se a seguinte base de Gröbner:

$$g_1 = \left(211600(x^2 + y^2)\right)s_1^2 + \left(-920(x^2y + y^3) - 22080000y\right)s_1 + x^4 + 2x^2y^2 + y^4 - 163600x^2 + 48000y^2 + 576000000, \quad (173)$$

$$g_2 = 460(c_1x + s_1y) - x^2 - y^2 - 24000, \quad (174)$$

$$g_3 = \left(460(x^2 + y^2)\right)s_1 + 78200xs_3 - x^2y - y^3 - 24000y, \quad (175)$$

$$g_4 = -x^2 - y^2 + 78200c_3 + 81800. \quad (176)$$

Ao analisar o discriminante do primeiro elemento na base, haverá até duas soluções reais para  $s_1$  quando:  $-846400x^2(x^2 + y^2 - 3600)(x^2 + y^2 - 160000) \geq 0$ .

O gráfico desta inequação é uma coroa circular com raios interno e externo iguais a 60 e 400, respectivamente. Esse é um resultado óbvio ao observar o modelo geométrico do manipulador.

Desse modo, não existem configurações de juntas para posicionar o órgão terminal do braço do robô em pontos no interior da circunferência de raio 60, como por exemplo, nos pontos  $(x, y) = (0, 0)$  e  $(x, y) = (40, 30)$ .

Pode-se afirmar que apenas os pontos  $(x, y)$  que obedecem à relação  $(L_1 - L_3) \leq \sqrt{x^2 + y^2} \leq (L_1 + L_3)$  podem ser atingidos. Este é um resultado óbvio ao observar a geometria do braço robótico na Figura 22.



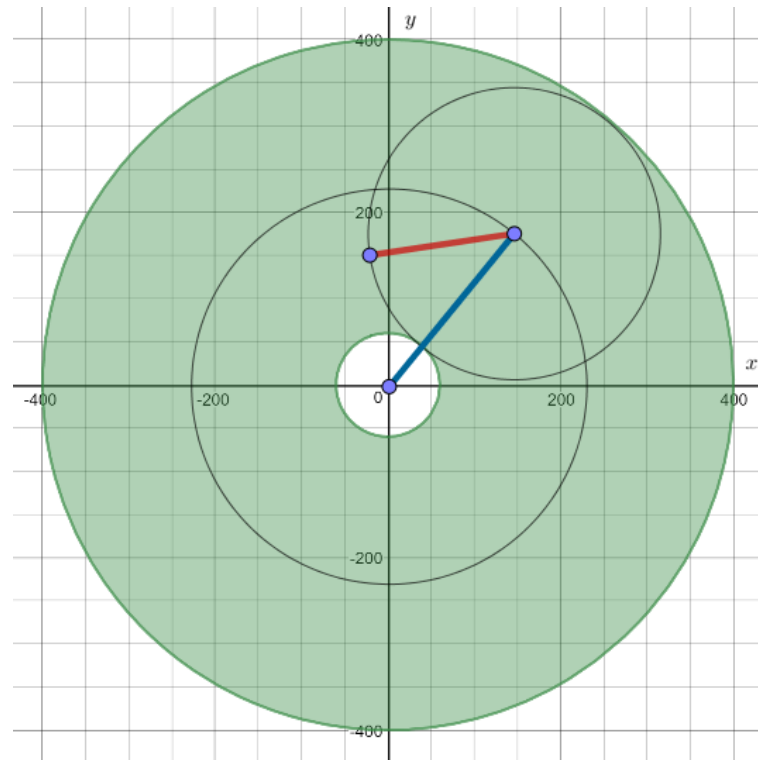


Figura 22: Modelo geométrico do SCARA Stäubli TS40

Vários modelos de robôs do tipo SCARA de diversos fabricantes, como FANUC, ABB, Kuka, entre outros, se enquadram na referida condição. À vista disso, essa análise adicional foi realizada. O modelo SCARA Stäubli TS40 pode ser observado na Figura 23.

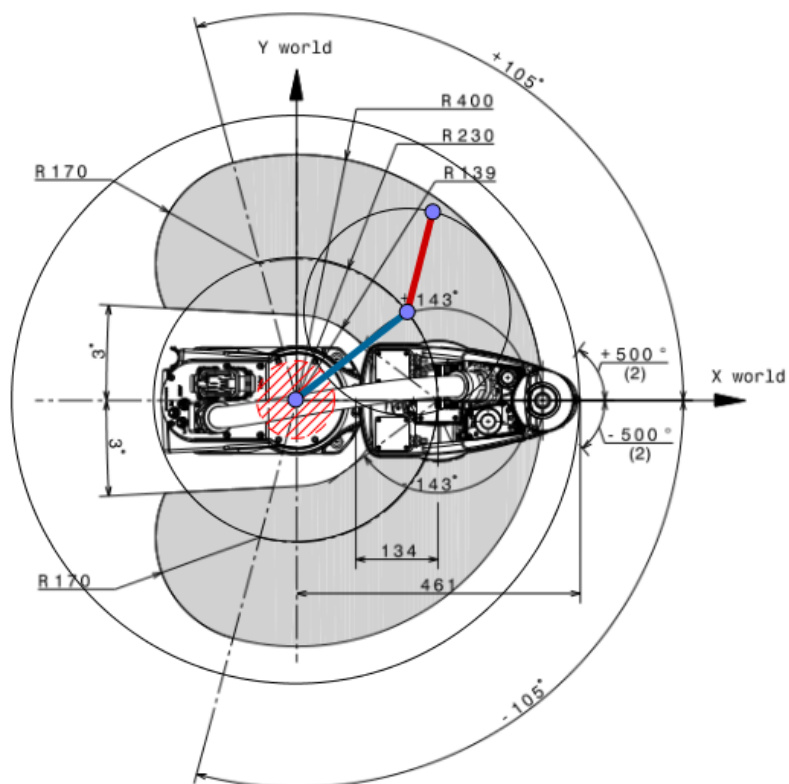


Figura 23: SCARA Stäubli TS40

Com base na Equação (177), pode-se mapear todos os pontos atingíveis pelo efetuador do robô TS40 com uma única configuração de juntas.

$$-846400x^2(x^2 + y^2 - 3600)(x^2 + y^2 - 160000) = 0 \quad (177)$$

As soluções ocorrem quando um dos fatores se anula, isto é:  $x^2 = 0$  ou  $x^2 + y^2 - 3600 = 0$ , ou  $x^2 + y^2 - 160000 = 0$ .

Pode-se observar na Figura 24 que a região atingível pelo efetuador do robô com uma única configuração de juntas é descrita pelos pontos pertencentes a uma das duas circunferências com raios iguais a 60 e 400. Isso coincide respectivamente com os limites interno e externo do volume de trabalho do manipulador robótico SCARA Stäubli TS40.

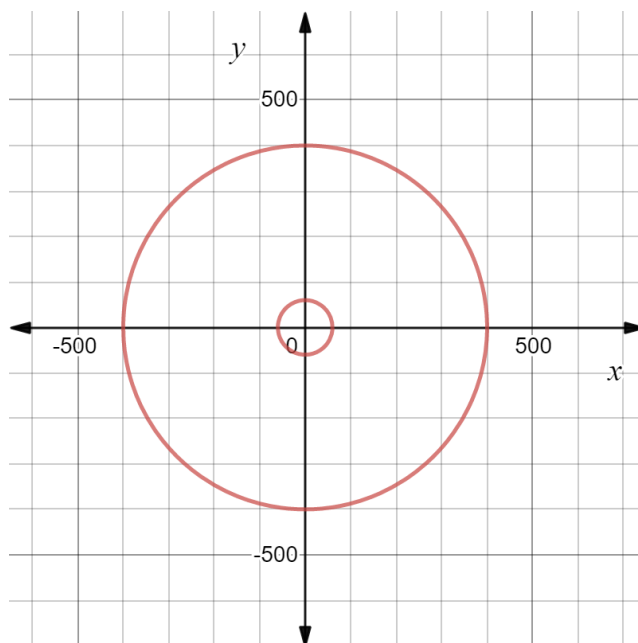


Figura 24: Pontos singulares de fronteira para o SCARA Stäubli TS40

Estes pontos devem ser evitados por conta das singularidades no limite de volume de trabalho do robô. O assunto aludido foi abordado com mais detalhes na Seção 4.2 deste trabalho.

A identificação de configurações singulares é importante por diversas razões:

- i. as singularidades representam configurações de juntas em que a mobilidade do manipulador é reduzida, não sendo possível um movimento arbitrário ao efetuador final;
- ii. pequenas velocidades do órgão terminal podem corresponder a grandes velocidades das juntas nas regiões de singularidade;
- iii. nas singularidades, pequenas forças e torques do órgão terminal podem corresponder a grandes forças e torques das juntas.

Nesse sentido, verifica-se a importância de utilizar o método das Bases de Gröbner para a solução da cinemática inversa de um manipulador do tipo SCARA, uma vez que, ao manipular os polinômios da base gerada, é possível determinar pontos

que só podem ser atingidos pelo efetuador com uma única configuração de junta. Neste caso, os pontos devem ser mapeados e evitados.

Para essa família de manipuladores robóticos, ao utilizar-se a Teoria de Bases de Gröbner, é possível determinar todas as configurações singulares sem a necessidade de emprego de um método complementar, como o Jacobiano, por exemplo, apresentado na Seção 4.3.

Ao analisar todo o desenvolvimento matemático desta seção, resumidamente, tem-se:

- i. existem exatamente duas soluções reais (configurações de juntas), considerando  $L_1 = L_3 = 110$ , quando  $x \neq 0$  e  $(x, y)$  satisfaz:

$$x^2 + y^2 < 220^2; \quad (178)$$

- ii. exatamente duas soluções reais (configurações de junta) são determinadas, considerando  $L_1 = L_3 = 110$ , quando  $(0, y)$  satisfaz:

$$-220 < y < 220; \quad (179)$$

- iii. os pontos  $(x, y)$  que não atendem qualquer das opções acima, uma vez que estão fora do volume de trabalho do robô, não representam soluções reais que satisfazem:

$$x^2 + y^2 > 220^2; \quad (180)$$

- iv. considerando  $L_1 > L_3$ , apenas os pontos  $(x, y)$  que satisfaçam a condição  $(L_1 - L_3) \leq \sqrt{x^2 + y^2} \leq (L_1 + L_3)$  podem ser atingidos.

### 5.3.1 Análise dos resultados obtidos

O robô estudado é um manipulador robótico considerado relativamente simples, formado por quatro graus de liberdade. Concentrando nos três primeiros graus de liberdade, a análise da geometria do robô tornou o problema mais simplificado e conseqüentemente, produziu equações menos complicadas de serem manipuladas.

Porém, se o método de Paul com base no algoritmo de Denavit-Hartenberg fosse aplicado a um braço robótico com seis graus de liberdade, determinar todas as soluções possíveis poderia ser um trabalho matematicamente exaustivo. A geometria do braço do robô pode ser complexa, e manipular o modelo cinemático para determinar mais soluções pode se tornar uma tarefa exaustiva.

Ao calcular uma base de Gröbner para robôs do tipo SCARA, é possível determinar todas as soluções possíveis. Em particular, usando a base de Gröbner, dado o ponto  $(x; y)$ , pode-se facilmente determinar o número máximo de configurações geométricas para posicionar o efetuador final do robô e, mais importante, é possível descobrir todas as soluções reais que produzem essas configurações.

Desta forma, é possível determinar todas as configurações singulares sem a necessidade de utilização de um método complementar, como o Jacobiano, por exemplo. Problemas de instabilidade que normalmente surgem quando os cálculos são executados por algoritmos numéricos em configurações quase singulares não ocorrem.

Porém, com relação ao custo computacional, faz-se necessário uma análise mais minuciosa das equações deduzidas nas Seções 4.2 e 5.2 para obtenção das variáveis de junta  $\theta_1$  e  $\theta_3$ .

Dividindo a Equação (153), deduzida na Seção 5.2, por  $c_1$ , obtido pela (143), 2<sup>o</sup> polinômio da base de Gröbner gerada na mesma seção, obtém-se a Equação (181).

$$\theta_1 = \text{Atan2} \left( \begin{array}{l} \left( y(x^2 + y^2) \pm x\sqrt{-(x^2 + y^2)(x^2 + y^2 - (220^2))} \right), \\ \left( x(x^2 + y^2) \mp y\sqrt{-(x^2 + y^2)(x^2 + y^2 - (220^2))} \right) \end{array} \right) \quad (181)$$

Quando  $s_3$  é isolado a partir de (144), 3<sup>o</sup> polinômio da base de Gröbner gerada na Seção 5.2,  $s_1$  é substituído nessa mesma equação; e o resultado dessa operação é dividido por  $c_3$  isolado em (145), 4<sup>o</sup> polinômio da base de Gröbner gerada nessa mesma seção, obtém-se a Equação (182).

$$\theta_3 = \text{Atan2} \left( \pm\sqrt{-(x^2 + y^2)(x^2 + y^2 - (220^2))}, (x^2 + y^2 - 2(110)^2) \right) \quad (182)$$

Observa-se que as Equações (181) e (182) oferecem as mesmas soluções determinadas pelas equações deduzidas na Seção 4.2 para solução da cinemática inversa do manipulador SCARA Stäubli TS20 com base no método de Paul, conforme esperado para as duas aplicações, porém com o mesmo custo computacional.

Pode-se afirmar que, com relação ao custo computacional, não há vantagem em se utilizar a Teoria de Bases de Gröbner para solucionar a cinemática inversa de manipuladores do tipo SCARA. As 129 operações de ponto flutuante para gerar a base de Gröbner representam um custo computacional extra. Isso ocorre devido à simplicidade geométrica desta família de manipuladores robóticos.

Pelo aludido motivo, um manipulador robótico antropomórfico será utilizado como estudo de caso, uma vez que um robô com esta geometria tem a sua cinemática inversa solucionada por meio de equações não lineares, transcendentais e que são resolvidas com elevado custo computacional.

## 6 CINEMÁTICA INVERSA DO PUMA 560 POR MEIO DO MÉTODO DE PAUL

O Unimation PUMA (*Programmable Universal Machine for Assembly*) 560 é um manipulador robótico com seis graus de liberdade e todas as juntas rotativas, ou seja, é um mecanismo 6R. De acordo com Ghosal (2006), o PUMA 560 é um manipulador robótico antropomórfico visto extensivamente em livros da área, de instituições acadêmicas e de pesquisa.

Segundo Craig (2017), o PUMA 560 é um manipulador robótico industrial antropomórfico desenvolvido por Victor Scheinman na empresa Unimation. Inicialmente desenvolvido para a General Motors, o PUMA 560 foi baseado em projetos desenvolvidos por Scheinman na Universidade de Stanford.

Esse manipulador possui seis graus de liberdade, sendo todas as seis juntas rotativas. Por esse motivo, ele é classificado com um manipulador 6R. As características dele estão descritas a seguir.

- i. Alcance máximo de 878 mm do eixo central ao centro do pulso;
- ii. Massa do braço: 83 kg (aproximadamente);
- iii. Repetibilidade:  $\pm 0,1$  mm;
- iv. Velocidade máxima (manipulação de uma carga de 2,5 kg): movimentos em linha reta de 500 mm/s;
- v. Velocidade máxima (manipulação de uma carga de 4,0 kg): movimentos em linha reta de 470 mm/s.

O Unimation PUMA 560 é classificado, quanto à sua geometria, como um manipulador antropomórfico 6R com pulso esférico, uma vez os seus três últimos eixos se cruzam, permitindo a rotação em torno de três eixos simultaneamente. Pulsos esféricos (*spherical wrists*) são dispositivos robóticos que desacoplam os movimentos de

posicionamento e orientação nos robôs manipuladores e são necessários para operações que exigem agilidade do manipulador.

## 6.1 CINEMÁTICA DIRETA DO PUMA 560

O Unimation PUMA 560 é representado na Figura 25, com os seus seis sistemas de referência fixados aos elos.

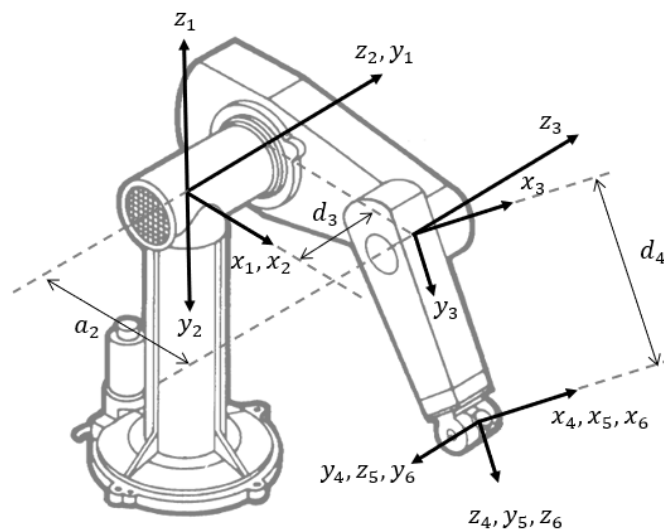


Figura 25: Sistemas de referência do Puma 560  
Fonte: Adaptado de (CHAKRAVARTI e SIVAKUMAR, 2016)

Pode-se observar que o sistema de referência  $\{0\}$ , não mostrado na Figura 25, coincide com o sistema de referência  $\{1\}$  quando  $\theta_1$  é nulo. Nota-se também que para o manipulador Puma 560, assim como para outros robôs industriais, os eixos das juntas 4, 5 e 6 se cruzam em um ponto comum, e esse ponto de intersecção coincide com a origem dos sistemas de referência  $\{4\}$ ,  $\{5\}$  e  $\{6\}$ , sendo que os eixos dessas três juntas são mutualmente ortogonais.



De acordo com Craig (2017), para determinar uma matriz geral de transformação  $T$  para a solução da cinemática direta do manipulador Puma 560, faz-se necessário construir uma transformação que define o sistema  $\{i\}$  em relação ao sistema  $\{i-1\}$ ; dessa forma pretende-se determinar  ${}^{i-1}A_i$ . Ao definir um sistema de referência para cada elo, desmembra-se o problema cinemático em  $n$  subproblemas, e com o objetivo de solucioná-los, desmembra-se ainda cada subproblema em quatro outros.

Inicialmente, definem-se três sistemas de referência intermediários  $\{P\}$ ,  $\{Q\}$  e  $\{R\}$  para cada elo do manipulador. Observa-se na Figura 26 que o sistema de referência  $\{R\}$  difere do sistema de referência  $\{i-1\}$  apenas por uma rotação de  $\alpha_{i-1}$ ; o sistema de referência  $\{Q\}$  difere de  $\{R\}$  por uma translação  $a_{i-1}$ ; o sistema de referência  $\{P\}$  difere de  $\{Q\}$  por uma rotação  $\theta_i$ ; e o sistema de referência  $\{i\}$  difere de  $\{P\}$  por uma translação  $d_i$ .

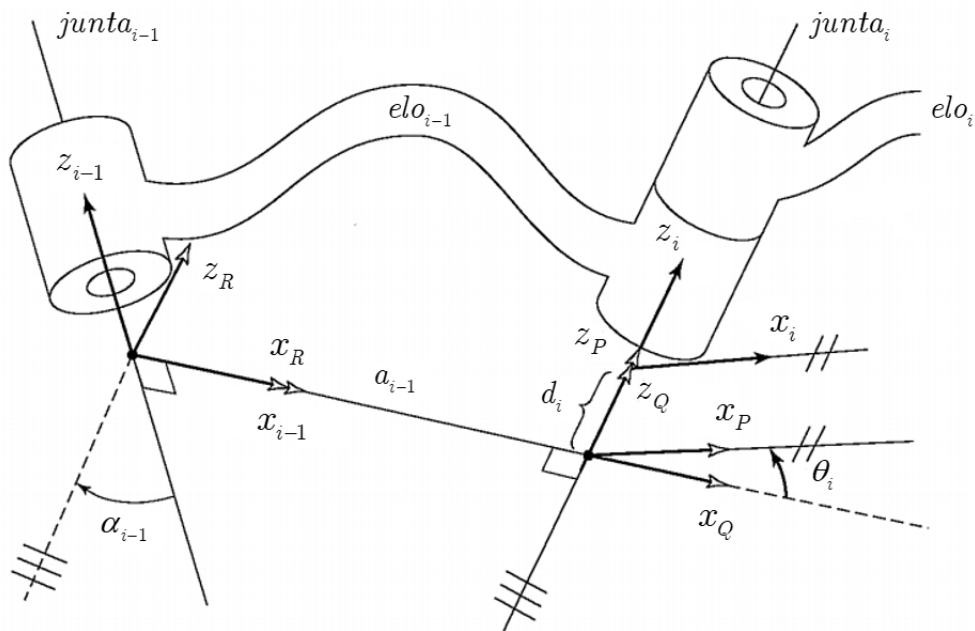


Figura 26: Localização dos sistemas de referência intermediários  $\{P\}$ ,  $\{Q\}$  e  $\{R\}$

Fonte: Adaptado de (CRAIG, 2017)

Desta forma, tem-se:

$${}^{i-1}A_i = Rot_{x,\alpha_{i-1}} \cdot Transl_{x,a_{i-1}} \cdot Rot_{z,\theta_i} \cdot Transl_{z,d_i}$$

$$= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} & 0 & a_{i-1} \\ s_{\theta_i} c_{\alpha_{i-1}} & c_{\theta_i} c_{\alpha_{i-1}} & -s_{\alpha_{i-1}} & -s_{\alpha_{i-1}} d_i \\ s_{\theta_i} s_{\alpha_{i-1}} & c_{\theta_i} s_{\alpha_{i-1}} & c_{\alpha_{i-1}} & c_{\alpha_{i-1}} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (183)$$

Todos os parâmetros cinemáticos deste manipulador estão listados na Tabela 4, em que  $i$  representa o número da junta.

Tabela 4: Parâmetros cinemáticos do Puma 560

$i$	$\alpha_{i-1}$ (graus)	$a_{i-1}$ (mm)	$d_i$ (mm)	$\theta_i$ (graus)
1	0	0	0	$\theta_1$
2	-90	0	0	$\theta_2$
3	0	$a_2$	$d_3$	$\theta_3$
4	-90	$a_3$	$d_4$	$\theta_4$
5	90	0	0	$\theta_5$
6	-90	0	0	$\theta_6$

Após a construção de todas as matrizes de transformação (184) a (189) e multiplicação delas (190), obtém-se a matriz de transformação homogênea (191), solução do problema da cinemática direta do manipulador Unimation PUMA 560.

$${}^0A_1 = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (184)$$

$${}^1A_2 = \begin{bmatrix} c_2 & -s_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_2 & -c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (185)$$

$${}^2A_3 = \begin{bmatrix} c_3 & -s_3 & 0 & a_2 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (186)$$

$${}^3A_4 = \begin{bmatrix} c_4 & -s_4 & 0 & a_3 \\ 0 & 0 & 1 & d_4 \\ -s_4 & -c_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (187)$$

$${}^4A_5 = \begin{bmatrix} c_5 & -s_5 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_5 & c_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (188)$$

$${}^5A_6 = \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_6 & -c_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (189)$$

$$T = {}^0A_1(\theta_1) \cdot {}^1A_2(\theta_2) \cdot {}^2A_3(\theta_3) \cdot {}^3A_4(\theta_4) \cdot {}^4A_5(\theta_5) \cdot {}^5A_6(\theta_6) \quad (190)$$

$$T = \begin{bmatrix} n_x & o_x & a_x & x \\ n_y & o_y & a_y & y \\ n_z & o_z & a_z & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (191)$$

Em que as entradas de  $T$  são dadas por:

$$n_x = c_1 (c_{23} (c_4 c_5 c_6 - s_4 s_5) - s_{23} s_5 c_5) + s_1 (s_4 c_5 c_6 + c_4 s_6), \quad (192)$$

$$n_y = s_1 (c_{23} (c_4 c_5 c_6 - s_4 s_6) - s_{23} s_5 c_6) - c_1 (s_4 c_5 c_6 + c_4 s_6), \quad (193)$$

$$n_z = -s_{23} (c_4 c_5 c_6 - s_4 s_6) - c_{23} s_5 c_6, \quad (194)$$

$$o_x = c_1 (c_{23} (-c_4 c_5 s_6 - s_4 c_6) + s_{23} s_5 s_6) + s_1 (c_4 c_6 - s_4 c_5 s_6), \quad (195)$$

$$o_y = s_1 (c_{23} (-c_4 c_5 s_6 - s_4 c_6) + s_{23} s_5 s_6) - c_1 (c_4 c_6 - s_4 c_5 s_6), \quad (196)$$

$$o_z = -s_{23} (-c_4 c_5 s_6 - s_4 c_6) + c_{23} s_5 s_6, \quad (197)$$

$$a_x = -c_1 (c_{23} c_4 s_5 + s_{23} c_5) - s_1 s_4 s_5, \quad (198)$$

$$a_y = -s_1 (c_{23} c_4 s_5 + s_{23} c_5) + c_1 s_4 s_5, \quad (199)$$

$$a_z = s_{23}c_4s_5 - c_{23}c_5, \quad (200)$$

$$x = c_1 (a_2c_2 + a_3c_{23} - d_4s_{23}) - d_3s_1, \quad (201)$$

$$y = s_1 (a_2c_2 + a_3c_{23} - d_4s_{23}) + d_3c_1, \quad (202)$$

$$z = -a_3s_{23} - a_2s_2 - d_4c_{23}. \quad (203)$$

De acordo com Wenz e Worn (2007), são necessárias 544 operações de ponto flutuante (flop) para a solução do problema cinemático direto de um manipulador 6R com base na matriz (191).

A Tabela 5 ilustra o detalhamento de todas as operações matemáticas realizadas para solução do problema cinemático direto de um manipulador 6R.

Tabela 5: Número de operações para solução da cinemática direta do PUMA 560

+	-	*	/	cos	sen	Atan2	$\sqrt{\quad}$	$\wedge 2$
54	18	128	0	21	22	0	0	0

Utiliza-se a Tabela 3, apresentada na subseção 4.4.1, para determinar o número total de operações de ponto flutuante para solução da cinemática direta do PUMA 560.

## 6.2 CINEMÁTICA INVERSA DO PUMA 560

Segundo Craig (2017), para a solução da cinemática inversa de um manipulador robótico com seis graus de liberdade, tem-se 12 equações e 6 incógnitas. No entanto, entre as 9 equações que surgem da porção de matriz rotacional de  $T$ , somente três são independentes. Estas, somadas às equações da porção de vetor posição de  $T$ , resultam

em seis equações com seis incógnitas. Tais equações são não lineares, transcendentais, e que são resolvidas com elevado custo computacional.

A abordagem sistemática para resolver as equações cinemáticas, proposta por Paul (1981), reescreve as equações cinemáticas diretas de várias maneiras. Desse modo, ambos os lados da equação matricial da cinemática direta são pré e pós-multiplicados por matrizes de transformação inversa. Ao fazer isso, são produzidas equações equivalentes que possuem as mesmas soluções. Cada equação matricial fornece 12 equações não lineares provenientes das três principais linhas da matriz. A quarta linha da matriz é sempre trivial.

O principal problema é encontrar nesse universo, equações adequadas que possam ser resolvidas analiticamente.

Pretende-se solucionar

$$T = \begin{bmatrix} n_x & o_x & a_x & x \\ n_y & o_y & a_y & y \\ n_z & o_z & a_z & z \\ 0 & 0 & 0 & 1 \end{bmatrix} = {}^0A_1(\theta_1) \cdot {}^1A_2(\theta_2) \cdot {}^2A_3(\theta_3) \cdot {}^3A_4(\theta_4) \cdot {}^4A_5(\theta_5) \cdot {}^5A_6(\theta_6) \quad (204)$$

e determinar  $(\theta_i)$  quando  $T$  é dado com valores numéricos.

Reescreve-se (204) colocando a dependência de  $(\theta_1)$  do lado esquerdo da equação, essência do método de Paul:

$$\left[ {}^0A_1(\theta_1) \right]^{-1} \cdot T = {}^1A_2(\theta_2) \cdot {}^2A_3(\theta_3) \cdot {}^3A_4(\theta_4) \cdot {}^4A_5(\theta_5) \cdot {}^5A_6(\theta_6). \quad (205)$$

Invertendo  ${}^0A_1(\theta_1)$ , a Equação (205) é reescrita como:

$$\begin{aligned} & \begin{bmatrix} c_1 & s_1 & 0 & 0 \\ -s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} n_x & o_x & a_x & x \\ n_y & o_y & a_y & y \\ n_z & o_z & a_z & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ & = \begin{bmatrix} c_1 n_x + s_1 n_y & c_1 o_x + s_1 o_y & c_1 a_x + s_1 a_y & c_1 x + s_1 y \\ c_1 n_y - s_1 n_x & c_1 o_y - s_1 o_x & -s_1 a_x + c_1 a_y & c_1 y - s_1 x \\ n_z & o_z & a_z & z \\ 0 & 0 & 0 & 1 \end{bmatrix} = {}^1A_6(\theta_i), \end{aligned} \quad (206)$$

sendo  ${}^1A_6(\theta_i)$  dado por:

$$\begin{aligned}
 {}^1A_6(\theta_i) &= {}^1A_2(\theta_2) \cdot {}^2A_3(\theta_3) \cdot {}^3A_4(\theta_4) \cdot {}^4A_5(\theta_5) \cdot {}^5A_6(\theta_6) \\
 &= \begin{bmatrix} c_2 & -s_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_2 & -c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} c_3 & -s_3 & 0 & a_2 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} c_4 & -s_4 & 0 & a_3 \\ 0 & 0 & 1 & d_4 \\ -s_4 & -c_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &\cdot \begin{bmatrix} c_5 & -s_5 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_5 & c_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_6 & -c_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \eta & \lambda & \mu & \varpi \\ \mathcal{G} & \sigma & \zeta & \tau \\ \nu & \omega & \xi & \psi \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{207}
 \end{aligned}$$

em que as entradas de  ${}^1A_6(\theta_i)$  são dadas por:

$$\eta = c_{23}(c_4c_5c_6 - s_4s_6) - s_{23}s_5s_6, \tag{208}$$

$$\mathcal{G} = -s_4c_5c_6 - c_4s_6, \tag{209}$$

$$\nu = -s_{23}(c_4c_5c_6 - s_4s_6) - c_{23}s_5s_6, \tag{210}$$

$$\lambda = -c_{23}(c_4c_5s_6 + s_4c_6) + s_{23}s_5s_6, \tag{211}$$

$$\sigma = s_4c_5s_6 - c_4c_6, \tag{212}$$

$$\omega = s_{23}(c_4c_5s_6 + s_4c_6) + c_{23}s_5s_6, \tag{213}$$

$$\mu = -c_{23}c_4s_5 - s_{23}c_5, \tag{214}$$

$$\zeta = s_4s_5, \tag{215}$$

$$\xi = s_{23}c_4s_5 - c_{23}c_5, \tag{216}$$

$$\varpi = a_2c_2 + a_3c_{23} - d_4s_{23}, \tag{217}$$

$$\tau = d_3, \tag{218}$$

$$\psi = -a_3s_{23} - a_2s_2 - d_4c_{23}. \tag{219}$$

Este método, proposto por Paul (1981), é utilizado com o objetivo de separar as variáveis em busca de uma equação solucionável.

Igualando os elementos (linha 2, coluna 4) dos dois membros da Equação (206), tem-se:

$$-s_1x + c_1y = d_3. \quad (220)$$

Para solucionar uma equação com esta forma, são feitas as seguintes substituições trigonométricas:

$$x = \rho c_\phi, \quad (221)$$

$$y = \rho s_\phi, \quad (222)$$

em que:

$$\rho = \sqrt{x^2 + y^2}, \quad (223)$$

$$\phi = \text{Atan2}(y, x). \quad (224)$$

Substituindo (221) e (222) em (220), obtém-se:

$$c_1s_\phi - s_1c_\phi = \frac{d_3}{\rho}. \quad (225)$$

Com base na equação para diferença de ângulos, tem-se:

$$\text{sen}(\phi - \theta_1) = \frac{d_3}{\rho}. \quad (226)$$

Dessa forma:

$$\cos(\phi - \theta_1) = \pm \sqrt{1 - \frac{d_3^2}{\rho^2}}. \quad (227)$$

Com base nas Equações (226) e (227), determina-se (228).

$$\begin{aligned} \phi - \theta_1 &= \text{Atan2}\left(\frac{d_3}{\rho}, \pm \sqrt{1 - \frac{d_3^2}{\rho^2}}\right) \\ &= \text{Atan2}\left(\frac{d_3}{\rho}, \pm \sqrt{\frac{\rho^2 - d_3^2}{\rho^2}}\right) \\ &= \text{Atan2}\left(\frac{d_3}{\rho}, \pm \frac{\sqrt{(\sqrt{x^2 + y^2})^2 - d_3^2}}{\rho}\right) \\ &= \text{Atan2}\left(\frac{d_3}{\rho}, \pm \frac{\sqrt{x^2 + y^2 - d_3^2}}{\rho}\right) \end{aligned} \quad (228)$$

Finalmente,  $\theta_1$  pode ser determinado por meio da Equação (229).

$$\theta_1 = \text{Atan2}(y, x) - \text{Atan2}\left(d_3, \pm\sqrt{x^2 + y^2 - d_3^2}\right) \quad (229)$$

Observa-se na Equação (229), devido ao sinal de mais ou menos, que duas soluções possíveis para  $\theta_1$  podem ser determinadas.

Igualando tanto os elementos (linha 1, coluna 4) quanto os elementos (linha 3, coluna 4) dos dois membros da Equação (206), tem-se:

$$c_1x + s_1y = a_2c_2 + a_3c_{23} - d_4s_{23}, \quad (230)$$

$$-z = a_3s_{23} + a_2s_2 + d_4c_{23}. \quad (231)$$

Elevam-se ao quadrado as Equações (230), (231) e (220).

$$\begin{aligned} c_1^2x^2 + 2c_1s_1xy + s_1^2y^2 &= a_2^2c_2^2 + 2a_2a_3c_2c_{23} \\ &\quad - 2a_2c_2d_4s_{23} + a_3^2c_{23}^2 - 2a_3c_{23}d_4s_{23} + d_4^2s_{23}^2 \end{aligned} \quad (232)$$

$$z^2 = a_2^2s_2^2 + 2a_2a_3s_2s_{23} + 2a_2c_{23}d_4s_2 + a_3^2s_{23}^2 + 2a_3c_{23}d_4s_{23} + c_{23}^2d_4^2 \quad (233)$$

$$c_1^2y^2 - 2c_1s_1xy + s_1^2x^2 = d_3^2 \quad (234)$$

Somando as equações resultantes, obtém-se (235).

$$\begin{aligned} c_1^2x^2 + 2c_1s_1xy + s_1^2y^2 + z^2 + c_1^2y^2 - 2c_1s_1xy + s_1^2x^2 &= a_2^2c_2^2 \\ + 2a_2a_3c_2c_{23} - 2a_2c_2d_4s_{23} + a_3^2c_{23}^2 - 2a_3c_{23}d_4s_{23} + d_4^2s_{23}^2 + a_2^2s_2^2 \\ + 2a_2a_3s_2s_{23} + 2a_2c_{23}d_4s_2 + a_3^2s_{23}^2 + 2a_3c_{23}d_4s_{23} + c_{23}^2d_4^2 + d_3^2 \end{aligned} \quad (235)$$

Após algumas simplificações, a Equação (236) é obtida.

$$x^2 + y^2 + z^2 - a_2^2 - a_3^2 - d_3^2 - d_4^2 = 2a_2\left(a_3(c_2c_{23} + s_2s_{23}) + d_4(s_2c_{23} - c_2s_{23})\right) \quad (236)$$

A partir deste ponto,  $c_{23}$  e  $s_{23}$  são substituídos respectivamente por  $(c_2c_3 - s_2s_3)$  e  $(s_2c_3 + s_3c_2)$ , com o objetivo de produzir a Equação (241).

$$x^2 + y^2 + z^2 - a_2^2 - a_3^2 - d_3^2 - d_4^2 = 2a_2 \begin{pmatrix} a_3(c_2(c_2c_3 - s_2s_3) + s_2(s_2c_3 + s_3c_2)) \\ -d_4(c_2(s_2c_3 + s_3c_2) - s_2(c_2c_3 - s_2s_3)) \end{pmatrix} \quad (237)$$

$$x^2 + y^2 + z^2 - a_2^2 - a_3^2 - d_3^2 - d_4^2 = 2a_2 \begin{pmatrix} a_3(c_2^2c_3 - c_2s_2s_3 + s_2^2c_3 + c_2s_2s_3) \\ -d_4(c_2s_2c_3 + c_2^2s_3 - c_2s_2c_3 + s_2^2s_3) \end{pmatrix} \quad (238)$$



$$x^2 + y^2 + z^2 - a_2^2 - a_3^2 - d_3^2 - d_4^2 = 2a_2 \begin{pmatrix} a_3 (c_3 (c_2^2 + s_2^2) - c_2 s_2 s_3 + c_2 s_2 s_3) \\ -d_4 (s_3 (c_2^2 + s_2^2) + c_2 s_2 c_3 - c_2 s_2 c_3) \end{pmatrix} \quad (239)$$

$$x^2 + y^2 + z^2 - a_2^2 - a_3^2 - d_3^2 - d_4^2 = 2a_2 (a_3 c_3 - d_4 s_3) \quad (240)$$

$$\left( \frac{x^2 + y^2 + z^2 - a_2^2 - a_3^2 - d_3^2 - d_4^2}{2a_2} \right) = a_3 c_3 - d_4 s_3 \quad (241)$$

$$K = a_3 c_3 - d_4 s_3 \quad (242)$$

Em que:

$$K = \left( \frac{x^2 + y^2 + z^2 - a_2^2 - a_3^2 - d_3^2 - d_4^2}{2a_2} \right). \quad (243)$$

Pode-se observar que a dependência de  $\theta_1$  foi eliminada na Equação (241). A Equação (242) tem a mesma forma que a Equação (220) e, portanto, pode ser resolvida utilizando o mesmo tipo de substituição trigonométrica para gerar uma solução para  $\theta_3$ . Dessa forma, tem-se:

$$-s_3 d_4 + c_3 a_3 = K. \quad (244)$$

Para solucionar uma equação com esta forma, são feitas as substituições trigonométricas:

$$d_4 = \lambda c_\varphi, \quad (245)$$

$$a_3 = \lambda s_\varphi. \quad (246)$$

em que:

$$\lambda = \sqrt{d_4^2 + a_3^2}, \quad (247)$$

$$\varphi = \text{Atan2}(a_3, d_4). \quad (248)$$

Substituindo (245) e (246) em (244), obtém-se:

$$c_3 s_\varphi - s_3 c_\varphi = \frac{K}{\lambda}. \quad (249)$$

A partir da equação para diferença de ângulos, tem-se:

$$\text{sen}(\varphi - \theta_3) = \frac{K}{\lambda}. \quad (250)$$

Desse modo:

$$\cos(\varphi - \theta_3) = \pm \sqrt{1 - \frac{K^2}{\lambda^2}}. \quad (251)$$

Com base nas Equações (250) e (251), determina-se (252).

$$\begin{aligned} \varphi - \theta_3 &= \text{Atan2}\left(\frac{K}{\lambda}, \pm \sqrt{1 - \frac{K^2}{\lambda^2}}\right) \\ &= \text{Atan2}\left(\frac{K}{\lambda}, \pm \sqrt{\frac{\lambda^2 - K^2}{\lambda^2}}\right) \\ &= \text{Atan2}\left(\frac{K}{\lambda}, \pm \frac{\sqrt{(\sqrt{x^2 + y^2})^2 - K^2}}{\lambda}\right) \\ &= \text{Atan2}\left(\frac{K}{\lambda}, \pm \frac{\sqrt{x^2 + y^2 - K^2}}{\lambda}\right) \end{aligned} \quad (252)$$

Finalmente,  $\theta_3$  pode ser determinado pela Equação (253).

$$\theta_3 = \text{Atan2}(a_3, d_4) - \text{Atan2}\left(K, \pm \sqrt{a_3^2 + d_4^2 - (K)^2}\right) \quad (253)$$

O sinal de mais ou menos em (253) produz duas soluções distintas para  $\theta_3$ . A partir deste ponto, a Equação (204) é reescrita de forma que todo o lado esquerdo seja uma função de apenas termos conhecidos e  $\theta_2$ .

$$[{}^2A_3(\theta_3)]^{-1} \cdot [{}^1A_2(\theta_2)]^{-1} \cdot [{}^0A_1(\theta_1)]^{-1} \cdot T = {}^3A_4(\theta_4) \cdot {}^4A_5(\theta_5) \cdot {}^5A_6(\theta_6) \quad (254)$$

$$\begin{aligned} &\begin{bmatrix} c_3 & s_3 & 0 & -a_2c_3 \\ -s_3 & c_3 & 0 & a_2s_3 \\ 0 & 0 & 1 & -d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} c_2 & 0 & -s_2 & 0 \\ -s_2 & 0 & -c_2 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &\cdot \begin{bmatrix} c_1 & s_1 & 0 & 0 \\ -s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} n_x & o_x & a_x & x \\ n_y & o_y & a_y & y \\ n_z & o_z & a_z & z \\ 0 & 0 & 0 & 1 \end{bmatrix} = {}^3A_6(\theta_i) \end{aligned} \quad (255)$$

$$\begin{bmatrix} \eta^* & \lambda^* & \mu^* & \varpi^* \\ \varrho^* & \sigma^* & \zeta^* & \tau^* \\ \upsilon^* & \omega^* & \xi^* & \psi^* \\ 0 & 0 & 0 & 1 \end{bmatrix} = {}^3A_6(\theta_i) \quad (256)$$

Em que as entradas de  ${}^3A_6(\theta_i)$  são dadas por:

$$\eta^* = (c_3c_2 - s_3s_2)(c_1n_x + s_1n_y) + (-s_3c_2 - c_3s_2)n_z, \quad (257)$$

$$\lambda^* = (c_3c_2 - s_3s_2)(c_1o_x + s_1o_y) + (-s_3c_2 - c_3s_2)o_z, \quad (258)$$

$$\mu^* = (c_3c_2 - s_3s_2)(c_1a_x + s_1a_y) + (-s_3c_2 - c_3s_2)a_z, \quad (259)$$

$$\varpi^* = (c_3c_2 - s_3s_2)(c_1x + s_1y) + (-s_3c_2 - c_3s_2)z - a_2c_3, \quad (260)$$

$$\varrho^* = (-s_3c_2 - c_3s_2)(c_1n_x + s_1n_y)(-c_3c_2 + s_3s_2)n_z, \quad (261)$$

$$\sigma^* = (-s_3c_2 - c_3s_2)(c_1o_x + s_1o_y) + (-c_3c_2 + s_3s_2)o_z, \quad (262)$$

$$\zeta^* = (-s_3c_2 - c_3s_2)(c_1a_x + s_1a_y) + (-c_3c_2 + s_3s_2)a_z, \quad (263)$$

$$\tau^* = (-s_3c_2 - c_3s_2)(c_1x + s_1y) + (-c_3c_2 + s_3s_2)z + a_2s_3, \quad (264)$$

$$\upsilon^* = c_1n_y - s_1n_x, \quad (265)$$

$$\omega^* = c_1o_y - s_1o_x, \quad (266)$$

$$\xi^* = -s_1a_x + c_1a_y, \quad (267)$$

$$\psi^* = c_1y - s_1x - d_3, \quad (268)$$

sendo  ${}^3A_6(\theta_i)$  dado por:

$$\begin{aligned} {}^3A_6(\theta_i) &= {}^3A_4(\theta_4) \cdot {}^4A_5(\theta_5) \cdot {}^5A_6(\theta_6) \\ &= \begin{bmatrix} c_4 & -s_4 & 0 & a_3 \\ 0 & 0 & 1 & d_4 \\ -s_4 & -c_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} c_5 & -s_5 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_5 & c_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_6 & -c_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} c_4c_5c_6 - s_4s_6 & -c_4c_5s_6 - s_4c_6 & -c_4s_5 & a_3 \\ s_5c_6 & -s_5s_6 & c_5 & d_4 \\ -s_4c_5c_6 - c_4s_6 & s_4c_5s_6 - c_4c_6 & s_4s_5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \end{aligned} \quad (269)$$

Igualando tanto os elementos (linha 1, coluna 4) quanto os elementos (linha 2, coluna 4) dos dois membros da Equação (256), obtém-se:

$$c_1 c_{23} x + s_1 c_{23} y - s_{23} z - a_2 c_3 = a_3, \quad (270)$$

$$-c_1 s_{23} x - s_1 s_{23} y - c_{23} z + a_2 s_3 = d_4. \quad (271)$$

Essas duas equações podem ser resolvidas simultaneamente para  $s_{23}$  e  $c_{23}$ , resultando em:

$$s_{23} = \frac{c_1 s_3 a_2 x + s_1 s_3 a_2 y - c_1 d_4 x - s_1 d_4 y - a_2 c_3 z - a_3 z}{c_1^2 x^2 + 2c_1 s_1 xy + s_1^2 y^2 + z^2}, \quad (272)$$

$$c_{23} = \frac{c_1 c_3 a_2 x + s_1 c_3 a_2 y + c_1 a_3 x + s_1 a_3 y + a_2 s_3 z - d_4 z}{c_1^2 x^2 + 2c_1 s_1 xy + s_1^2 y^2 + z^2}. \quad (273)$$

Após algumas simplificações, tem-se:

$$s_{23} = \frac{(-a_3 - a_2 c_3)z - (c_1 x + s_1 y)(d_4 - a_2 s_3)}{z^2 + (c_1 x + s_1 y)^2}, \quad (274)$$

$$c_{23} = \frac{(a_2 s_3 - d_4)z - (a_3 + a_2 c_3)(c_1 x + s_1 y)}{z^2 + (c_1 x + s_1 y)^2}. \quad (275)$$

Observa-se que os denominadores das duas equações são iguais e positivos, o que nos permite obter facilmente uma expressão para determinação da soma de  $\theta_2$  e  $\theta_3$ , em que  $\theta_{23}$  denota a soma  $\theta_2 + \theta_3$ .

$$\theta_{23} = \text{Atan2} \left( \frac{(-a_3 - a_2 c_3)z - (c_1 x + s_1 y)(d_4 - a_2 s_3)}{(a_2 s_3 - d_4)z - (a_3 + a_2 c_3)(c_1 x + s_1 y)} \right) \quad (276)$$

A Equação (276) fornece quatro valores para  $\theta_{23}$  de acordo com as quatro combinações possíveis de soluções para  $\theta_1$  e  $\theta_3$ . Após determinação dessas quatro soluções para  $\theta_{23}$ , os valores de  $\theta_2$  são obtidos por  $\theta_2 = \theta_{23} - \theta_3$ , ou seja:

$$\theta_2 = \text{Atan2} \left( \frac{(-a_3 - a_2 c_3)z - (c_1 x + s_1 y)(d_4 - a_2 s_3)}{(a_2 s_3 - d_4)z - (a_3 + a_2 c_3)(c_1 x + s_1 y)} \right) - \theta_3. \quad (277)$$

Dessarte:

$$\theta_2 = \text{Atan2} \left( \begin{array}{l} (-a_3 - a_2 c_3)z - (c_1 x + s_1 y)(d_4 - a_2 s_3), \\ (a_2 s_3 - d_4)z - (a_3 + a_2 c_3)(c_1 x + s_1 y) \end{array} \right) - \left[ \begin{array}{l} \text{Atan2}(a_3, d_4) \\ \left( \frac{x^2 + y^2 + z^2 - a_2^2 - a_3^2 - d_3^2 - d_4^2}{2a_2} \right), \\ \pm \sqrt{a_3^2 + d_4^2 - \left( \frac{x^2 + y^2 + z^2 - a_2^2 - a_3^2 - d_3^2 - d_4^2}{2a_2} \right)^2} \end{array} \right]. \quad (278)$$

Igualando tanto os elementos (linha 1, coluna 3) quanto os elementos (linha 3, coluna 3) dos dois membros da Equação (256), tem-se:

$$a_x c_1 c_{23} + a_y s_1 c_{23} - a_z s_{23} = -c_4 s_5, \quad (279)$$

$$-a_x s_1 + a_y c_1 = s_4 s_5. \quad (280)$$

Considerando que  $s_5 \neq 0$ , pode-se determinar  $\theta_4$  como:

$$\theta_4 = \text{Atan2} \left( \begin{array}{l} (-a_x s_1 + a_y c_1), \\ (-a_x c_1 (c_2 c_3 - s_2 s_3) - a_y s_1 (c_2 c_3 - s_2 s_3) + a_z (c_2 s_3 + s_2 c_3)) \end{array} \right). \quad (281)$$

Quando  $s_5 = 0$ , o robô encontra-se em uma configuração singular, na qual os eixos das juntas 4 e 6 estão alinhados e produzem o mesmo movimento do último elo do manipulador. Nessa situação indesejada, apenas a soma ou diferença dos ângulos  $\theta_4$  e  $\theta_6$  podem ser calculados. Tal situação pode ser detectada ao verificar quando os dois argumentos do (Atan2) na Equação (281) estão próximos de zero.

A Equação (204) é reescrita de forma que todo o lado esquerdo seja uma função de apenas termos conhecidos e  $\theta_4$ .

$$\left[ {}^3A_4(\theta_4) \right]^{-1} \cdot \left[ {}^2A_3(\theta_3) \right]^{-1} \cdot \left[ {}^1A_2(\theta_2) \right]^{-1} \cdot \left[ {}^0A_1(\theta_1) \right]^{-1} \cdot T = {}^4A_5(\theta_5) \cdot {}^5A_6(\theta_6) \quad (282)$$

$$\begin{bmatrix} c_4 & 0 & -s_4 & -a_3c_4 \\ -s_4 & 0 & -c_4 & a_3s_4 \\ 0 & 1 & 0 & -d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} c_3 & s_3 & 0 & -a_2c_3 \\ -s_3 & c_3 & 0 & a_2s_3 \\ 0 & 0 & 1 & -d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} c_2 & 0 & -s_2 & 0 \\ -s_2 & 0 & -c_2 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (283)$$

$$\cdot \begin{bmatrix} c_1 & s_1 & 0 & 0 \\ -s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} n_x & o_x & a_x & x \\ n_y & o_y & a_y & y \\ n_z & o_z & a_z & z \\ 0 & 0 & 0 & 1 \end{bmatrix} = {}^4A_6(\theta_i)$$

$$\begin{bmatrix} \Phi & \Gamma & \Delta & \Theta \\ \Sigma & \Pi & \Upsilon & \Omega \\ \Xi & \Psi & Z & \Lambda \\ 0 & 0 & 0 & 1 \end{bmatrix} = {}^4A_6(\theta_i) \quad (284)$$

Em que as entradas de  ${}^4A_6(\theta_i)$  são dadas por:

$$\begin{aligned} \Phi &= ((c_4c_3c_2 - c_4s_3s_2)c_1 + s_4s_1)n_x \\ &+ ((c_4c_3c_2 - c_4s_3s_2)s_1 - s_4c_1)n_y + (-c_4s_3c_2 - c_4c_3s_2)n_z, \end{aligned} \quad (285)$$

$$\begin{aligned} \Gamma &= ((c_4c_3c_2 - c_4s_3s_2)c_1 + s_4s_1)o_x \\ &+ ((c_4c_3c_2 - c_4s_3s_2)s_1 - s_4c_1)o_y + (-c_4s_3c_2 - c_4c_3s_2)o_z, \end{aligned} \quad (286)$$

$$\begin{aligned} \Delta &= ((c_4c_3c_2 - c_4s_3s_2)c_1 + s_4s_1)a_x \\ &+ ((c_4c_3c_2 - c_4s_3s_2)s_1 - s_4c_1)a_y + (-c_4s_3c_2 - c_4c_3s_2)a_z, \end{aligned} \quad (287)$$

$$\begin{aligned} \Theta &= ((c_4c_3c_2 - c_4s_3s_2)c_1 + s_4s_1)x + ((c_4c_3c_2 - c_4s_3s_2)s_1 - s_4c_1)y \\ &+ (-c_4s_3c_2 - c_4c_3s_2)z - a_2c_4c_3 - a_3c_4 + d_3s_4, \end{aligned} \quad (288)$$

$$\begin{aligned} \Sigma &= ((-s_4c_3c_2 + s_4s_3s_2)c_1 + c_4s_1)n_x + ((-s_4c_3c_2 + s_4s_3s_2)s_1 - c_4c_1)n_y \\ &+ (s_4s_3c_2 + s_4c_3s_2)n_z, \end{aligned} \quad (289)$$

$$\begin{aligned} \Pi &= ((-s_4c_3c_2 + s_4s_3s_2)c_1 + c_4s_1)o_x + ((-s_4c_3c_2 + s_4s_3s_2)s_1 - c_4c_1)o_y \\ &+ (s_4s_3c_2 + s_4c_3s_2)o_z, \end{aligned} \quad (290)$$

$$\begin{aligned} \Upsilon &= ((-s_4c_3c_2 + s_4s_3s_2)c_1 + c_4s_1)a_x + ((-s_4c_3c_2 + s_4s_3s_2)s_1 - c_4c_1)a_y \\ &+ (s_4s_3c_2 + s_4c_3s_2)a_z, \end{aligned} \quad (291)$$

$$\begin{aligned} \Omega &= ((-s_4c_3c_2 + s_4s_3s_2)c_1 + c_4s_1)x + ((-s_4c_3c_2 + s_4s_3s_2)s_1 - c_4c_1)y \\ &+ (s_4s_3c_2 + s_4c_3s_2)z + a_2s_4c_3 + a_3s_4 + d_3c_4, \end{aligned} \quad (292)$$

$$\Xi = (-s_3c_2 - c_3s_2)c_1n_x + (-s_3c_2 - c_3s_2)s_1n_y + (-c_3c_2 + s_3s_2)n_z, \quad (293)$$

$$\Psi = (-s_3c_2 - c_3s_2)c_1o_x + (-s_3c_2 - c_3s_2)s_1o_y + (-c_3c_2 + s_3s_2)o_z, \quad (294)$$

$$Z = (-s_3c_2 - c_3s_2)c_1a_x + (-s_3c_2 - c_3s_2)s_1a_y + (-c_3c_2 + s_3s_2)a_z, \quad (295)$$

$$\Lambda = (-s_3c_2 - c_3s_2)c_1x + (-s_3c_2 - c_3s_2)s_1y + (-c_3c_2 + s_3s_2)z + a_2s_3 - d_4, \quad (296)$$

sendo  ${}^4A_6(\theta_i)$  dada por:

$$\begin{aligned} {}^4A_6(\theta_i) &= {}^4A_5(\theta_5) \cdot {}^5A_6(\theta_6) \\ &= \begin{bmatrix} c_5 & -s_5 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_5 & c_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_6 & -c_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_5c_6 & -c_5s_6 & -s_5 & 0 \\ s_6 & c_6 & 0 & 0 \\ s_5c_6 & -s_5s_6 & c_5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \end{aligned} \quad (297)$$

Igualando tanto os elementos (linha 1, coluna 3) quanto os elementos (linha 3, coluna 3) dos dois membros da Equação (284), tem-se:

$$\begin{pmatrix} a_x (c_1 (c_2c_3 - s_2s_3)c_4 + s_1s_4) \\ +a_y (s_1 (c_2c_3 - s_2s_3)c_4 - c_1s_4) \\ -a_z ((c_2s_3 + s_2c_3)c_4) \end{pmatrix} = -s_5, \quad (298)$$

$$\begin{pmatrix} a_x (-c_1 (c_2s_3 + s_2c_3)) \\ +a_y (-s_1 (c_2s_3 + s_2c_3)) \\ +a_z (-(c_2c_3 - s_2s_3)) \end{pmatrix} = c_5. \quad (299)$$

Desse modo, pode-se determinar  $\theta_5$  como:

$$\theta_5 = \text{Atan2}(s_5, c_5) = \text{Atan2} \left( \begin{array}{c} - \begin{pmatrix} a_x (c_1 (c_2c_3 - s_2s_3)c_4 + s_1s_4) \\ +a_y (s_1 (c_2c_3 - s_2s_3)c_4 - c_1s_4) \\ -a_z ((c_2s_3 + s_2c_3)c_4) \end{pmatrix} \\ \begin{pmatrix} a_x (-c_1 (c_2s_3 + s_2c_3)) \\ +a_y (-s_1 (c_2s_3 + s_2c_3)) \\ +a_z (-(c_2c_3 - s_2s_3)) \end{pmatrix} \end{array} \right). \quad (300)$$

A Equação (204) é reescrita de forma que todo o lado esquerdo seja uma função de apenas termos conhecidos e  $\theta_5$ .

$$[{}^4A_5(\theta_5)]^{-1} \cdot [{}^3A_4(\theta_4)]^{-1} \cdot [{}^2A_3(\theta_3)]^{-1} \cdot [{}^1A_2(\theta_2)]^{-1} \cdot [{}^0A_1(\theta_1)]^{-1} \cdot T = {}^5A_6(\theta_6) \quad (301)$$

$$\begin{aligned}
& \begin{bmatrix} c_5 & 0 & s_5 & 0 \\ -s_5 & 0 & c_5 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} c_4 & 0 & -s_4 & -a_3c_4 \\ -s_4 & 0 & -c_4 & a_3s_4 \\ 0 & 1 & 0 & -d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} c_3 & s_3 & 0 & -a_2c_3 \\ -s_3 & c_3 & 0 & a_2s_3 \\ 0 & 0 & 1 & -d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
& \cdot \begin{bmatrix} c_2 & 0 & -s_2 & 0 \\ -s_2 & 0 & -c_2 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} c_1 & s_1 & 0 & 0 \\ -s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} n_x & o_x & a_x & x \\ n_y & o_y & a_y & y \\ n_z & o_z & a_z & z \\ 0 & 0 & 0 & 1 \end{bmatrix} = {}^5A_6(\theta_6)
\end{aligned} \tag{302}$$

$$\begin{bmatrix} \Phi^* & \Gamma^* & \Delta^* & \Theta^* \\ \Sigma^* & \Pi^* & \Upsilon^* & \Omega^* \\ \Xi^* & \Psi^* & Z^* & \Lambda^* \\ 0 & 0 & 0 & 1 \end{bmatrix} = {}^5A_6(\theta_6) \tag{303}$$

Em que as entradas de  ${}^5A_6(\theta_i)$  são dadas por:

$$\begin{aligned}
\Phi^* &= \left( (c_5c_4c_3 - s_5s_3)c_2 - (c_5c_4s_3 + s_5c_3)s_2 \right) c_1 + c_5s_4s_1 n_x \\
&+ \left( (c_5c_4c_3 - s_5s_3)c_2 - (c_5c_4s_3 + s_5c_3)s_2 \right) s_1 - c_5s_4c_1 n_y \\
&+ \left( -(c_5c_4c_3 - s_5s_3)s_2 - (c_5c_4s_3 + s_5c_3)c_2 \right) n_z,
\end{aligned} \tag{304}$$

$$\begin{aligned}
\Gamma^* &= \left( (c_5c_4c_3 - s_5s_3)c_2 - (c_5c_4s_3 + s_5c_3)s_2 \right) c_1 + c_5s_4s_1 o_x \\
&+ \left( (c_5c_4c_3 - s_5s_3)c_2 - (c_5c_4s_3 + s_5c_3)s_2 \right) s_1 - c_5s_4c_1 o_y \\
&+ \left( -(c_5c_4c_3 - s_5s_3)s_2 - (c_5c_4s_3 + s_5c_3)c_2 \right) o_z,
\end{aligned} \tag{305}$$

$$\begin{aligned}
\Delta^* &= \left( (c_5c_4c_3 - s_5s_3)c_2 - (c_5c_4s_3 + s_5c_3)s_2 \right) c_1 + c_5s_4s_1 a_x \\
&+ \left( (c_5c_4c_3 - s_5s_3)c_2 - (c_5c_4s_3 + s_5c_3)s_2 \right) s_1 - c_5s_4c_1 a_y \\
&+ \left( -(c_5c_4c_3 - s_5s_3)s_2 - (c_5c_4s_3 + s_5c_3)c_2 \right) a_z,
\end{aligned} \tag{306}$$

$$\begin{aligned}
\Theta^* &= \left( (c_5c_4c_3 - s_5s_3)c_2 - (c_5c_4s_3 + s_5c_3)s_2 \right) c_1 + c_5s_4s_1 x \\
&+ \left( (c_5c_4c_3 - s_5s_3)c_2 - (c_5c_4s_3 + s_5c_3)s_2 \right) s_1 - c_5s_4c_1 y \\
&+ \left( -(c_5c_4c_3 - s_5s_3)s_2 - (c_5c_4s_3 + s_5c_3)c_2 \right) z \\
&- a_2c_5c_4c_3 + a_2s_5s_3 + a_3c_5c_4 + d_3c_5s_4 - d_4s_5,
\end{aligned} \tag{307}$$

$$\begin{aligned}
\Sigma^* &= \left( (-s_5c_4c_3 - c_5s_3)c_2 - (-s_5c_4s_3 + c_5c_3)s_2 \right) c_1 - s_5s_4s_1 n_x \\
&+ \left( (-s_5c_4c_3 - c_5s_3)c_2 - (-s_5c_4s_3 + c_5c_3)s_2 \right) s_1 + s_5s_4c_1 n_y \\
&+ \left( -(-s_5c_4c_3 - c_5s_3)s_2 - (-s_5c_4s_3 + c_5c_3)c_2 \right) n_z,
\end{aligned} \tag{308}$$



$$\begin{aligned}\Pi^* &= \left( \left( (-s_5 c_4 c_3 - c_5 s_3) c_2 - (-s_5 c_4 s_3 + c_5 c_3) s_2 \right) c_1 - s_5 s_4 s_1 \right) o_x \\ &+ \left( \left( (-s_5 c_4 c_3 - c_5 s_3) c_2 - (-s_5 c_4 s_3 + c_5 c_3) s_2 \right) s_1 + s_5 s_4 c_1 \right) o_y \\ &+ \left( -(-s_5 c_4 c_3 - c_5 s_3) s_2 - (-s_5 c_4 s_3 + c_5 c_3) c_2 \right) o_z,\end{aligned}\quad (309)$$

$$\begin{aligned}\Upsilon^* &= \left( \left( (-s_5 c_4 c_3 - c_5 s_3) c_2 - (-s_5 c_4 s_3 + c_5 c_3) s_2 \right) c_1 - s_5 s_4 s_1 \right) a_x \\ &+ \left( \left( (-s_5 c_4 c_3 - c_5 s_3) c_2 - (-s_5 c_4 s_3 + c_5 c_3) s_2 \right) s_1 + s_5 s_4 c_1 \right) a_y \\ &+ \left( -(-s_5 c_4 c_3 - c_5 s_3) s_2 - (-s_5 c_4 s_3 + c_5 c_3) c_2 \right) a_z,\end{aligned}\quad (310)$$

$$\begin{aligned}\Omega^* &= \left( \left( (-s_5 c_4 c_3 - c_5 s_3) c_2 - (-s_5 c_4 s_3 + c_5 c_3) s_2 \right) c_1 - s_5 s_4 s_1 \right) x \\ &+ \left( \left( (-s_5 c_4 c_3 - c_5 s_3) c_2 - (-s_5 c_4 s_3 + c_5 c_3) s_2 \right) s_1 + s_5 s_4 c_1 \right) y \\ &+ \left( -(-s_5 c_4 c_3 - c_5 s_3) s_2 - (-s_5 c_4 s_3 + c_5 c_3) c_2 \right) z \\ &+ a_2 s_5 c_4 c_3 + a_2 c_5 s_3 + a_3 s_5 c_4 - d_3 s_5 s_4 - d_4 c_5,\end{aligned}\quad (311)$$

$$\begin{aligned}\Xi^* &= \left( (s_4 c_3 c_2 - s_4 s_3 s_2) c_1 - c_4 s_1 \right) n_x \\ &+ \left( (s_4 c_3 c_2 - s_4 s_3 s_2) s_1 + c_4 c_1 \right) n_y + \left( -s_4 s_3 c_2 - s_4 c_3 s_2 \right) n_z,\end{aligned}\quad (312)$$

$$\begin{aligned}\Psi^* &= \left( (s_4 c_3 c_2 - s_4 s_3 s_2) c_1 - c_4 s_1 \right) o_x \\ &+ \left( (s_4 c_3 c_2 - s_4 s_3 s_2) s_1 + c_4 c_1 \right) o_y + \left( -s_4 s_3 c_2 - s_4 c_3 s_2 \right) o_z,\end{aligned}\quad (313)$$

$$\begin{aligned}\mathbf{Z}^* &= \left( (s_4 c_3 c_2 - s_4 s_3 s_2) c_1 - c_4 s_1 \right) a_x \\ &+ \left( (s_4 c_3 c_2 - s_4 s_3 s_2) s_1 + c_4 c_1 \right) a_y + \left( -s_4 s_3 c_2 - s_4 c_3 s_2 \right) a_z,\end{aligned}\quad (314)$$

$$\begin{aligned}\Lambda^* &= \left( (s_4 c_3 c_2 - s_4 s_3 s_2) c_1 - c_4 s_1 \right) x + \left( (s_4 c_3 c_2 - s_4 s_3 s_2) s_1 + c_4 c_1 \right) y \\ &+ \left( -s_4 s_3 c_2 - s_4 c_3 s_2 \right) z - a_2 s_4 c_3 - a_3 s_4 - d_3 c_4,\end{aligned}\quad (315)$$

sendo  ${}^5A_6(\theta_6)$  dada por:

$${}^5A_6(\theta_6) = \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_6 & -c_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (316)$$

Igualando tanto os elementos (linha 3, coluna 1) quanto os elementos (linha 1, coluna 1) dos dois membros da Equação (303), tem-se:

$$s_6 = \begin{pmatrix} -n_x \left( c_1 (c_2 c_3 - s_2 s_3) s_4 - s_1 c_4 \right) - n_y \left( s_1 (c_2 c_3 - s_2 s_3) s_4 + c_1 c_4 \right) \\ + n_z \left( (c_2 s_3 + s_2 c_3) s_4 \right) \end{pmatrix}, \quad (317)$$

$$c_6 = \begin{pmatrix} n_x \left( (c_1 (c_2 c_3 - s_2 s_3) c_4 + s_1 s_4) c_5 - c_1 (c_2 s_3 + s_2 c_3) s_5 \right) \\ + n_y \left( \left( (s_1 (c_2 c_3 - s_2 s_3) c_4) - c_1 s_4 \right) c_5 - s_1 (c_2 s_3 + s_2 c_3) s_5 \right) \\ - n_z \left( c_4 (c_2 s_3 + s_2 c_3) c_5 + (c_2 c_3 - s_2 s_3) s_5 \right) \end{pmatrix}. \quad (318)$$

Dessa forma, pode-se determinar  $\theta_6$  como:

$$\theta_6 = \text{Atan2} \left( \begin{pmatrix} -n_x (c_1 (c_2 c_3 - s_2 s_3) s_4 - s_1 c_4) - n_y (s_1 (c_2 c_3 - s_2 s_3) s_4 + c_1 c_4) \\ + n_z ((c_2 s_3 + s_2 c_3) s_4) \end{pmatrix}, \begin{pmatrix} n_x ((c_1 (c_2 c_3 - s_2 s_3) c_4 + s_1 s_4) c_5 - c_1 (c_2 s_3 + s_2 c_3) s_5) \\ + n_y \left( \left( (s_1 (c_2 c_3 - s_2 s_3) c_4) - c_1 s_4 \right) c_5 - s_1 (c_2 s_3 + s_2 c_3) s_5 \right) \\ - n_z (c_4 (c_2 s_3 + s_2 c_3) c_5 + (c_2 c_3 - s_2 s_3) s_5) \end{pmatrix} \right). \quad (319)$$

Por conta dos sinais de mais ou menos que aparecem na Equação (229) e (253), elas fornecem quatro soluções. Porém, ainda existem mais quatro soluções obtidas quando o punho do manipulador for rotacionado em 180 graus; ou seja, para cada uma das quatro soluções inicialmente encontradas tem-se:

$$\theta_4' = \theta_4 + 180^\circ, \quad (320)$$

$$\theta_5' = -\theta_5, \quad (321)$$

$$\theta_6' = \theta_6 + 180^\circ. \quad (322)$$

Resumidamente, a solução da cinemática inversa deste manipulador robótico é dada por (323) a (331).

$$\theta_1 = \text{Atan2}(y, x) - \text{Atan2} \left( d_3, \pm \sqrt{x^2 + y^2 - d_3^2} \right) \quad (323)$$

$$\theta_3 = \text{Atan2}(a_3, d_4) - \text{Atan2} \left( \begin{pmatrix} \left( \frac{x^2 + y^2 + z^2 - a_2^2 - a_3^2 - d_3^2 - d_4^2}{2a_2} \right), \\ \pm \sqrt{a_3^2 + d_4^2 - \left( \frac{x^2 + y^2 + z^2 - a_2^2 - a_3^2 - d_3^2 - d_4^2}{2a_2} \right)^2} \end{pmatrix} \right) \quad (324)$$

$$\theta_2 = \text{Atan 2} \left( \begin{array}{l} (-a_3 - a_2 c_3)z - (c_1 x + s_1 y)(d_4 - a_2 s_3), \\ (a_2 s_3 - d_4)z - (a_3 + a_2 c_3)(c_1 x + s_1 y) \end{array} \right) - \left( \begin{array}{l} \text{Atan 2}(a_3, d_4) - \text{Atan 2} \left( \begin{array}{l} \left( \frac{x^2 + y^2 + z^2 - a_2^2 - a_3^2 - d_3^2 - d_4^2}{2a_2} \right), \\ \pm \sqrt{\frac{a_3^2 + d_4^2}{\left( \frac{x^2 + y^2 + z^2 - a_2^2 - a_3^2 - d_3^2 - d_4^2}{2a_2} \right)^2}} \end{array} \right) \end{array} \right) \quad (325)$$

$$\theta_4 = \text{Atan 2} \left( (-a_x s_1 + a_y c_1), (-a_x c_1 c_{23} - a_y s_1 c_{23} + a_z s_{23}) \right) \quad (326)$$

$$\theta_5 = \text{Atan 2} \left( \begin{array}{l} - \left( \begin{array}{l} a_x (c_1 c_{23} c_4 + s_1 s_4) \\ + a_y (s_1 c_{23} c_4 - c_1 s_4) - a_z (s_{23} c_4) \end{array} \right), \\ (a_x (-c_1 s_{23}) + a_y (-s_1 s_{23}) + a_z (-c_{23})) \end{array} \right) \quad (327)$$

$$\theta_6 = \text{Atan 2} \left( \begin{array}{l} \left( \begin{array}{l} -n_x (c_1 c_{23} s_4 - s_1 c_4) - n_y (s_1 c_{23} s_4 + c_1 c_4) \\ + n_z (s_{23} s_4) \end{array} \right), \\ \left( \begin{array}{l} n_x ((c_1 c_{23} c_4 + s_1 s_4) c_5 - c_1 s_{23} s_5) \\ + n_y ((s_1 c_{23} c_4 - c_1 s_4) c_5) - s_1 s_{23} s_5 \\ - n_z (c_4 s_{23} c_5 + s_1 c_{23} c_4 s_5) \end{array} \right) \end{array} \right) \quad (328)$$

$$\theta_4' = \theta_4 + 180^\circ \quad (329)$$

$$\theta_5' = -\theta_5 \quad (330)$$

$$\theta_6' = \theta_6 + 180^\circ \quad (331)$$

Observa-se que as Equações (326), (327) e (328) foram obtidas com base nas Equações (281), (300) e (319), substituindo  $(c_2 c_3 - s_2 s_3)$  e  $(c_2 s_3 + s_2 c_3)$  por  $c_{23}$  e  $s_{23}$ , respectivamente. Essas substituições foram realizadas por conta da necessidade de se obter equações computacionalmente menos onerosas, ou seja, com o menor número possível de operações de ponto flutuante.

O número de operações de ponto flutuante aumenta à medida que o número de soluções cinemáticas se amplia. Em geral, a cinemática inversa gerada contém várias soluções. Conhecer todas as soluções é útil para otimizar e verificar um processo de

planejamento de trajetórias. A ordem dos cálculos para determinar todas as oito configurações do Unimation PUMA 560 é representada por intermédio de uma estrutura em árvore na Figura 27.

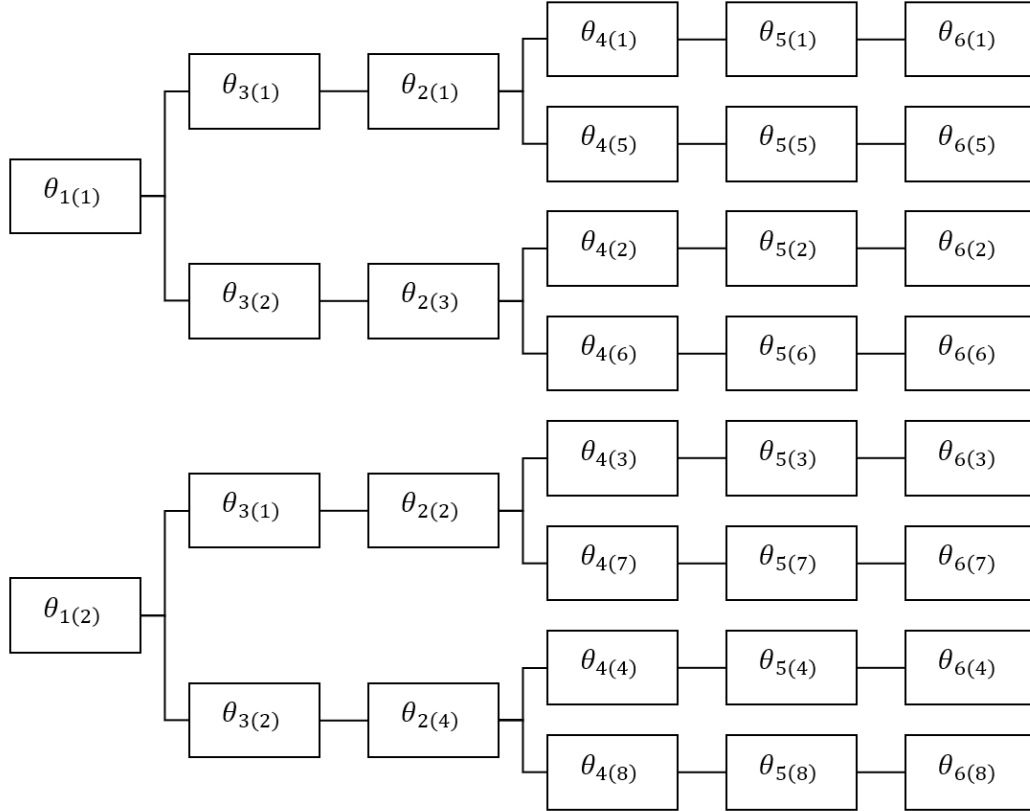


Figura 27: Estrutura em árvore das oito configurações cinemáticas inversas do PUMA 560

Com o objetivo de mensurar o custo computacional para determinar todas as configurações de junta para posicionar e orientar o efetuador no espaço, os seguintes valores de junta:  $\theta = [0^\circ, 0^\circ, 0^\circ, 0^\circ, 0^\circ, 0^\circ]$ , escolhidos arbitrariamente, e demais parâmetros não nulos:  $a_2 = 432$ ,  $a_3 = 20$ ,  $d_3 = 125$  e  $d_4 = 430$ , foram substituídos em (190). Assim, obtém-se (332).

$$T = \begin{bmatrix} 1 & 0 & 0 & 452 \\ 0 & -1 & 0 & 125 \\ 0 & 0 & -1 & -430 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (332)$$

A partir deste ponto, as Equações (192) a (203) são reescritas substituindo  $n_x$ ,  $n_y$ ,  $n_z$ ,  $o_x$ ,  $o_y$ ,  $o_z$ ,  $a_x$ ,  $a_y$ ,  $a_z$ ,  $x$ ,  $y$  e  $z$  pelos respectivos valores da matriz numérica obtida.

Utiliza-se o MAPLE para determinar todos os valores de  $\theta = [\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6]$ , que permitam ao robô assumir a configuração (posição + orientação) dada por (332). Todos esses resultados foram obtidos por meio do código em MAPLE apresentado no Apêndice D deste trabalho.

A Tabela 6 ilustra o detalhamento de todas as operações matemáticas realizadas para solução deste problema, ângulo a ângulo.

Tabela 6: Número de operações para solução da cinemática inversa do PUMA 560, ângulo a ângulo, por meio do método de Paul

Ângulo	Adição	Multiplicação	Divisão	Função
$\theta_{1(1)}$	0	0	0	0
$\theta_{1(2)}$	1	1	0	1
$\theta_{2(1)}$	77	134	5	105
$\theta_{2(2)}$	76	134	5	105
$\theta_{2(3)}$	78	135	5	106
$\theta_{2(4)}$	78	135	5	106
$\theta_{3(1)}$	0	0	0	0
$\theta_{3(2)}$	1	1	0	1
$\theta_{4(1)}$	0	0	0	0
$\theta_{4(2)}$	0	0	0	0
$\theta_{4(3)}$	0	0	0	0
$\theta_{4(4)}$	0	0	0	0
$\theta_{4(5)}$	0	1	0	0

Ângulo	Adição	Multiplicação	Divisão	Função
$\theta_{4(6)}$	0	1	0	0
$\theta_{4(7)}$	0	0	0	0
$\theta_{4(8)}$	0	1	0	0
$\theta_{5(1)}$	0	1	0	0
$\theta_{5(2)}$	80	142	5	113
$\theta_{5(3)}$	81	142	5	113
$\theta_{5(4)}$	80	142	5	113
$\theta_{5(5)}$	0	1	0	0
$\theta_{5(6)}$	81	142	5	113
$\theta_{5(7)}$	80	142	5	113
$\theta_{5(8)}$	81	142	5	113
$\theta_{6(1)}$	8	12	1	12
$\theta_{6(2)}$	264	469	21	376
$\theta_{6(3)}$	265	469	21	376
$\theta_{6(4)}$	264	469	21	376
$\theta_{6(5)}$	8	13	1	12
$\theta_{6(6)}$	265	469	21	376
$\theta_{6(7)}$	265	469	21	376
$\theta_{6(8)}$	265	469	21	376

De acordo com os dados de desempenho computacional mensurados por meio do MAPLE, são necessárias 34402 operações de ponto flutuante (2398 adições, 4236 multiplicações, 178 divisões e 3382 funções) para a solução do problema cinemático inverso deste manipulador (determinação das oito configurações de juntas) utilizando as equações deduzidas nesta seção com base no método de Paul.

Pode-se observar na Tabela 7 as oito configurações de junta distintas, número máximo de configurações para esse manipulador.

Tabela 7: Configurações de juntas do Puma 560 (em radianos) obtidas com base no método de Paul

$\theta_1$ (radianos)	$\theta_2$ (radianos)	$\theta_3$ (radianos)	$\theta_4$ (radianos)	$\theta_5$ (radianos)	$\theta_6$ (radianos)
-2,6020	3,1416	-3,0486	3,1416	0,092956	0,53961
-2,6020	3,1416	-3,0486	0	-0,092956	-2,6020
0	0	0	0	0	0
-2,6020	1,6207	0	0	-1,6207	-2,6020
-2,6020	1,6207	0	3,1416	1,6207	0,53961
0	1,5209	-3,0486	0	1,5277	0
0	1,5209	-3,0486	3,1416	-1,5277	3,1416
0	0	0	3,1416	0	3,1416

Foi utilizado o método de Paul (1981) como um segundo algoritmo para resolver o problema da cinemática inversa dos manipuladores robóticos estudados, porém outros métodos, computacionalmente mais caros, poderiam ser utilizados, por exemplo: vetores *screw*, simbólico, análise da posição zero, Newton-Raphson, pseudoinversa, integração das velocidades, entre outros.

Existem vários critérios de decisão possíveis para escolher uma entre todas as soluções encontradas, por exemplo, a mais próxima da configuração atual, o caminho mais curto para o alvo e a minimização de energia e tempo. A área de movimento articular mecanicamente limitada também é considerada. Além disso, existe a possibilidade de zero soluções, é o caso quando o ponto que deve ser atingido encontra-se fora do volume de trabalho ou dentro do volume de trabalho, mas deve ser atingido em uma orientação impossível.

Como este trabalho visa comprovar que, ao utilizar a Teoria de Bases de Gröbner, o custo computacional para determinar a solução da cinemática inversa de manipuladores seriais antropomórficos é reduzido consideravelmente, a cinemática inversa do manipulador antropomórfico PUMA 560 será solucionada por meio da Teoria de Bases de Gröbner, no Capítulo 7.



## 7 CINEMÁTICA INVERSA DO PUMA 560 POR MEIO DA TEORIA DE BASES DE GRÖBNER

Devido à configuração geométrica desta família de manipuladores, não foi possível representar os movimentos das seis variáveis de junta no sistema  $xyz$  utilizando o ponto  $(x; y; z)$  para representar o efetuador final no espaço tridimensional. Dessa forma, a modelagem do problema não pode ser feita com equações polinomiais simples que descrevam o comportamento dos ângulos de junta, como foi possível para o manipulador do tipo SCARA na Seção 5.1.

Para o manipulador PUMA 560, pretende-se determinar todas as configurações de junta para posicionar e orientar o efetuador do manipulador robótico no espaço, essência da cinemática inversa.

Guzmán-Giménez, *et al.* (2020), apresentaram um procedimento sistemático para solução do problema da cinemática inversa de sistemas robóticos de cadeia aberta não redundantes utilizando a Teoria de Bases de Gröbner, após solução da cinemática direta com base no algoritmo de Denavit-Hartenberg. Os autores também utilizaram como exemplo o robô antropomórfico Unimation PUMA 560, porém eles se restringiram às três primeiras juntas do manipulador robótico, desconsiderando as três juntas responsáveis pela orientação do efetuador do robô antropomórfico.

Quando a análise cinemática é limitada às juntas  $\theta_1$ ,  $\theta_2$  e  $\theta_3$ , é possível determinar os valores dessas variáveis para atingir um ponto  $(x; y; z)$ , porém sabe-se que é extremamente necessário determinar os valores das juntas  $\theta_4$ ,  $\theta_5$  e  $\theta_6$ , responsáveis pela orientação do efetuador no espaço tridimensional. A determinação de todas as variáveis de juntas de um manipulador antropomórfico é fundamental em

diversas aplicações da Robótica Industrial, por exemplo: soldagem, pintura, movimentação de cargas, montagem, entre outros.

Com o objetivo de analisar a eficiência computacional para solução do problema da cinemática inversa de robôs antropomórficos utilizando a Teoria de Bases de Gröbner, dois casos serão estudados: o 1º caso utilizará a mesma matriz de transformação homogênea (332) usada na Seção 6.2 para determinação das oito configurações de juntas pelo método de Paul; no 2º caso, uma matriz de transformação homogênea obtida por um novo conjunto de variáveis de juntas será utilizada com o objetivo de confirmar a eficiência computacional para a solução da cinemática inversa ao utilizar a Teoria de Bases de Gröbner.

### 7.1 CINEMÁTICA INVERSA DO PUMA 560 (1º CASO)

Para atender a este objetivo, são substituídos arbitrariamente os seguintes valores de juntas:  $\theta = [0^\circ, 0^\circ, 0^\circ, 0^\circ, 0^\circ, 0^\circ]$ , e demais parâmetros não nulos:  $a_2 = 432$ ,  $a_3 = 20$ ,  $d_3 = 125$  e  $d_4 = 430$  em (190), exatamente conforme realizado na Seção 6.2.

A partir deste ponto, as equações (192) a (203) são reescritas substituindo  $n_x$ ,  $n_y$ ,  $n_z$ ,  $o_x$ ,  $o_y$ ,  $o_z$ ,  $a_x$ ,  $a_y$ ,  $a_z$ ,  $x$ ,  $y$  e  $z$  pelos respectivos valores da matriz numérica obtida, exatamente igual à matriz (332). Todos esses 12 polinômios, adicionados aos 6 polinômios obtidos por meio da relação trigonométrica fundamental para cada variável de junta, são utilizados para gerar uma base de Gröbner utilizando a ordem lexicográfica  $c_6 \succ s_6 \succ c_5 \succ s_5 \succ c_4 \succ s_4 \succ c_2 \succ s_2 \succ c_3 \succ s_3 \succ c_1 \succ s_1$ , seguindo a ordem para determinação das variáveis de juntas determinada pelo método de Paul. O código em MAPLE para determinação da base, com suas respectivas saídas, pode ser visualizado no Apêndice E deste trabalho.

Foram utilizados os mesmos valores atribuídos arbitrariamente no Capítulo 6, e todos os algoritmos foram executados no mesmo *software* de Computação Algébrica, com o objetivo de comparar a eficiência computacional para solução do problema da cinemática inversa utilizando o método de Paul e a Teoria de Bases de Gröbner.

Com a realização de 65 operações de ponto flutuante (25 adições e 40 multiplicações), a seguinte base de Gröbner contendo 13 polinômios é obtida:

$$g_1 = 219929s_1^2 + 113000s_1, \quad (333)$$

$$g_2 = 125c_1 - 452s_1 - 125, \quad (334)$$

$$g_3 = 1853s_3^2 + 172s_3, \quad (335)$$

$$g_4 = 2c_3 - 43s_3 - 2, \quad (336)$$

$$g_5 = 407528437s_1s_3 + 18913894s_1 + 9730100s_2 + 104694500s_3, \quad (337)$$

$$g_6 = 24325250c_2 - 248996875s_3 - 49703954s_1 - 24325250, \quad (338)$$

$$g_7 = s_1s_4, \quad (339)$$

$$g_8 = s_4s_3, \quad (340)$$

$$g_9 = c_4^2 + s_4^2 - 1, \quad (341)$$

$$g_{10} = 47504664c_4s_1s_3 - 47284735c_4s_1 + 261819000c_4s_3 + 24325250s_5, \quad (342)$$

$$g_{11} = -510675138s_1s_3 + 12162625c_5 - 24851977s_1 - 125388000s_3 - 12162625, \quad (343)$$

$$g_{12} = -c_4s_1 + s_4 + s_6, \quad (344)$$

$$g_{13} = -452c_4s_1 - 125c_4 + 125c_6. \quad (345)$$

Wang, Hang e Yang (2006) também apresentaram uma solução para a cinemática inversa de um manipulador serial 6R utilizando a Teoria de Bases de Gröbner, produzindo uma base de Gröbner com 72 polinômios por meio do método de solução da cinemática de mecanismos proposta por Duffy e Crane (1980). Como o método seguido pelos autores não é comparado com nenhum outro, a sua eficiência computacional não pôde ser confirmada.

Para encontrar todos os valores de  $\theta = [\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6]$ , que permitam ao robô assumir a configuração dada por (332), utiliza-se o comando Solve do MAPLE.

Em um total de 409 operações de ponto flutuante (25 adições, 40 multiplicações e 43 funções), obtêm-se oito configurações de junta distintas, exatamente iguais às configurações determinadas na Seção 6.2, conforme pode-se observar na Tabela 8.

Tabela 8: Configurações de juntas (em radianos) do Puma 560 obtidas por meio da Teoria de Bases de Gröbner

$\theta_1$ (radianos)	$\theta_2$ (radianos)	$\theta_3$ (radianos)	$\theta_4$ (radianos)	$\theta_5$ (radianos)	$\theta_6$ (radianos)
-2,6020	3,1416	-3,0486	3,1416	0,092956	0,53961
-2,6020	3,1416	-3,0486	0	-0,092956	-2,6020
0	0	0	0	0	0
-2,6020	1,6207	0	0	-1,6207	-2,6020
-2,6020	1,6207	0	3,1416	1,6207	0,53961
0	1,5209	-3,0486	0	1,5277	0
0	1,5209	-3,0486	3,1416	-1,5277	3,1416
0	0	0	3,1416	0	3,1416

O código em MAPLE para determinação das oito soluções, com suas respectivas saídas, encontra-se no Apêndice F deste trabalho.

Todas as soluções encontradas no Capítulo 6, utilizando o método de Paul para solução da cinemática inversa do manipulador PUMA 560, foram determinadas pelo MAPLE realizando 34402 operações de ponto flutuante.

Com o objetivo de validar os resultados obtidos, pode-se observar que, na Tabela 9, onde todos os ângulos da Tabela 8 foram convertidos em graus, a linha em destaque contém os valores das juntas  $\theta = [0^\circ, 0^\circ, 0^\circ, 0^\circ, 0^\circ, 0^\circ]$  utilizados para construir a matriz de transformação  $T$  utilizada neste caso.

Tabela 9: Configurações de juntas (em graus) do Puma 560 obtidas por meio da Teoria de Bases de Gröbner

$\theta_1$ (graus)	$\theta_2$ (graus)	$\theta_3$ (graus)	$\theta_4$ (graus)	$\theta_5$ (graus)	$\theta_6$ (graus)
-149,084	180,000	-174,672	180,000	5,325	30,917
-149,084	180,000	-174,672	0	-5,325	-149,084
0	0	0	0	0	0
-149,084	92,859	0	0	-92,859	-149,084
-149,084	92,859	0	180,000	92,859	30,917
0	87,141	-174,672	0	87,530	0
0	87,141	-174,672	180,000	-87,530	180,000
0	0	0	180,000	0	180,000

Adicionando as 65 operações de ponto flutuante para obtenção da base de Gröbner ao número de flop para obtenção das oito configurações de junta distintas por meio da base de Gröbner gerada (409 flop); obtém-se 474 operações de ponto flutuante. Pode-se observar que, para este conjunto de configurações, houve uma redução de 34402 para 474 do número de operações de ponto flutuante realizadas para a solução do problema cinemático inverso.

Pode-se observar na Tabela 10 o detalhamento do número de todas as operações matemáticas realizadas e do número de operações de ponto flutuante necessários para solução da cinemática inversa deste manipulador por meio dos dois métodos analisados.

Tabela 10: Dados de desempenho computacional para solução da cinemática inversa do manipulador Puma 560 por meio dos dois métodos

Método		Operações Matemáticas		Operações de Ponto Flutuante
Paul		Adição	2398	34402 flop
		Multiplicação	4236	
		Divisão	178	
		Função	3382	
Teoria de Bases de Gröbner	Geração da Base	Adição	25	65 flop
		Multiplicação	40	
		Divisão	0	
		Função	0	
	Determinação das Soluções	Adição	25	409 flop
		Multiplicação	40	
		Divisão	0	
		Função	43	

Com base nos dados apresentados na Tabela 10, observa-se que o número de operações matemáticas foi reduzido consideravelmente, e essa redução produziu uma diminuição do número de operações de ponto flutuante para determinação de todas as soluções.

Com o objetivo de confirmar a eficiência computacional para a solução da cinemática inversa de manipuladores robóticos antropomórficos utilizando a Teoria de Bases de Gröbner, uma nova configuração de juntas para o manipulador antropomórfico Unimation PUMA 560 será analisada na Seção 7.2.

## 7.2 CINEMÁTICA INVERSA DO PUMA 560 (2º CASO)

Considerar-se-á neste 2º caso, uma matriz de transformação homogênea obtida por meio de um novo conjunto de variáveis de juntas arbitrárias.

São substituídos arbitrariamente os seguintes valores de juntas:  $\theta = [-150^\circ, 90^\circ, 0^\circ, 180^\circ, 90^\circ, 30^\circ]$ , e demais parâmetros não nulos:  $a_2 = 432$ ,  $a_3 = 20$ ,  $d_3 = 125$  e  $d_4 = 430$  em (190), matriz de transformação homogênea determinada no Capítulo 6. Por conseguinte, determina-se a matriz (346).

$$T = \begin{bmatrix} 1 & 0 & 0 & \frac{125}{2} + 215\sqrt{3} \\ 0 & -1 & 0 & 215 - \frac{125\sqrt{3}}{2} \\ 0 & 0 & -1 & -452 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (346)$$

Estes mesmos valores, atribuídos neste caso, foram substituídos nas equações, deduzidas na Seção 6.2, para a solução da cinemática inversa desse manipulador por meio do método de Paul.

Utiliza-se o MAPLE para determinar todos os valores de  $\theta = [\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6]$ , que permitam ao robô assumir a configuração (posição + orientação) dada por (346). Todos esses resultados foram obtidos pelo código em MAPLE apresentado no Apêndice G deste trabalho.

A Tabela 11 ilustra o detalhamento de todas as operações matemáticas realizadas para solução deste problema, ângulo por ângulo.

Tabela 11: Número de operações para solução da cinemática inversa do PUMA 560 por meio do método de Paul – 2<sup>o</sup> Caso

Ângulo	Adição	Multiplicação	Divisão	Função
$\theta_{1(1)}$	7	12	1	12
$\theta_{1(2)}$	8	12	1	12
$\theta_{2(1)}$	11	20	0	12
$\theta_{2(2)}$	91	161	5	125
$\theta_{2(3)}$	92	161	5	125
$\theta_{2(4)}$	92	161	5	125
$\theta_{3(1)}$	10	19	0	12
$\theta_{3(2)}$	11	19	0	12
$\theta_{4(1)}$	0	0	0	0
$\theta_{4(2)}$	0	0	0	0
$\theta_{4(3)}$	0	0	0	0
$\theta_{4(4)}$	0	0	0	0
$\theta_{4(5)}$	0	1	0	0
$\theta_{4(6)}$	0	1	0	0
$\theta_{4(7)}$	0	0	0	0
$\theta_{4(8)}$	0	1	0	0
$\theta_{5(1)}$	0	1	0	0
$\theta_{5(2)}$	80	142	5	113
$\theta_{5(3)}$	81	142	5	113
$\theta_{5(4)}$	80	142	5	113
$\theta_{5(5)}$	0	1	0	0
$\theta_{5(6)}$	81	142	5	113
$\theta_{5(7)}$	80	142	5	113



Ângulo	Adição	Multiplicação	Divisão	Função
$\theta_{5(8)}$	81	142	5	113
$\theta_{6(1)}$	8	12	1	12
$\theta_{6(2)}$	264	469	21	376
$\theta_{6(3)}$	265	469	21	376
$\theta_{6(4)}$	264	469	21	376
$\theta_{6(5)}$	8	13	1	12
$\theta_{6(6)}$	265	469	21	376
$\theta_{6(7)}$	265	469	21	376
$\theta_{6(8)}$	265	469	21	376

De acordo com os dados de desempenho computacional mensurados por meio do MAPLE, são realizadas 34514 operações de ponto flutuante (2409 adições, 4261 multiplicações, 175 divisões e 3393 funções) para a solução do problema cinemático inverso deste manipulador utilizando as equações deduzidas na Seção 6.2 pelo método de Paul.

Pode-se observar na Tabela 12 as oito configurações de junta distintas, número máximo de configurações para esse manipulador.

Tabela 12: Configurações de juntas do Puma 560 (em radianos) obtidas por meio do método de Paul – 2º Caso

$\theta_1$ (radianos)	$\theta_2$ (radianos)	$\theta_3$ (radianos)	$\theta_4$ (radianos)	$\theta_5$ (radianos)	$\theta_6$ (radianos)
-2,6180	1,5708	0	0	-1,5708	-2,6180
-2,6180	1,5708	0	3,1416	1,5708	0,52361
-2,6180	3,0917	-3,0486	3,1416	0,043080	0,52361
-2,6180	3,0917	-3,0486	0	-0,043080	-2,6180
-0,042214	1,5708	-3,0486	0	1,4778	-0,042211
-0,042214	1,5708	-3,0486	3,1416	-1,4778	3,0994
-0,042214	0,049876	0	0	-0,049871	-0,042211
-0,042214	0,049876	0	3,1416	0,049871	3,0994

A partir deste ponto, as Equações (192) a (203) são reescritas substituindo  $n_x$ ,  $n_y$ ,  $n_z$ ,  $o_x$ ,  $o_y$ ,  $o_z$ ,  $a_x$ ,  $a_y$ ,  $a_z$ ,  $x$ ,  $y$  e  $z$  pelos respectivos valores da matriz numérica obtida. Todos esses 12 polinômios, adicionados aos 6 polinômios obtidos por intermédio da relação trigonométrica fundamental para cada variável de junta, são utilizados para gerar uma base de Gröbner utilizando a ordem lexicográfica  $c_6 \succ s_6 \succ c_5 \succ s_5 \succ c_4 \succ s_4 \succ c_2 \succ s_2 \succ c_3 \succ s_3 \succ c_1 \succ s_1$ . O código em MAPLE para determinação da base, com suas respectivas saídas, pode ser visualizado no Apêndice H deste trabalho.

Todos os algoritmos foram executados no mesmo *software* de Computação Algébrica, com o objetivo de comparar a eficiência computacional para solução do problema da cinemática inversa utilizando o método de Paul e a Teoria de Bases de Gröbner.

Com a realização de 466 operações de ponto flutuante (46 adições, 100 multiplicações e 40 funções), a seguinte base de Gröbner contendo 12 polinômios é obtida:

$$g_1 = 8600\sqrt{3}s_1 + 32084s_1^2 + 4300\sqrt{3} + 2500s_1 - 6771, \quad (347)$$

$$g_2 = -8021\sqrt{3}s_1 - 1250\sqrt{3} + 5521c_1 - 8600s_1 - 4300, \quad (348)$$

$$g_3 = 1853s_3^2 + 172s_3, \quad (349)$$

$$g_4 = 2c_3 - 43s_3 - 2, \quad (350)$$

$$\begin{aligned} g_5 = & 18578641250s_1s_3\sqrt{3} + 862257500\sqrt{3}s_1 + 9289320625\sqrt{3}s_3 \\ & + 63910525900s_1s_3 + 431128750\sqrt{3} + 2966165800s_1 + 2148795284s_2 \\ & + 9959875000s_3 - 665712384, \end{aligned} \quad (351)$$

$$\begin{aligned} g_6 = & -226593250\sqrt{3}s_1 - 113296625\sqrt{3} + 537198821c_2 - 779480780s_1 \\ & - 5780183345s_3 - 389740390, \end{aligned} \quad (352)$$

$$g_7 = s_4, \quad (353)$$

$$g_8 = c_4^2 - 1, \quad (354)$$

$$\begin{aligned} g_9 = & 433134000c_4s_3s_1\sqrt{3} - 431128750s_1c_4\sqrt{3} + 216567000c_4s_3\sqrt{3} \\ & + 1489980960c_4s_3s_1 - 215564375\sqrt{3}c_4 - 1483082900s_1c_4 + 11821264848c_4s_3 \\ & + 332856192c_4 + 1074397642s_5, \end{aligned} \quad (355)$$

$$\begin{aligned} g_{10} = & -4656190500s_1s_3\sqrt{3} - 226593250\sqrt{3}s_1 - 2328095250\sqrt{3}s_3 \\ & - 16017295320s_1s_3 - 113296625\sqrt{3} + 537198821c_5 - 779480780s_1 \\ & - 2226636864s_3 - 389740390, \end{aligned} \quad (356)$$

$$g_{11} = -c_4s_1 + s_6, \quad (357)$$

$$g_{12} = -8021s_1c_4\sqrt{3} - 1250\sqrt{3}c_4 - 8600s_1c_4 - 4300c_4 + 5521c_6. \quad (358)$$

Para encontrar todos os valores de  $\theta = [\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6]$ , que permitam ao robô assumir a configuração dada por (346), utiliza-se o comando Solve do MAPLE.

Em um total de 954 operações de ponto flutuante (46 adições, 100 multiplicações e 101 funções), obtém-se oito configurações de junta distintas, exatamente iguais às configurações determinadas nesta seção (Tabela 12), conforme pode-se observar na Tabela 13.

Tabela 13: Configurações de juntas (em radianos) do Puma 560 obtidas por meio da Teoria de Bases de Gröbner – 2º Caso

$\theta_1$ (radianos)	$\theta_2$ (radianos)	$\theta_3$ (radianos)	$\theta_4$ (radianos)	$\theta_5$ (radianos)	$\theta_6$ (radianos)
-2,6180	1,5708	0	0	-1,5708	-2,6180
-2,6180	1,5708	0	3,1416	1,5708	0,52361
-2,6180	3,0917	-3,0486	3,1416	0,043080	0,52361
-2,6180	3,0917	-3,0486	0	-0,043080	-2,6180
-0,042214	1,5708	-3,0486	0	1,4778	-0,042211
-0,042214	1,5708	-3,0486	3,1416	-1,4778	3,0994
-0,042214	0,049876	0	0	-0,049871	-0,042211
-0,042214	0,049876	0	3,1416	0,049871	3,0994

O código em MAPLE para determinação das oito soluções, com suas respectivas saídas, encontra-se no Apêndice I deste trabalho.

Utilizando o método de Paul, todas as oito soluções foram calculadas pelo MAPLE, sendo necessário a realização de 34514 operações de ponto flutuante para determinação das oito configurações de juntas distintas.

Pode-se observar na Tabela 14, em que todos os ângulos da Tabela 13 foram convertidos em graus, que a linha em destaque contém os valores das juntas  $\theta = [-150^\circ, 90^\circ, 0^\circ, 180^\circ, 90^\circ, 30^\circ]$ , utilizados para construir a matriz de transformação  $T$  utilizada neste caso.

Tabela 14: Configurações de juntas (em graus) do Puma 560 obtidas por meio da Teoria de Bases de Gröbner – 2º Caso

$\theta_1$ (graus)	$\theta_2$ (graus)	$\theta_3$ (graus)	$\theta_4$ (graus)	$\theta_5$ (graus)	$\theta_6$ (graus)
-150,000	90,000	0	0	-90,000	-150,000
-150,000	90,000	0	180,000	90,000	30,000
-150,000	177,141	-174,672	180,000	2,468	30,000
-150,000	177,141	-174,672	0	-2,468	-150,000
-2,418	90,000	-174,672	0	84,671	-2,418
-2,418	90,000	-174,672	180,000	-84,671	177,582
-2,418	2,857	0	0	-2,857	-2,418
-2,418	2,857	0	180,000	2,857	177,582

Adicionando as 466 operações de ponto flutuante para obtenção da base de Gröbner, ao número de flop para obtenção das oito configurações de junta distintas por meio da base de Gröbner gerada (954 flop); obtém-se 1420 operações de ponto flutuante. Pode-se observar que, para esse conjunto de configurações, houve uma redução de 34514 para 1420 do número de operações de ponto flutuante realizadas para a solução do problema cinemático inverso.

Na Tabela 15, pode-se observar o detalhamento do número de todas as operações matemáticas realizadas e do número de operações de ponto flutuante necessários para solução da cinemática inversa deste manipulador por meio dos dois métodos analisados.

Tabela 15: Dados de desempenho computacional para solução da cinemática inversa do manipulador Puma 560 por meio dos dois métodos – 2º Caso

Método		Operações Matemáticas		Operações de Ponto Flutuante
Paul		Adição	2409	34514 flop
		Multiplicação	4261	
		Divisão	175	
		Função	3393	
Teoria de Bases de Gröbner	Geração da Base	Adição	46	466 flop
		Multiplicação	100	
		Divisão	0	
		Função	40	
	Determinação das Soluções	Adição	46	954 flop
		Multiplicação	100	
		Divisão	0	
		Função	101	

Com base nos dados apresentados na Tabela 15, pode-se observar que, mesmo considerando uma matriz de transformação homogênea obtida a partir de um conjunto de variáveis de juntas não nulas, o número de operações matemáticas foi reduzido consideravelmente. Essa diminuição pode ser justificada pela redução do número de funções utilizadas pelo segundo método, redução ocorrida devido ao fato de se trabalhar apenas com equações polinomiais quando uma base de Gröbner é utilizada.

A ordem lexicográfica (359) foi utilizada originalmente para gerar as bases de Gröbner utilizadas nos dois casos analisados.

$$c_6 \succ s_6 \succ c_5 \succ s_5 \succ c_4 \succ s_4 \succ c_2 \succ s_2 \succ c_3 \succ s_3 \succ c_1 \succ s_1 \quad (359)$$

A título de experimentação, duas novas bases de Gröbner, uma para cada matriz de transformação homogênea utilizada nos dois casos analisados, foram geradas com base na nova ordem lexicográfica (360).

$$c_1 \succ s_1 \succ c_3 \succ s_3 \succ c_2 \succ s_2 \succ c_4 \succ s_4 \succ c_5 \succ s_5 \succ c_6 \succ s_6 \quad (360)$$

Os códigos em MAPLE para determinação destas novas bases, com suas respectivas saídas, podem ser visualizados no Apêndice J deste trabalho. Observa-se, por meio das saídas apresentadas pelo MAPLE, que essa nova ordem produz bases maiores, exigindo maiores esforços computacionais.

As oito soluções para cada caso, exatamente iguais às determinadas pela ordem lexicográfica inicialmente escolhida, também são determinadas pelo MAPLE.

Pelos dados de desempenho computacional apresentados no Apêndice J, observa-se que o número de operações de ponto flutuante para determinação das oito soluções são relativamente próximos, porém, quando o custo total (geração da base de Gröbner + determinação de todas as configurações de juntas) é analisado, dados apresentados na Tabela 16, constata-se que a ordem lexicográfica inicialmente escolhida é a melhor opção para os dois casos avaliados, uma vez que o custo computacional total é aproximadamente 51,92% menor no 1º Caso e, aproximadamente 38,95% menor no 2º Caso.

Tabela 16: Comparação do desempenho computacional para solução da cinemática inversa do manipulador Puma 560 utilizando ordens lexicográficas distintas para os dois casos estudados

Ordem lexicográfica	1º Caso	2º Caso
$c_6 \succ s_6 \succ c_5 \succ s_5 \succ c_4 \succ s_4 \succ c_2 \succ s_2 \succ c_3 \succ s_3 \succ c_1 \succ s_1$	474 flop	1420 flop
$c_1 \succ s_1 \succ c_3 \succ s_3 \succ c_2 \succ s_2 \succ c_4 \succ s_4 \succ c_5 \succ s_5 \succ c_6 \succ s_6$	986 flop	2326 flop

Por conseguinte, pode-se identificar que, por meio dos dados de desempenho computacional obtidos, a melhor ordem lexicográfica a ser utilizada é aquela que inicie partindo das variáveis de juntas mais próximas do efetuador final até a base do manipulador, seguindo a ordem para determinação das variáveis de juntas determinada pelo método de Paul. Ou seja, para o manipulador Unimation PUMA 560, tem-se  $c_6 \succ s_6 \succ c_5 \succ s_5 \succ c_4 \succ s_4 \succ c_2 \succ s_2 \succ c_3 \succ s_3 \succ c_1 \succ s_1$  como sendo a ordem lexicográfica que deve ser preferencialmente adotada. Todos os códigos, com suas respectivas saídas que confirmam essa afirmação, podem ser visualizados neste mesmo Apêndice J.

A eficiência computacional ao se utilizar o método da Teoria de Bases de Gröbner para a solução da cinemática inversa de manipuladores antropomórficos pode ser questionada devido ao fato do manipulador PUMA 560 possuir três eixos que se cruzam em seu pulso, o que permite o desacoplamento do problema da cinemática inversa em dois subproblemas: um de posicionamento e outro de orientação, procedimento conhecido como método do desacoplamento de Pieper, mencionado na Seção 2.2.

Essa característica geométrica implica em uma redução do esforço computacional para a determinação de uma base de Gröbner, bem como para obtenção de todas as configurações de juntas, uma vez que existem apenas dois parâmetros  $d_i$  diferentes de zero.

Com o objetivo de ilustrar o alcance do método da Teoria de Bases de Gröbner, considerar-se-á um manipulador 6R com pulso não esférico. A cinemática inversa deste manipulador será resolvida por meio de uma base de Gröbner na Seção 7.3.



### 7.3 CASO ESPECIAL: MANIPULADOR 6R COM PULSO NÃO ESFÉRICO

Segundo Ghosal (2006), para manipuladores onde três eixos consecutivos não se cruzam, os cálculos necessários para resolver o problema da cinemática inversa são muito mais complexos.

Com o objetivo de confirmar a eficiência computacional para a solução da cinemática inversa de manipuladores robóticos antropomórficos utilizando a Teoria de Bases de Gröbner, a partir deste ponto, serão determinadas todas as configurações de juntas de um manipulador 6R com pulso não esférico.

Todos os parâmetros cinemáticos deste novo manipulador estão listados na Tabela 17, em que  $i$  representa o número da junta.

Tabela 17: Parâmetros cinemáticos de um manipular 6R com pulso não esférico

$i$	$\alpha_{i-1}$ (graus)	$a_{i-1}$ (mm)	$d_i$ (mm)	$\theta_i$ (graus)
1	0	0	0	$\theta_1$
2	-90	0	0	$\theta_2$
3	0	$a_2$	$d_3$	$\theta_3$
4	-90	$a_3$	$d_4$	$\theta_4$
5	90	0	$d_5$	$\theta_5$
6	-90	0	0	$\theta_6$

Diferentemente da Tabela 4, é notado na Tabela 17 a presença de um terceiro parâmetro  $d_i$ , identificado como  $d_5$ .

Após a construção de todas as matrizes de transformação (361) a (366), e multiplicação delas (367); obtém-se a matriz de transformação homogênea (368), solução do problema da cinemática direta deste manipulador.

$${}^0A_1 = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (361)$$

$${}^1A_2 = \begin{bmatrix} c_2 & -s_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_2 & -c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (362)$$

$${}^2A_3 = \begin{bmatrix} c_3 & -s_3 & 0 & a_2 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (363)$$

$${}^3A_4 = \begin{bmatrix} c_4 & -s_4 & 0 & a_3 \\ 0 & 0 & 1 & d_4 \\ -s_4 & -c_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (364)$$

$${}^4A_5 = \begin{bmatrix} c_5 & -s_5 & 0 & 0 \\ 0 & 0 & -1 & d_5 \\ s_5 & c_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (365)$$

$${}^5A_6 = \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_6 & -c_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (366)$$

$$T = {}^0A_1(\theta_1) \cdot {}^1A_2(\theta_2) \cdot {}^2A_3(\theta_3) \cdot {}^3A_4(\theta_4) \cdot {}^4A_5(\theta_5) \cdot {}^5A_6(\theta_6) \quad (367)$$

$$T = \begin{bmatrix} n_x & o_x & a_x & x \\ n_y & o_y & a_y & y \\ n_z & o_z & a_z & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (368)$$

Em que as entradas de  $T$  são dadas por:

$$\begin{aligned} n_x = & (((c_1c_2c_3 - c_1s_2s_3)c_4 + s_1s_4)c_5 + (-c_1c_2s_3 - c_1c_3s_2)s_5)c_6 \\ & - ((c_1c_2c_3 - c_1s_2s_3)s_4 - s_1c_4)s_6, \end{aligned} \quad (369)$$

$$n_y = (((c_2c_3s_1 - s_1s_2s_3)c_4 - c_1s_4)c_5 + (-c_2s_1s_3 - c_3s_1s_2)s_5)c_6 - ((c_2c_3s_1 - s_1s_2s_3)s_4 + c_1c_4)s_6, \quad (370)$$

$$n_z = ((-c_2s_3 - c_3s_2)c_4c_5 + (-c_2c_3 + s_2s_3)s_5)c_6 - (-c_2s_3 - c_3s_2)s_4s_6, \quad (371)$$

$$o_x = -(((c_1c_2c_3 - c_1s_2s_3)c_4 + s_1s_4)c_5 + (-c_1c_2s_3 - c_1c_3s_2)s_5)s_6 - ((c_1c_2c_3 - c_1s_2s_3)s_4 - s_1c_4)c_6, \quad (372)$$

$$o_y = -(((c_2c_3s_1 - s_1s_2s_3)c_4 - c_1s_4)c_5 + (-c_2s_1s_3 - c_3s_1s_2)s_5)s_6 - ((c_2c_3s_1 - s_1s_2s_3)s_4 + c_1c_4)c_6, \quad (373)$$

$$o_z = -((-c_2s_3 - c_3s_2)c_4c_5 + (-c_2c_3 + s_2s_3)s_5)s_6 - (-c_2s_3 - c_3s_2)s_4c_6, \quad (374)$$

$$a_x = -((c_1c_2c_3 - c_1s_2s_3)c_4 + s_1s_4)s_5 + (-c_1c_2s_3 - c_1c_3s_2)c_5, \quad (375)$$

$$a_y = -((c_2c_3s_1 - s_1s_2s_3)c_4 - c_1s_4)s_5 + (-c_2s_1s_3 - c_3s_1s_2)c_5, \quad (376)$$

$$a_z = -(-c_2s_3 - c_3s_2)c_4s_5 + (-c_2c_3 + s_2s_3)c_5, \quad (377)$$

$$x = -(c_1c_2c_3 - c_1s_2s_3)s_4 + s_1c_4)d_5 + (c_1c_2c_3 - c_1s_2s_3)a_3 + (-c_1c_2s_3 - c_1c_3s_2)d_4 + c_1c_2a_2 - s_1d_3, \quad (378)$$

$$y = -(c_2c_3s_1 - s_1s_2s_3)s_4 - c_1c_4)d_5 + (c_2c_3s_1 - s_1s_2s_3)a_3 + (-c_2s_1s_3 - c_3s_1s_2)d_4 + s_1c_2a_2 + c_1d_3, \quad (379)$$

$$z = -(-c_2s_3 - c_3s_2)s_4d_5 + (-c_2s_3 - c_3s_2)a_3 + (-c_2c_3 + s_2s_3)d_4 - s_2a_2. \quad (380)$$

São substituídos arbitrariamente os seguintes valores de junta:  $\theta = [0^\circ, 0^\circ, 0^\circ, 0^\circ, 0^\circ, 0^\circ]$ , e demais parâmetros não nulos:  $a_2 = 432$ ,  $a_3 = 20$ ,  $d_3 = 125$ ,  $d_4 = 430$  e  $d_5 = 20$  em (367).

Foram utilizados os mesmos valores atribuídos aos parâmetros  $a_2$ ,  $a_3$ ,  $d_3$  e  $d_4$  do manipulador PUMA 560. Dessarte, determina-se a matriz:

$$T = \begin{bmatrix} 1 & 0 & 0 & 452 \\ 0 & -1 & 0 & 105 \\ 0 & 0 & -1 & -430 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (381)$$

Este exemplo servirá para comprovar que o método da Teoria de Bases de Gröbner é uma solução computacionalmente viável para a solução da cinemática

inversa de manipuladores robóticos antropomórficos, inclusive para os que possuem três eixos consecutivos que não se cruzam.

Desse modo, as Equações (369) a (380) são reescritas substituindo  $n_x, n_y, n_z, o_x, o_y, o_z, a_x, a_y, a_z, x, y$  e  $z$  pelos respectivos valores da matriz numérica obtida (381).

Todos esses 12 polinômios, adicionados aos 6 polinômios obtidos por meio da relação trigonométrica fundamental para cada variável de junta, são utilizados para gerar uma base de Gröbner, utilizando a ordem lexicográfica  $c_6 \succ s_6 \succ c_5 \succ s_5 \succ c_4 \succ s_4 \succ c_2 \succ s_2 \succ c_3 \succ s_3 \succ c_1 \succ s_1$ . O código em MAPLE para determinação da base, com suas respectivas saídas, pode ser visualizado no Apêndice K deste trabalho.

Com a realização de 1100 operações de ponto flutuante (188 adições e 912 multiplicações), uma base de Gröbner contendo 14 polinômios é obtida. Por conta do comprimento dos coeficientes dos polinômios gerados, a base não poderá ser apresentada no corpo do texto desta seção, alguns coeficientes chegam a ter 315 dígitos. Todos os polinômios da base de Gröbner podem ser visualizados no Apêndice K deste trabalho.

Para encontrar todos os valores de  $\theta = [\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6]$ , que permitam ao robô assumir a configuração (posição + orientação) dada por (381), utiliza-se o comando Solve do MAPLE por meio da base de Gröbner gerada.

Em um total de 2924 operações de ponto flutuante (188 adições, 912 multiplicações e 228 funções), obtém-se oito configurações de junta distintas, conforme pode-se visualizar na Tabela 18.

Tabela 18: Configurações de juntas (em radianos) de um manipular 6R com pulso não esférico

$\theta_1$ (rad)	$\theta_2$ (rad)	$\theta_3$ (rad)	$\theta_4$ (rad)	$\theta_5$ (rad)	$\theta_6$ (rad)
0	0	0	0	0	0
0	1,5209	-3,0486	0	1,5277	0
-0,083106	0	0	2,6020	0	-2,6851
-2,6851	3,1416	-3,0486	0	-0,092956	-2,6851
-2,6851	1,6207	0	0	-1,6207	-2,6851
-2,6188-0,060089i	3,1416	3,1416	1,8665-1,1349i	0	1,8005+1,0748i
0,089541	1,5470	-3,0755	3,1416	-1,5287	3,0521
1,8314-0,28856i	1,5716-1,3116i	1,5700+1,3116i	-0,092287-4,0819i	3,1416	1,9365-4,3538i

O código em MAPLE para determinação das oito soluções, com suas respectivas saídas, encontra-se no Apêndice L deste trabalho. Pode-se observar que, das oito configurações determinadas, o manipulador não poderá assumir duas, uma vez que os valores de  $\theta$  destacados na Tabela 18 são dados por números complexos.

Com o objetivo de validar os resultados obtidos, pode-se observar na Tabela 19, onde todos os ângulos da Tabela 18 foram convertidos em graus, que a linha em destaque contém os valores das juntas  $\theta = [0^\circ, 0^\circ, 0^\circ, 0^\circ, 0^\circ, 0^\circ]$  utilizados para construir a matriz de transformação (381) utilizada neste caso.

Tabela 19: Configurações de juntas (em graus) de um manipular 6R com pulso não esférico

$\theta_1$ (graus)	$\theta_2$ (graus)	$\theta_3$ (graus)	$\theta_4$ (graus)	$\theta_5$ (graus)	$\theta_6$ (graus)
0	0	0	0	0	0
0	87,141	-174,671	0	87,530	0
-4,761	0	0	149,083	0	-153,844
-153,844	180,000	-174,671	0	-5,325	-153,844
-153,844	92,859	0	0	-92,859	-153,844
Indeterminado	180,000	180,000	Indeterminado	0	Indeterminado
5,130	88,636	-176,213	180,000	-87,588	174,872
Indeterminado	Indeterminado	Indeterminado	Indeterminado	180,0004209	Indeterminado

Com o intuito de validar todas as configurações de juntas encontradas, todos os conjuntos de soluções foram substituídos na Equação (368), matriz de transformação homogênea do manipulador, na ordem apresentada na Tabela 19.

Todas as matrizes obtidas são aproximadamente iguais à matriz de transformação homogênea, exceto as duas matrizes geradas por meio das duas configurações em destaque na Tabela 18, obviamente por conta das variáveis de juntas complexas, indicando que, efetivamente pelo uso da Teoria de Bases de Gröbner, é possível obter todas as soluções possíveis.

Todos os códigos implementados no MAPLE, que confirmam essa afirmação, encontram-se no Apêndice M deste trabalho.

Adicionando as 1100 operações de ponto flutuante para obtenção da base de Gröbner ao número de operações de ponto flutuante para obtenção das oito configurações de junta distintas pelos polinômios da base gerada (2924 flop); obtém-se 4024 operações de ponto flutuante.

Todos os dados de eficiência computacional para este manipular 6R com pulso não esférico estão representados na Tabela 20.

Tabela 20: Dados de desempenho computacional para um manipular 6R com pulso não esférico

Teoria de Bases de Gröbner	Operações Matemáticas		Operações de Ponto Flutuante
Geração da Base	Adição	188	1100 flop
	Multiplicação	912	
	Divisão	0	
	Função	0	
Determinação das Soluções	Adição	188	2924 flop
	Multiplicação	912	
	Divisão	0	
	Função	228	

Com base nos dados de desempenho computacional apresentados na Tabela 20, pode-se afirmar que, mesmo para um manipulador 6R com pulso não esférico, a Teoria de Bases de Gröbner é um método eficiente computacionalmente para solução do problema cinemático inverso de um manipulador antropomórfico, independentemente de sua configuração de juntas do punho.

Mesmo considerando que sua geometria, de acordo com Ghosal (2006), implica em cálculos muito mais complexos para a solução do problema da cinemática inversa comparado a um manipulador 6R com punho esférico, o custo computacional total para determinação de todas as soluções corresponde a aproximadamente 11,7% do custo total para solução de um manipulador 6R com punho esférico utilizando o método de Paul por meio de uma matriz de transformação homogênea determinada pela mesma configuração de juntas.

Pode-se afirmar, com base em todos os dados de eficiência computacional obtidos, que o método proposto neste trabalho para a solução da cinemática inversa de um robô manipulador antropomórfico é mais eficiente computacionalmente que o método de Paul.

## 8 CONCLUSÕES

Ao utilizar a Teoria de Bases de Gröbner para a solução da cinemática inversa de robôs do tipo SCARA, um modelo algébrico é desenvolvido para descrever o comportamento do manipulador robótico, sem a necessidade da construção das matrizes de transformação homogênea, porém vale salientar que isso é possível devido apenas à simplicidade da geometria dessa família de manipuladores robóticos.

Para manipuladores SCARA, o método da Teoria de Bases de Gröbner evita o problema da determinação de soluções apenas dentro de um domínio limitador. Ao desenvolver um modelo algébrico usando equações polinomiais simples, todo o círculo unitário é incorporado com o auxílio das identidades trigonométricas. Não existe a necessidade da manipulação de equações do modelo para determinar mais soluções.

Por meio da utilização das Bases de Gröbner, é possível determinar todos os pontos que podem ser alcançados dentro do volume de trabalho de um manipulador SCARA. De fato, quando a Teoria de Bases de Gröbner é utilizada para a solução da cinemática inversa dessa família de manipuladores robóticos, as configurações de juntas que devem ser evitadas também podem ser evidenciadas; configurações essas, consideradas singulares.

Com relação à eficiência computacional, foco deste trabalho, após uma análise mais minuciosa das equações determinadas por dois métodos para obtenção das variáveis de junta rotativas do manipulador SCARA, observação feita na Subseção 5.3.1, levando em consideração apenas o custo computacional, pode-se afirmar que, para essa família de manipuladores robóticos, não há vantagem ao aplicar a Teoria de Bases de Gröbner para solucionar o problema da cinemática inversa, uma vez que as equações para determinação das duas variáveis de juntas, deduzidas por meio dos dois métodos, são exatamente iguais. Essa desvantagem resume-se ao fato de que, ao se



utilizar esse método para solucionar o problema da cinemática inversa de manipuladores do tipo SCARA, adiciona-se um custo computacional extra desnecessário para gerar a base de Gröbner que produzirá equações exatamente iguais às obtidas pelo método clássico.

Como o foco principal deste trabalho é a análise da eficiência computacional ao se utilizar a Teoria de Bases de Gröbner para solucionar o problema cinemático inverso de manipuladores antropomórficos, foi escolhido o Unimation PUMA 560, um manipulador antropomórfico 6R com pulso esférico. A escolha desse manipulador deu-se por conta de sua geometria, uma vez que a sua cinemática inversa é solucionada por meio de equações não lineares, transcendentais e que são resolvidas com elevado custo computacional.

Ao utilizar o método das Bases de Gröbner para a solução da cinemática inversa de manipuladores robóticos antropomórficos, observa-se, claramente nos testes realizados durante esta pesquisa, que ele se mostrou computacionalmente mais eficiente que o método de Paul, uma vez que as equações produzidas para determinação das seis variáveis de junta são matematicamente mais simples de serem resolvidas. Sendo que, por meio de substituições retroativas, pode-se determinar todas as soluções de configurações de juntas para posicionar e orientar o efetuador do manipulador robótico no espaço.

Com base nos dados de desempenho computacional obtidos pelas mudanças na ordem lexicográfica para determinação das bases de Gröbner, a sequência que inicia partindo das variáveis de juntas mais próximas do efetuador final até a base do manipulador produziu soluções com menor custo computacional.

O número de operações de ponto flutuante necessárias para resolver o problema da cinemática inversa de um manipulador antropomórfico, ao utilizar uma base de Gröbner, é reduzido consideravelmente. De acordo com os dados de desempenho computacional obtidos, pode-se afirmar que o método proposto neste trabalho para a

---

solução da cinemática inversa de um robô manipulador 6R mostrou-se ser mais eficiente computacionalmente que o método de Paul, inclusive para o caso em que o manipulador robótico não possua três eixos consecutivos que se cruzem, ou seja, para um manipulador com pulso não esférico.

Dados obtidos confirmam que, mesmo considerando que sua geometria implica em cálculos muito mais complexos para a solução do problema da cinemática inversa comparado a um manipulador 6R com punho esférico, o custo computacional total para determinação de todas as soluções corresponde a aproximadamente 11,7% do custo total para solução de um manipulador 6R com punho esférico utilizando o método de Paul por uma matriz de transformação homogênea determinada pela mesma configuração de juntas. Dessa forma, entende-se evidenciar a razão pela qual esse método é mais eficiente para essa família de manipuladores robóticos.

Pode-se concluir que a utilização da Teoria de Bases de Gröbner para a solução da cinemática inversa de manipuladores antropomórficos é uma alternativa com maior eficiência computacional viável. Em trabalhos futuros, pretende-se demonstrar que esse método também pode ser eficiente computacionalmente ao ser comparado com outros para a solução do problema cinemático inverso de outras classes de manipuladores robóticos, como, por exemplo, robôs seriais redundantes.

## REFERÊNCIAS

ADAMS, W.; LOUSTAUNAU, P. An Introduction to Groebner bases. Providence: American Mathematical Society, v. 3, 1994.

ADDISON, C. et al. The Genesis distributed-memory benchmarks. Part I: Methodology and general relativity benchmark with results for the SUPRENUM computer. Concurrency: Practice and Experience, v. V, p. 1-22, Fevereiro 1993.

BARRIENTOS, A. et al. Fundamentos de Robótica. 2<sup>a</sup>. ed. Madrid: McGraw-Hill, 2007.

BOAS, A. A. V.; MARCOS, E. D. N. Uma Introdução a Teoria de Bases de Gröbner para Álgebras Associativas. São Paulo: Livraria da Física, 2013.

BOSE, N. K. Gröbner Bases: An Algorithmic Method in Polynomial Ideal Theory. In: BOSE, N. K. Multidimensional Systems Theory and Applications. Dordrecht: Springer Netherlands, 1995. p. 89-127.

BUCHBERGER, B. Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal, PhD thesis. Innsbruck: Dissertation an dem Math. Inst. der Universität von Innsbruck, 1965.

BUCHBERGER, B. Bruno Buchberger's PhD thesis 1965: An algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal. Journal of Symbolic Computation, v. 41, p. 475-511, 2006.

---

CHAKRAVARTI, M.; SIVAKUMAR, A. L. A GA-Integrated Adaptive Model Reference Controller in Robot Tracking Applications. *Journal of Multidisciplinary Engineering Science Studies (JMESS)*, 2016.

COX, D.; LITTLE, ; O'SHEA, D. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*. 4<sup>a</sup>. ed. New York: Springer, 1992.

CRAIG, J. J. *Introduction To Robotics: Mechanics And Control*. 4th. ed. Boston: Addison-Wesley, 2017.

DAYA, B.; KHAWANDI, S.; AKOUM, M. Applying Neural Network Architecture for Inverse Kinematics Problem in Robotics. *Journal of Software Engineering and Applications*, v. III, p. 230-239, 2010.

DENAVIT, J.; HARTENBERG, R. S. A kinematic notation for lower-pair mechanisms based on matrices. *Trans ASME J. Appl. Mech.*, p. 215-221, 1955.

DUFFY, J.; CRANE, C. D. Displacement analysis of the general 7-Link 7R mechanism. *Mechanism and Machine Theory*, v. XV, p. 153–169, 1980.

FAUGÈRE, J.-C. A new efficient algorithm for computing Gröbner bases (F4). *Journal of Pure and Applied Algebra*, v. 139, p. 61-88, 1999.

GHOSAL, A. *Robotics: Fundamental Concepts and Analysis*. New Delhi: Oxford University Press, 2006.

---

GUZMÁN-GIMÉNEZ, J. et al. Synthesis of the Inverse Kinematic Model of Non-Redundant Open-Chain Robotic Systems Using Groebner Basis Theory. *Applied Sciences*, April 2020.

HARTENBERG, R. S.; DENAVIT, J. Kinematic synthesis of linkages. Col: McGraw-Hill series in mechanical engineering. New York: McGraw-Hill, 1965.

KENDRICKS, K. D. Solving the Inverse Kinematic Robotics Problem for the GMF-A510 Robot: A Comparison Study of the Denavit-Hartenberg Matrix and Groebner Basis Theory. PhD Thesis. Auburn: The Graduate School of Auburn University: Auburn University Libraries, 2007.

LEE, C. S. G. Robot Arm Kinematics, Dynamics, and Control. *Computer*, Ann Arbor, p. 62-80, 1982.

NI, Z.; WU, R. 6R Robot Inverse Solution Algorithm Based on Quaternion Matrix and Groebner Base. *Advances in Linear Algebra & Matrix Theory*, v. VIII, p. 33-40, 2018.

PAUL, R. P. Robot Manipulators: Mathematics, Programming, And Control (Artificial Intelligence). Cambridge, MA: MIT Press, 1981.

PIEPER, D. L. The kinematics of manipulators under computer control. PhD thesis. Stanford: Stanford University, Department of Mechanical Engineering, 1968.

---

ROBBIANO, L. Groebner Bases and Statistics. In: BUCHBERGER, B.; WINKLER, F. Gröbner Bases and Applications (London Mathematical Society Lecture Note Series). Cambridge: Cambridge University Press, 1998. p. 179-204.

SCIAVICCO, L.; SICILIANO, B. Modeling and Control of Robot Manipulator. New York: McGraw-Hill, 1996.

SPONG, M. W.; VIDYASAGAR, M. Robot dynamics and control. [S.l.]: Wiley India Pvt. Limited, 2008.

TSAI, L.-W. Robot Analysis: The Mechanics of Serial and Parallel Manipulators. New York: John Wiley & Sons, 1999.

UEBERHUBER, C. W. Numerical Computation 1: Methods, Software, and Analysis. New York: Springer, 1997.

ULLRICH, R. A. Robótica - Uma Introdução. Rio de Janeiro: Editora Campus, 1987.

WANG, D. Groebner Bases Applied to Geometric Theorem Proving and Discovering. In: BUCHBERGER, B.; WINKLER, F. Gröbner Bases and Applications (London Mathematical Society Lecture Note Series). Cambridge: Cambridge University Press, 1998. p. 281-302.

WANG, Y.; HANG, L.; YANG, T. Inverse Kinematics Analysis of General 6R Serial Robot Mechanism Based on Groebner Base. Frontiers of Mechanical Engineering in China, n. I, p. 115-124, January 2006.

WENZ, M.; WORN, H. Solving the inverse kinematics problem symbolically by means of knowledge-based and linear algebra-based methods. IEEE International Conference on Emerging Technologies and Factory Automation, ETFA, p. 1346-1353, Outubro 2007.

## APÊNDICE A

A divisão do polinômio  $f = x^2y - y$  por  $g_1 = x^2 - y$  e  $g_2 = xy - x$  será realizada utilizando a ordem lexicográfica graduada  $x \succ_{grlex} y$ , e em seguida, os cálculos serão refeitos invertendo a ordem de  $g_1$  e  $g_2$ , segundo exemplo apresentado na Subseção 3.4.2.

Tem-se as seguintes etapas para a divisão de  $f$  por  $g_1$  e  $g_2$ :

- i. Inicialmente, atribui-se  $f$  a  $p$ , os divisores  $g_1$  e  $g_2$  são listados na chave e os quocientes  $a_1$ ,  $a_2$  e o resto  $r$  abaixo da chave. Tem-se o seguinte esquema:

$$\begin{array}{r|l}
 p : x^2y - y & g_1 : x^2 - y \\
 & g_2 : xy - x \\
 \hline
 & a_1 : 0 \\
 & a_2 : 0 \\
 & r : 0
 \end{array}$$

- ii. Observa-se que  $tl(g_1) = x^2$  divide  $tl(p) = x^2y$ . Dessa forma,  $a_1 = a_1 + \frac{tl(p)}{tl(g_1)} = y$

e  $p = p - \frac{tl(p)}{tl(g_1)} f = y^2 - y$ .

$$\begin{array}{r|l}
 p : x^2y - y & g_1 : x^2 - y \\
 -(x^2y - y^2) & g_2 : xy - x \\
 \hline
 y^2 - y & a_1 : y \\
 & a_2 : 0 \\
 & r : 0
 \end{array}$$



- iii. Percebe-se que  $tl(g_1) = x^2$  não divide  $tl(p) = y^2$ , bem como  $tl(g_2) = xy$  não divide  $tl(p) = y^2$ . Dessarte, move-se  $tl(p) = y^2$  para o resto  $r$ , ou seja,  $r = r + tl(p) = y^2$  e  $p = p - tl(p) = -y$ .

$$\begin{array}{r|l}
 x^2y - y & g_1 : x^2 - y \\
 -(x^2y - y^2) & g_2 : xy - x \\
 \hline
 p : \cancel{y^2} - y & a_1 : y \\
 & a_2 : 0 \\
 & r : y^2
 \end{array}$$

- iv. Nota-se que  $tl(g_1) = x^2$  não divide  $tl(p) = -y$ , bem como  $tl(g_2) = xy$  não divide  $tl(p) = -y$ . Por conseguinte, move-se  $tl(p) = -y$  para o resto  $r$ , ou seja,  $r = r + tl(p) = y^2 - y$  e  $p = p - tl(p) = 0$ .

$$\begin{array}{r|l}
 x^2y - y & g_1 : x^2 - y \\
 -(x^2y - y^2) & g_2 : xy - x \\
 \hline
 p : \cancel{y} & a_1 : y \\
 & a_2 : 0 \\
 & r : y^2 - y
 \end{array}$$

- v. Como  $p = 0$ , o algoritmo da pseudodivisão é concluído. Dessa forma, tem-se que:
- $$x^2y - y = y \cdot (x^2 - y) + 0 \cdot (xy - x) + (y^2 - y).$$

Partindo dos mesmos polinômios  $f$ ,  $g_1$  e  $g_2$ , utilizando a ordem lexicográfica graduada  $x \succ_{grlex} y$ , os cálculos serão refeitos invertendo a ordem de  $g_1$  e  $g_2$ . Tem-se, então, as seguintes etapas para a divisão de  $f$  por  $g_2$  e  $g_1$ :

- i. Inicialmente, atribui-se  $f$  a  $p$ , os divisores  $g_2$  e  $g_1$  são listados na chave e os quocientes  $a_1$ ,  $a_2$  e o resto  $r$  abaixo da chave. Tem-se o seguinte esquema:

$$\begin{array}{r|l}
 p : x^2y - y & g_2 : xy - x \\
 & g_1 : x^2 - y \\
 \hline
 & a_1 : 0 \\
 & a_2 : 0 \\
 & r : 0
 \end{array}$$

- ii. Nota-se que  $tl(g_2) = xy$  divide  $tl(p) = x^2y$ . Dessa forma,  $a_1 = a_1 + \frac{tl(p)}{tl(g_2)} = x$  e

$$p = p - \frac{tl(p)}{tl(g_2)} f = x^2 - y.$$

$$\begin{array}{r|l}
 p : x^2y - y & g_2 : xy - x \\
 -(x^2y - x^2) & g_1 : x^2 - y \\
 \hline
 x^2 - y & a_1 : x \\
 & a_2 : 0 \\
 & r : 0
 \end{array}$$

iii. Observa-se que  $tl(g_2) = xy$  não divide  $tl(p) = x^2$ , todavia,  $tl(g_1) = x^2$  divide

$$tl(p) = x^2. \text{ Portanto, } a_2 = a_2 + \frac{tl(p)}{tl(g_1)} = 1 \text{ e } p = p - \frac{tl(p)}{tl(g_2)} f = 0.$$

$x^2y - y$	$g_2 : xy - x$
$-(x^2y - x^2)$	$g_1 : x^2 - y$
$p : x^2 - y$	$a_1 : x$
$-(x^2 - y)$	$a_2 : 1$
$0$	$r : 0$

iv. Como  $p = 0$ , o algoritmo da pseudodivisão é concluído. Dessa forma, tem-se que:

$$x^2y - y = x \cdot (xy - x) + 1 \cdot (x^2 - y).$$

A partir deste ponto, a divisão do polinômio  $f = x^4y^2 - y^2$  por  $g_1 = x^2 - y$  e  $g_2 = xy - x$  será realizada utilizando a ordem lexicográfica graduada  $x \succ_{grlex} y$ , e em seguida, os cálculos serão refeitos invertendo a ordem de  $g_1$  e  $g_2$ , terceiro exemplo apresentado na Subseção 3.4.2 deste trabalho.

Tem-se as seguintes etapas para a divisão de  $f$  por  $g_1$  e  $g_2$ :

- i. Inicialmente, atribui-se  $f$  a  $p$ , os divisores  $g_1$  e  $g_2$  são listados na chave e os quocientes  $a_1$ ,  $a_2$  e o resto  $r$  abaixo da chave. Tem-se o seguinte esquema.

$p : x^4y^2 - y^2$	$g_1 = x^2 - y$
	$g_2 = xy - x$
	$a_1 : 0$
	$a_2 : 0$
	$r : 0$

ii. Observa-se que  $tl(g_1) = x^2$  divide  $tl(p) = x^4y^2$ . Dessa forma,

$$a_1 = a_1 + \frac{tl(p)}{tl(g_1)} = x^2y^2 \text{ e } p = p - \frac{tl(p)}{tl(g_1)}f = x^2y^3 - y^2.$$

$p : x^4y^2 - y^2$	$g_1 = x^2 - y$
$-(x^4y^2 - x^2y^3)$	$g_2 = xy - x$
$x^2y^3 - y^2$	$a_1 : x^2y^2$
	$a_2 : 0$
	$r : 0$

iii. Percebe-se que  $tl(g_1) = x^2$  divide  $tl(p) = x^2y^3$ . Desse modo,

$$a_1 = a_1 + \frac{tl(p)}{tl(g_1)} = x^2y^2 + y^3 \text{ e } p = p - \frac{tl(p)}{tl(g_2)}f = y^4 - y^2.$$

$x^4y^2 - y^2$	$g_1 = x^2 - y$
$-(x^4y^2 - x^2y^3)$	$g_2 = xy - x$
$p : x^2y^3 - y^2$	$a_1 : x^2y^2 + y^3$
$-(x^2y^3 - y^4)$	$a_2 : 0$
$y^4 - y^2$	$r : 0$

iv. Nota-se que  $tl(g_1) = x^2$  não divide  $tl(p) = y^4$ , bem como  $tl(g_2) = xy$  não divide

$tl(p) = y^4$ . Nesse caso, move-se  $tl(p) = y^4$  para o resto  $r$ , ou seja,

$$r = r + tl(p) = y^4 \text{ e } p = p - tl(p) = -y^2.$$

$x^4y^2 - y^2$	$g_1 = x^2 - y$
$-(x^4y^2 - x^2y^3)$	$g_2 = xy - x$
$x^2y^3 - y^2$	$a_1 : x^2y^2 + y^3$
$-(x^2y^3 - y^4)$	$a_2 : 0$
$p : y^4 - y^2$	$r : y^4$

- v. Verifica-se que  $tl(g_1) = x^2$  não divide  $tl(p) = -y^2$ , bem como  $tl(g_2) = xy$  não divide  $tl(p) = -y^2$ . Nesse caso, move-se  $tl(p) = -y^2$  para o resto  $r$ , ou seja,  $r = r + tl(p) = y^4 - y^2$  e  $p = p - tl(p) = 0$ .

$$\begin{array}{r|l}
 x^4y^2 - y^2 & g_1 = x^2 - y \\
 -(x^4y^2 - x^2y^3) & g_2 = xy - x \\
 \hline
 x^2y^3 - y^2 & a_1 : x^2y^2 + y^3 \\
 -(x^2y^3 - y^4) & a_2 : 0 \\
 \hline
 p : \cancel{y^2} & r : y^4 - y^2
 \end{array}$$

- vi. Com  $p = 0$ , o algoritmo da pseudodivisão é concluído. Dessa forma, tem-se que:

$$x^4y^2 - y^2 = (x^2y^2 + y^3) \cdot (x^2 - y) + 0 \cdot (xy - x) + (y^4 - y^2).$$

Partindo dos mesmos polinômios  $f$ ,  $g_1$  e  $g_2$ , os cálculos serão refeitos invertendo a ordem de  $g_1$  e  $g_2$ . Tem-se, então, as seguintes etapas para a divisão de  $f$  por  $g_2$  e  $g_1$ :

- i. Inicialmente, atribui-se  $f$  a  $p$ , os divisores  $g_2$  e  $g_1$  são listados na chave e os quocientes  $a_1$ ,  $a_2$  e o resto  $r$  abaixo da chave. Tem-se o seguinte esquema:

$$\begin{array}{r|l}
 p : x^4y^2 - y^2 & g_2 = xy - x \\
 & g_1 = x^2 - y \\
 \hline
 & a_1 : 0 \\
 & a_2 : 0 \\
 & r : 0
 \end{array}$$

ii. Observa-se que  $tl(g_2) = xy$  divide  $tl(p) = x^4y^2$ . Dessarte,  $a_1 = a_1 + \frac{tl(p)}{tl(g_2)} = x^3y$

$$\text{e } p = p - \frac{tl(p)}{tl(g_2)}f = x^4y - y^2.$$

$p : x^4y^2 - y^2$	$g_2 = xy - x$
$-(x^4y^2 - x^4y)$	$g_1 = x^2 - y$
$x^4y - y^2$	$a_1 : x^3y$
	$a_2 : 0$
	$r : 0$

iii. Nota-se que  $tl(g_2) = xy$  divide  $tl(p) = x^4y$ . Dessa forma,

$$a_1 = a_1 + \frac{tl(p)}{tl(g_2)} = x^3y + x^3 \text{ e } p = p - \frac{tl(p)}{tl(g_2)}f = x^4 - y^2.$$

$x^4y^2 - y^2$	$g_2 = xy - x$
$-(x^4y^2 - x^4y)$	$g_1 = x^2 - y$
$p : x^4y - y^2$	$a_1 : x^3y + x^3$
$-(x^4y - x^4)$	$a_2 : 0$
$x^4 - y^2$	$r : 0$

iv. Verifica-se que  $tl(g_2) = xy$  não divide  $tl(p) = x^4$ , todavia,  $tl(g_1) = x^2$  divide

$$tl(p) = x^4. \text{ Desse modo, } a_2 = a_2 + \frac{tl(p)}{tl(g_1)} = x^2 \text{ e } p = p - \frac{tl(p)}{tl(g_1)} f = x^2y - y^2.$$

$x^4y^2 - y^2$	$g_2 = xy - x$
$-(x^4y^2 - x^4y)$	$g_1 = x^2 - y$
$x^4y - y^2$	$a_1 : x^3y + x^3$
$-(x^4y - x^4)$	$a_2 : x^2$
$p : x^4 - y^2$	$r : 0$
$-(x^4 - x^2y)$	
$x^2y - y^2$	

v. Observa-se que  $tl(g_2) = xy$  divide  $tl(p) = x^2y$ . Por conseguinte,

$$a_1 = a_1 + \frac{tl(p)}{tl(g_2)} = x^3y + x^3 + x \text{ e } p = p - \frac{tl(p)}{tl(g_2)} f = x^2 - y^2.$$

$x^4y^2 - y^2$	$g_2 = xy - x$
$-(x^4y^2 - x^4y)$	$g_1 = x^2 - y$
$x^4y - y^2$	$a_1 : x^3y + x^3 + x$
$-(x^4y - x^4)$	$a_2 : x^2$
$x^4 - y^2$	$r : 0$
$-(x^4 - x^2y)$	
$p : x^2y - y^2$	
$-(x^2y - x^2)$	
$x^2 - y^2$	

vi. Percebe-se que  $tl(g_2) = xy$  não divide  $tl(p) = x^2$ , todavia,  $tl(g_1) = x^2$  divide

$tl(p) = x^2$ . Dessa forma  $a_2 = a_2 + \frac{tl(p)}{tl(g_1)} = 1$  e  $p = p - \frac{tl(p)}{tl(g_1)}f = -y^2 + y$ .

$x^4y^2 - y^2$	$g_2 = xy - x$
$-(x^4y^2 - x^4y)$	$g_1 = x^2 - y$
$x^4y - y^2$	$a_1 : x^3y + x^3 + x$
$-(x^4y - x^4)$	$a_2 : x^2 + 1$
$x^4 - y^2$	$r : 0$
$-(x^4 - x^2y)$	
$x^2y - y^2$	
$-(x^2y - x^2)$	
$p : x^2 - y^2$	
$-(x^2 - y)$	
$-y^2 + y$	



vii. Nota-se que  $tl(g_2) = xy$  não divide  $tl(p) = -y^2$ , bem como  $tl(g_1) = x^2$  não divide  $tl(p) = -y^2$ . Nesse caso, move-se  $tl(p) = -y^2$  para o resto  $r$ , ou seja,  $r = r + tl(p) = -y^2$  e  $p = p - tl(p) = y$ .

$x^4y^2 - y^2$	$g_2 = xy - x$
$-(x^4y^2 - x^4y)$	$g_1 = x^2 - y$
<hr/> $x^4y - y^2$	<hr/> $a_1 : x^3y + x^3 + x$
$-(x^4y - x^4)$	$a_2 : x^2 + 1$
<hr/> $x^4 - y^2$	$r : -y^2$
$-(x^4 - x^2y)$	
<hr/> $x^2y - y^2$	
$-(x^2y - x^2)$	
<hr/> $x^2 - y^2$	
$-(x^2 - y)$	
<hr/> $p : \cancel{y^2} + y$	

- viii. Verifica-se que  $tl(g_2) = xy$  não divide  $tl(p) = y$ , bem como  $tl(g_1) = x^2$  não divide  $tl(p) = y$ . Nesse caso, move-se  $tl(p) = y$  para o resto  $r$ , ou seja,  $r = r + tl(p) = -y^2 + y$  e  $p = p - tl(p) = 0$ .

$x^4y^2 - y^2$	$g_2 = xy - x$
$-(x^4y^2 - x^4y)$	$g_1 = x^2 - y$
$x^4y - y^2$	$a_1 : x^3y + x^3 + x$
$-(x^4y - x^4)$	$a_2 : x^2 + 1$
$x^4 - y^2$	$r : -y^2 + y$
$-(x^4 - x^2y)$	
$x^2y - y^2$	
$-(x^2y - x^2)$	
$x^2 - y^2$	
$-(x^2 - y)$	
$p : \cancel{y}$	

- ix. Como  $p = 0$ , o algoritmo da pseudodivisão é concluído. Portanto, tem-se que:

$$x^4y^2 - y^2 = (x^3y + x^3 + x) \cdot (xy - x) + (x^2 + 1) \cdot (x^2 - y) + y - y^2.$$

## APÊNDICE B

Tomando um exemplo citado por Cox, Little e O'Shea (1992), uma base de Gröbner será produzida para o ideal  $\langle g_1 = x^3 - 2xy, g_2 = x^2y - 2y^2 + x \rangle$  sob a ordem lexicográfica graduada  $x \succ_{grlex} y$  pelo algoritmo de Buchberger. Todas as etapas do algoritmo para obtenção da base serão apresentadas a seguir.

- i. Atribui-se  $G = \{g_1 = x^3 - 2xy, g_2 = x^2y - 2y^2 + x\}$  e  $P = \{(g_1, g_2)\} \neq \emptyset$ .
- ii. Selecionando  $(g_1, g_2)$ , tem-se que  $P = \emptyset$ . Dessa forma, sabendo que  $L = mmc(ml(g_1), ml(g_2)) = mmc(x^3, x^2y) = x^3y$ , determina-se inicialmente o S-polinômio  $S\{g_1, g_2\}$ :

$$\begin{aligned}
 S\{g_1, g_2\} &= \frac{x^3y}{x^3}g_1 - \frac{x^3y}{x^2y}g_2, \\
 S\{g_1, g_2\} &= \frac{x^3y}{x^3}(x^3 - 2xy) - \frac{x^3y}{x^2y}(x^2y - 2y^2 + x), \\
 S\{g_1, g_2\} &= x^3y - 2xy^2 - x^3y + 2xy^2 - x^2, \\
 S\{g_1, g_2\} &= -x^2.
 \end{aligned} \tag{382}$$

$S\{g_1, g_2\} = -x^2$  é dividido pelo par de polinômios  $g_1$  e  $g_2$ . Dessarte, tem-se a seguinte pseudodivisão:

$$\begin{array}{r|l}
 -x^2 & g_1 = x^3 - 2xy \\
 & g_2 = x^2y - 2y^2 + x \\
 0 & \hline
 & a_1 : 0 \\
 & a_2 : 0 \\
 & r : -x^2
 \end{array}$$

Como  $g_3 = r = -x^2 \neq 0$ , então  $G = \{g_1 = x^3 - 2xy, g_2 = x^2y - 2y^2 + x, g_3 = -x^2\}$  e

$$P = \{(g_1, g_3), (g_2, g_3)\} \neq \emptyset.$$

iii. Selecionando  $(g_2, g_3)$ , tem-se que  $P = \{(g_1, g_3)\} \neq \emptyset$ . Desse modo, sabendo que  $L = mmc(ml(g_2), ml(g_3)) = mmc(x^2y, x^2) = x^2y$ , determina-se inicialmente o S-polinômio  $S\{g_2, g_3\}$ :

$$\begin{aligned} S\{g_2, g_3\} &= \frac{x^2y}{x^2y}g_2 - \frac{x^2y}{-x^2}g_3, \\ S\{g_2, g_3\} &= \frac{x^2y}{x^2y}(x^2y - 2y^2 + x) - \frac{x^2y}{-x^2}(-x^2), \\ S\{g_2, g_3\} &= x^2y - 2y^2 + x - x^2y, \\ S\{g_2, g_3\} &= -2y^2 + x. \end{aligned} \tag{383}$$

$S\{g_2, g_3\} = -2y^2 + x$  é dividido pelos polinômios  $g_1$ ,  $g_2$  e  $g_3$ . Dessa forma, tem-se a seguinte pseudodivisão:

$$\begin{array}{l|l} & g_1 = x^3 - 2xy \\ & g_2 = x^2y - 2y^2 + x \\ -2y^2 + x & g_3 = -x^2 \\ \hline x & a_1 : 0 \\ 0 & a_2 : 0 \\ & a_3 : 0 \\ & r : -2y^2 + x \end{array}$$

Como  $g_4 = r = -2y^2 + x \neq 0$ , então  $G = \{g_1, g_2, g_3, g_4 = -2y^2 + x\}$  e  $P = \{(g_1, g_3), (g_1, g_4), (g_2, g_4), (g_3, g_4)\} \neq \emptyset$ .

iv. Selecionando  $(g_3, g_4)$ , tem-se que  $P = \{(g_1, g_3), (g_1, g_4), (g_2, g_4)\} \neq \emptyset$ . Dessarte, sabendo que  $L = mmc(ml(g_3), ml(g_4)) = mmc(x^2, y^2) = x^2y^2$ , determina-se inicialmente o S-polinômio  $S\{g_3, g_4\}$ :

$$\begin{aligned}
 S\{g_3, g_4\} &= \frac{x^2 y^2}{-x^2} g_3 - \frac{x^2 y^2}{-2y^2} g_4, \\
 S\{g_3, g_4\} &= \frac{x^2 y^2}{-x^2} (-x^2) - \frac{x^2 y^2}{-2y^2} (-2y^2 + x), \\
 S\{g_3, g_4\} &= x^2 y^2 - x^2 y^2 + \frac{x^3}{2}, \\
 S\{g_3, g_4\} &= \frac{x^3}{2}.
 \end{aligned} \tag{384}$$

$S\{g_3, g_4\} = \frac{x^3}{2}$  é dividido pelos polinômios  $g_1, g_2, g_3$  e  $g_4$ . Por conseguinte, tem-se a seguinte pseudodivisão:

$\frac{x^3}{2}$	$g_1 = x^3 - 2xy$
$-\left(\frac{x^3}{2} - xy\right)$	$g_2 = x^2 y - 2y^2 + x$
<hr style="width: 100%;"/>	$g_3 = -x^2$
$xy$	$g_4 = -2y^2 + x$
<hr style="width: 100%;"/>	$a_1 : \frac{1}{2}$
	$a_2 : 0$
	$a_3 : 0$
	$a_4 : 0$
	$r : xy$

Como  $g_5 = r = xy \neq 0$ , então  $G = \{g_1, g_2, g_3, g_4, g_5 = xy\}$  e

$$P = \{(g_1, g_3), (g_1, g_4), (g_1, g_5), (g_2, g_4), (g_2, g_5), (g_3, g_5), (g_4, g_5)\} \neq \emptyset.$$

v. Selecionando  $(g_4, g_5)$ , tem-se que

$$P = \{(g_1, g_3), (g_1, g_4), (g_1, g_5), (g_2, g_4), (g_2, g_5), (g_3, g_5)\} \neq \emptyset. \text{ Portanto, sabendo que}$$

$$L = mmc(ml(g_4), ml(g_5)) = mmc(y^2, xy) = xy^2, \text{ determina-se inicialmente o S-}$$

polinômio  $S\{g_4, g_5\}$ :

$$\begin{aligned}
 S\{g_4, g_5\} &= \frac{xy^2}{-2y^2} g_4 - \frac{xy^2}{-2xy} g_5, \\
 S\{g_4, g_5\} &= \frac{xy^2}{-2y^2} (-2y^2 + x) - \frac{xy^2}{xy} (xy), \\
 S\{g_4, g_5\} &= xy^2 - \frac{x^2}{2} - xy^2, \\
 S\{g_4, g_5\} &= -\frac{x^2}{2}.
 \end{aligned} \tag{385}$$

$S\{g_4, g_5\} = -\frac{x^2}{2}$  é dividido pelos polinômios  $g_1, g_2, g_3, g_4$  e  $g_5$ . Desse modo,

tem-se a seguinte pseudodivisão:

$g_1 = x^3 - 2xy$	
$g_2 = x^2y - 2y^2 + x$	
$g_3 = -x^2$	
$g_4 = -2y^2 + x$	
$-\frac{x^2}{2}$	
$g_5 = xy$	
$-\left(-\frac{x^2}{2}\right)$	
$0$	$a_1 : 0$
	$a_2 : 0$
	$a_3 : \frac{1}{2}$
	$a_4 : 0$
	$a_5 : 0$
	$r : 0$

Como  $r = 0$ , então  $G = \{g_1, g_2, g_3, g_4, g_5\}$  e

$$P = \{(g_1, g_3), (g_1, g_4), (g_1, g_5), (g_2, g_4), (g_2, g_5), (g_3, g_5)\} \neq \emptyset.$$

vi. Selecionando  $(g_3, g_5)$ , tem-se que

$$P = \{(g_1, g_3), (g_1, g_4), (g_1, g_5), (g_2, g_4), (g_2, g_5)\} \neq \emptyset. \quad \text{Portanto, sabendo que}$$

$L = mmc(ml(g_3), ml(g_5)) = mmc(x^2, xy) = x^2y$ , determina-se inicialmente o S-polinômio  $S\{g_3, g_5\}$ :

$$\begin{aligned} S\{g_3, g_5\} &= \frac{x^2y}{-x^2}g_3 - \frac{x^2y}{xy}g_5, \\ S\{g_3, g_5\} &= \frac{x^2y}{-x^2}(-x^2) - \frac{x^2y}{xy}(xy), \\ S\{g_3, g_5\} &= x^2y - x^2y, \\ S\{g_3, g_5\} &= 0. \end{aligned} \tag{386}$$

$S\{g_1, g_4\} = 0$  é dividido pelos polinômios  $g_1, g_2, g_3, g_4$  e  $g_5$ . Por conseguinte, tem-se a seguinte pseudodivisão:

$$\begin{array}{l|l} & g_1 = x^3 - 2xy \\ & g_2 = x^2y - 2y^2 + x \\ & g_3 = -x^2 \\ & g_4 = -2y^2 + x \\ 0 & g_5 = xy \\ & \hline & a_1 : 0 \\ & a_2 : 0 \\ & a_3 : 0 \\ & a_4 : 0 \\ & a_5 : 0 \\ & r : 0 \end{array}$$

Como  $r = 0$ , então  $G = \{g_1, g_2, g_3, g_4, g_5\}$  e

$$P = \{(g_1, g_3), (g_1, g_4), (g_1, g_5), (g_2, g_4), (g_2, g_5)\} \neq \emptyset.$$

vii. Seleccionando  $(g_2, g_5)$ , tem-se que

$$P = \{(g_1, g_3), (g_1, g_4), (g_1, g_5), (g_2, g_4), (g_2, g_5)\} \neq \emptyset. \text{ Desse modo, sabendo que}$$

$L = mmc(ml(g_2), ml(g_5)) = mmc(x^2y, xy) = x^2y$ , determina-se inicialmente o S-polinômio  $S\{g_2, g_5\}$ :

$$\begin{aligned} S\{g_2, g_5\} &= \frac{x^2y}{x^2y}g_2 - \frac{x^2y}{xy}g_5, \\ S\{g_2, g_5\} &= \frac{x^2y}{x^2y}(x^2y - 2y^2 + x) - \frac{x^2y}{xy}(xy), \\ S\{g_2, g_5\} &= x^2y - 2y^2 + x - x^2y, \\ S\{g_2, g_5\} &= -2y^2 + x. \end{aligned} \tag{387}$$

$S\{g_2, g_5\} = -2y^2 + x$  é dividido pelos polinômios  $g_1, g_2, g_3, g_4$  e  $g_5$ . Dessa forma, tem-se a seguinte pseudodivisão:

	$g_1 = x^3 - 2xy$
	$g_2 = x^2y - 2y^2 + x$
	$g_3 = -x^2$
	$g_4 = -2y^2 + x$
$-2y^2 + x$	$g_5 = xy$
$-(-2y^2 + x)$	$a_1 : 0$
$0$	$a_2 : 0$
	$a_3 : 0$
	$a_4 : 1$
	$a_5 : 0$
	$r : 0$

Como  $r = 0$ , então  $G = \{g_1, g_2, g_3, g_4, g_5\}$  e

$$P = \{(g_1, g_3), (g_1, g_4), (g_1, g_5), (g_2, g_4)\} \neq \emptyset.$$



viii. Seleccionando  $(g_2, g_4)$ , tem-se que  $P = \{(g_1, g_3), (g_1, g_4), (g_1, g_5)\} \neq \emptyset$ . Portanto, sabendo que  $L = mmc(ml(g_2), ml(g_4)) = mmc(x^2y, y^2) = x^2y^2$ , determina-se inicialmente o S-polinômio  $S\{g_2, g_4\}$ :

$$\begin{aligned}
 S\{g_2, g_4\} &= \frac{x^2y^2}{x^2y}g_2 - \frac{x^2y^2}{-2y^2}g_4, \\
 S\{g_2, g_4\} &= \frac{x^2y^2}{x^2y}(x^2y - 2y^2 + x) - \frac{x^2y^2}{-2y^2}(-2y^2 + x), \\
 S\{g_2, g_4\} &= x^2y^2 - 2y^3 + xy - x^2y^2 + \frac{x^3}{2}, \\
 S\{g_2, g_4\} &= \frac{x^3}{2} - 2y^3 + xy.
 \end{aligned} \tag{388}$$

$S\{g_2, g_4\} = \frac{x^3}{2} - 2y^3 + xy$  é dividido pelos polinômios  $g_1, g_2, g_3, g_4$  e  $g_5$ . Por

consequente, tem-se a seguinte pseudodivisão:

$\frac{x^3}{2} - 2y^3 + xy$	$g_1 = x^3 - 2xy$
$-\left(\frac{x^3}{2} - xy\right)$	$g_2 = x^2y - 2y^2 + x$
$-2y^3 + 2xy$	$g_3 = -x^2$
$-(-2y^3 + xy)$	$g_4 = -2y^2 + x$
$xy$	$g_5 = xy$
$-(xy)$	$a_1 : \frac{1}{2}$
$0$	$a_2 : 0$
	$a_3 : 0$
	$a_4 : y$
	$a_5 : 1$
	$r : 0$

Como  $r = 0$ , então  $G = \{g_1, g_2, g_3, g_4, g_5\}$  e  $P = \{(g_1, g_3), (g_1, g_4), (g_1, g_5)\} \neq \emptyset$ .

ix. Selecionando  $(g_1, g_5)$ , tem-se que  $P = \{(g_1, g_3), (g_1, g_4)\} \neq \emptyset$ . Dessarte, sabendo que  $L = mmc(ml(g_1), ml(g_5)) = mmc(x^3, xy) = x^3y$ , determina-se inicialmente o S-polinômio  $S\{g_1, g_5\}$ :

$$\begin{aligned}
 S\{g_1, g_5\} &= \frac{x^3y}{x^3} g_1 - \frac{x^3y}{-2xy} g_5, \\
 S\{g_1, g_5\} &= \frac{x^3y}{x^3} (x^3 - 2xy) - \frac{x^3y}{xy} (xy), \\
 S\{g_1, g_5\} &= x^3y - 2xy^2 - x^3y, \\
 S\{g_1, g_5\} &= -2xy^2.
 \end{aligned} \tag{389}$$

$S\{g_1, g_5\} = -2xy^2$  é dividido pelos polinômios  $g_1, g_2, g_3, g_4$  e  $g_5$ . Dessa forma, tem-se a seguinte pseudodivisão:

$g_1 = x^3 - 2xy$	
$g_2 = x^2y - 2y^2 + x$	
$g_3 = -x^2$	
$g_4 = -2y^2 + x$	
$-2xy^2$	$g_5 = xy$
$-( -2xy^2 + x^2 )$	$a_1 : 0$
$-x^2$	$a_2 : 0$
$-(-x^2)$	$a_3 : 1$
$0$	$a_4 : x$
	$a_5 : 0$
	$r : 0$

Como  $r = 0$ , então  $G = \{g_1, g_2, g_3, g_4, g_5\}$  e  $P = \{(g_1, g_3), (g_1, g_4)\} \neq \emptyset$ .

x. Selecionando  $(g_1, g_4)$ , tem-se que  $P = \{(g_1, g_3)\} \neq \emptyset$ . Portanto, sabendo que  $L = mmc(ml(g_1), ml(g_4)) = mmc(x^3, y^2) = x^3y^2$ , determina-se inicialmente o S-polinômio  $S\{g_1, g_4\}$ :

$$\begin{aligned}
 S\{g_1, g_4\} &= \frac{x^3 y^2}{x^3} g_1 - \frac{x^3 y^2}{-2y^2} g_4, \\
 S\{g_1, g_4\} &= \frac{x^3 y^2}{x^3} (x^3 - 2xy) - \frac{x^3 y^2}{-2y^2} (-2y^2 + x), \\
 S\{g_1, g_4\} &= x^3 y^2 - 2xy^3 - x^3 y^2 + \frac{x^4}{2}, \\
 S\{g_1, g_4\} &= \frac{x^4}{2} - 2xy^3.
 \end{aligned} \tag{390}$$

$S\{g_1, g_4\} = \frac{x^4}{2} - 2xy^3$  é dividido pelos polinômios  $g_1, g_2, g_3, g_4$  e  $g_5$ . Por conseguinte, tem-se a seguinte pseudodivisão:

$\frac{x^4}{2} - 2xy^3$	$g_1 = x^3 - 2xy$
$-\left(\frac{x^4}{2} - x^2 y\right)$	$g_2 = x^2 y - 2y^2 + x$
$x^2 y - 2xy^3$	$g_3 = -x^2$
$-(x^2 y - 2y^2 + x)$	$g_4 = -2y^2 + x$
$-2xy^3 + 2y^2 - x$	$g_5 = xy$
$-\left(-2xy^3 + x^2 y\right)$	$a_1 : \frac{x}{2}$
$-x^2 y + 2y^2 - x$	$a_2 : 1 - 1 = 0$
$-\left(-x^2 y + 2y^2 - x\right)$	$a_3 : 0$
$0$	$a_4 : xy$
	$a_5 : 0$
	$r : 0$

Como  $r = 0$ , então  $G = \{g_1, g_2, g_3, g_4, g_5\}$  e  $P = \{(g_1, g_3)\} \neq \emptyset$ .

xi. Seleccionando  $(g_1, g_3)$ , tem-se que  $P = \emptyset$ . Desse modo, sabendo que  $L = mmc(ml(g_1), ml(g_3)) = mmc(x^3, x^2) = x^3$ , determina-se inicialmente o S-polinômio  $S\{g_1, g_3\}$ :

$$\begin{aligned} S\{g_1, g_3\} &= \frac{x^3}{x^3} g_1 - \frac{x^3}{-x^2} g_3, \\ S\{g_1, g_3\} &= \frac{x^3}{x^3} (x^3 - 2xy) - \frac{x^3}{-x^2} (-x^2), \\ S\{g_1, g_3\} &= x^3 - 2xy - x^3, \\ S\{g_1, g_3\} &= -2xy. \end{aligned} \tag{391}$$

$S\{g_1, g_3\} = -2xy$  é dividido pelos polinômios  $g_1, g_2, g_3, g_4$  e  $g_5$ . Dessarte, tem-se a seguinte pseudodivisão:

	$g_1 = x^3 - 2xy$
	$g_2 = x^2y - 2y^2 + x$
	$g_3 = -x^2$
	$g_4 = -2y^2 + x$
$-2xy$	$g_5 = xy$
$-(-2xy)$	$a_1 : 0$
$0$	$a_2 : 0$
	$a_3 : 0$
	$a_4 : 0$
	$a_5 : -2$
	$r : 0$

Como  $r = 0$  e  $P = \emptyset$ , então  $G = \{g_1, g_2, g_3, g_4, g_5\}$  e o algoritmo de Buchberger é concluído.

Destarte, segundo o algoritmo de Buchberger, a base de Gröbner do ideal  $\langle g_1 = x^3 - 2xy, g_2 = x^2y - 2y^2 + x \rangle$  sob a ordem lexicográfica graduada  $x \succ_{grlex} y$  será  $G = \{x^3 - 2xy, x^2y - 2y^2 + x, -x^2, -2y^2 + x, xy\}$ .

## APÊNDICE C

A função  $\text{Atan2}(y, x)$  ou  $\text{Arctan2}(y, x)$  é definida como o ângulo no plano Euclidiano, dado em radianos, entre o eixo  $x$  positivo e o raio até o ponto  $(x; y) \neq (0; 0)$ . Basicamente, ao utilizar a função  $\text{Atan2}$ , pretende-se retornar um valor correto e inequívoco para o ângulo  $\theta$  na conversão de coordenadas cartesianas  $(x; y)$  em coordenadas polares  $(r; \theta)$ .

A função arco-tangente de argumento único não pode distinguir entre direções diametralmente opostas. Por exemplo, o ângulo anti-horário do eixo  $x$  para o vetor  $(1; 1)$ , calculado da maneira usual como  $\arctan\left(\frac{1}{1}\right)$ , é  $\frac{\pi}{4}$  (radianos), ou  $45^\circ$ . No entanto, o ângulo entre o eixo  $x$  e o vetor  $(-1; -1)$  parece, pelo mesmo método, ser  $\arctan\left(\frac{-1}{-1}\right)$ , novamente  $\frac{\pi}{4}$ , mesmo que se possa esperar as respostas  $-\frac{3\pi}{4}$  ( $-135^\circ$ ) ou  $\frac{5\pi}{4}$  ( $225^\circ$ ). Além disso, uma tentativa de encontrar o ângulo entre o eixo  $x$  e os vetores  $(0; y)$ , com  $y \neq 0$ , requer avaliação do  $\arctan(y, 0)$ , que falha na divisão por zero.

A função  $\text{Atan2}$  calcula um valor único da tangente do arco por meio de duas variáveis  $y$  e  $x$ , em que os sinais de ambos os argumentos são usados para determinar o quadrante do resultado, selecionando assim o ramo desejado da tangente do arco de  $\frac{y}{x}$ , dessa forma, tem-se:

$$\text{Atan2}(y, x) = \begin{cases} \arctan\left(\frac{y}{x}\right) & \text{se } x > 0 \\ \arctan\left(\frac{y}{x}\right) + \pi & \text{se } x < 0 \text{ e } y \geq 0 \\ \arctan\left(\frac{y}{x}\right) - \pi & \text{se } x < 0 \text{ e } y < 0 \\ +\frac{\pi}{2} & \text{se } x = 0 \text{ e } y > 0 \\ -\frac{\pi}{2} & \text{se } x = 0 \text{ e } y < 0 \\ \text{indefinido} & \text{se } x = 0 \text{ e } y = 0 \end{cases}$$

Os códigos a seguir, implementados no MAPLE, são utilizados (a título de ilustração) para diferenciar as funções  $\arctan$  e  $\text{Atan2}$ . As sintaxes para as duas funções são  $(\arctan(y/x))$  e  $(\arctan(y,x))$ , para  $\arctan$  e  $\text{Atan2}$ , respectivamente.

```
> #####
> # Entradas (x>0)
> y := 1;
> x := 2;
> alpha1 := arctan(y/x);
> alpha2 := arctan(y,x);
> # Saídas
> alpha1 := arctan(1/2)
> alpha2 := arctan(1/2)
> #####

> #####
> # Entradas (x<0 e y<0)
> y := -1;
> x := -2;
> alpha1 := arctan(y/x);
> alpha2 := arctan(y,x);
> # Saídas
> alpha1 := arctan(1/2)
> alpha2 := arctan(1/2)-Pi
> #####
```

```
> #####
> # Entradas (x<0 e y>=0)
> y := 1;
> x := -2;
> alpha1 := arctan(y/x);
> alpha2 := arctan(y,x);
> # Saídas
> alpha1 := -arctan(1/2)
> alpha2 := -arctan(1/2)+Pi
> #####

> #####
> # Entradas (x=0 e y>0)
> y := 2;
> x := 0;
> alpha1 := arctan(y/x);
> alpha2 := arctan(y,x);
> # Saídas
> Error, numeric exception: division by zero
> alpha2 := (1/2)*Pi
> #####

> #####
> # Entradas (x=0 e y<0)
> y := -2;
> x := 0;
> alpha1 := arctan(y/x);
> alpha2 := arctan(y,x);
> # Saídas
> Error, numeric exception: division by zero
> alpha2 := -(1/2)*Pi
> #####

> #####
> # Entradas (x=0 e y=0)
> y := 2;
> x := 0;
> alpha1 := arctan(y/x);
> alpha2 := arctan(y,x);
> # Saídas
> Error, numeric exception: division by zero
> alpha2 := 0
> #####
```



## APÊNDICE D

```
> #####
> # Solução da Cinemática Inversa do PUMA 560 (Método de Paul)- 1° Caso
> #####
> with(codegen, cost);
> with(linalg);
> with(Units[Natural]);
> #####
> #Entradas
> #####
> c1_1 := cos(theta1_1);
> c1_2 := cos(theta1_2);
> c3_1 := cos(theta3_1);
> c3_2 := cos(theta3_2);
> c2_1 := cos(theta2_1);
> c2_2 := cos(theta2_2);
> c2_3 := cos(theta2_3);
> c2_4 := cos(theta2_4);
> c4_1 := cos(theta4_1);
> c4_2 := cos(theta4_2);
> c4_3 := cos(theta4_3);
> c4_4 := cos(theta4_4);
> c4_5 := cos(theta4_5);
> c4_6 := cos(theta4_6);
> c4_7 := cos(theta4_7);
> c4_8 := cos(theta4_8);
> c5_1 := cos(theta5_1);
> c5_2 := cos(theta5_2);
> c5_3 := cos(theta5_3);
> c5_4 := cos(theta5_4);
> c5_5 := cos(theta5_5);
> c5_6 := cos(theta5_6);
> c5_7 := cos(theta5_7);
> c5_8 := cos(theta5_8);
> c6_1 := cos(theta6_1);
> c6_2 := cos(theta6_2);
> c6_3 := cos(theta6_3);
> c6_4 := cos(theta6_4);
> c6_5 := cos(theta6_5);
> c6_6 := cos(theta6_6);
> c6_7 := cos(theta6_7);
> c6_8 := cos(theta6_8);
> s1_1 := sin(theta1_1);
> s1_2 := sin(theta1_2);
> s3_1 := sin(theta3_1);
> s3_2 := sin(theta3_2);
> s2_1 := sin(theta2_1);
> s2_2 := sin(theta2_2);
> s2_3 := sin(theta2_3);
> s2_4 := sin(theta2_4);
> s4_1 := sin(theta4_1);
```

```
> s4_2 := sin(theta4_2);
> s4_3 := sin(theta4_3);
> s4_4 := sin(theta4_4);
> s4_5 := sin(theta4_5);
> s4_6 := sin(theta4_6);
> s4_7 := sin(theta4_7);
> s4_8 := sin(theta4_8);
> s5_1 := sin(theta5_1);
> s5_2 := sin(theta5_2);
> s5_3 := sin(theta5_3);
> s5_4 := sin(theta5_4);
> s5_5 := sin(theta5_5);
> s5_6 := sin(theta5_6);
> s5_7 := sin(theta5_7);
> s5_8 := sin(theta5_8);
> s6_1 := sin(theta6_1);
> s6_2 := sin(theta6_2);
> s6_3 := sin(theta6_3);
> s6_4 := sin(theta6_4);
> s6_5 := sin(theta6_5);
> s6_6 := sin(theta6_6);
> s6_7 := sin(theta6_7);
> s6_8 := sin(theta6_8);
> c2_13_1 := cos(theta2_1+theta3_1);
> s2_13_1 := sin(theta2_1+theta3_1);
> c2_23_1 := cos(theta2_2+theta3_1);
> s2_23_1 := sin(theta2_2+theta3_1);
> c2_33_2 := cos(theta2_3+theta3_2);
> s2_33_2 := sin(theta2_3+theta3_2);
> c2_43_2 := cos(theta2_4+theta3_2);
> s2_43_2 := sin(theta2_4+theta3_2);
> a2 := 432;
> d3 := 125;
> a3 := 20;
> d4 := 430;
> nx := 1;
> ny := 0;
> nz := 0;
> ox := 0;
> oy := -1;
> oz := 0;
> ax := 0;
> ay := 0;
> az := -1;
> x := 452;
> y := 125;
> z := -430;
> theta1_1 := arctan(y, x)-arctan(d3, sqrt(-d3^2+x^2+y^2));
> theta1_2 := arctan(y, x)-arctan(d3, -sqrt(-d3^2+x^2+y^2));
> theta3_1 := arctan(a3, d4)-arctan((-a2^2-a3^2-d3^2-
d4^2+x^2+y^2+z^2)/(2*a2), sqrt(a3^2+d4^2-((-a2^2-a3^2-d3^2-
d4^2+x^2+y^2+z^2)/(2*a2))^2));
> theta3_2 := arctan(a3, d4)-arctan((-a2^2-a3^2-d3^2-
d4^2+x^2+y^2+z^2)/(2*a2), -sqrt(a3^2+d4^2-((-a2^2-a3^2-d3^2-
d4^2+x^2+y^2+z^2)/(2*a2))^2));
```

```

> theta2_1 := arctan((-a2*c3_1-a3)*z-(c1_1*x+s1_1*y)*(-a2*s3_1+d4),
(a2*s3_1-d4)*z-(a2*c3_1+a3)*(c1_1*x+s1_1*y))-theta3_1;
> theta2_2 := arctan((-a2*c3_1-a3)*z-(c1_2*x+s1_2*y)*(-a2*s3_1+d4),
(a2*s3_1-d4)*z-(a2*c3_1+a3)*(c1_2*x+s1_2*y))-theta3_1;
> theta2_3 := arctan((-a2*c3_2-a3)*z-(c1_1*x+s1_1*y)*(-a2*s3_2+d4),
(a2*s3_2-d4)*z-(a2*c3_2+a3)*(c1_1*x+s1_1*y))-theta3_2;
> theta2_4 := arctan((-a2*c3_2-a3)*z-(c1_2*x+s1_2*y)*(-a2*s3_2+d4),
(a2*s3_2-d4)*z-(a2*c3_2+a3)*(c1_2*x+s1_2*y))-theta3_2;
> theta4_1 := arctan(-ax*s1_1+ay*c1_1, -ax*c1_1*c2_13_1-
ay*c2_13_1*s1_1+az*s2_13_1);
> theta4_2 := arctan(-ax*s1_2+ay*c1_2, -ax*c1_2*c2_23_1-
ay*c2_23_1*s1_2+az*s2_23_1);
> theta4_3 := arctan(-ax*s1_1+ay*c1_1, -ax*c1_1*c2_33_2-
ay*c2_33_2*s1_1+az*s2_33_2);
> theta4_4 := arctan(-ax*s1_2+ay*c1_2, -ax*c1_2*c2_43_2-
ay*c2_43_2*s1_2+az*s2_43_2);
> theta4_5 := theta4_1+Pi;
> theta4_6 := theta4_2+Pi;
> theta4_7 := theta4_3+Pi;
> theta4_8 := theta4_4+Pi;
> theta5_1 := arctan(-ax*(c1_1*c2_13_1*c4_1+s1_1*s4_1)-
ay*(c2_13_1*c4_1*s1_1-c1_1*s4_1)+az*s2_13_1*c4_1, -ax*c1_1*s2_13_1-
ay*s1_1*s2_13_1-az*c2_13_1);
> theta5_2 := arctan(-ax*(c1_2*c2_23_1*c4_2+s1_2*s4_2)-
ay*(c2_23_1*c4_2*s1_2-c1_2*s4_2)+az*s2_23_1*c4_2, -ax*c1_2*s2_23_1-
ay*s1_2*s2_23_1-az*c2_23_1);
> theta5_3 := arctan(-ax*(c1_1*c2_33_2*c4_3+s1_1*s4_3)-
ay*(c2_33_2*c4_3*s1_1-c1_1*s4_3)+az*s2_33_2*c4_3, -ax*c1_1*s2_33_2-
ay*s1_1*s2_33_2-az*c2_33_2);
> theta5_4 := arctan(-ax*(c1_2*c2_43_2*c4_4+s1_2*s4_4)-
ay*(c2_43_2*c4_4*s1_2-c1_2*s4_4)+az*s2_43_2*c4_4, -ax*c1_2*s2_43_2-
ay*s1_2*s2_43_2-az*c2_43_2);
> theta5_5 := -theta5_1;
> theta5_6 := -theta5_2;
> theta5_7 := -theta5_3;
> theta5_8 := -theta5_4;
> theta6_1 := arctan(-nx*(c1_1*c2_13_1*s4_1-c4_1*s1_1)-
ny*(c2_13_1*s1_1*s4_1+c1_1*c4_1)+nz*s2_13_1*s4_1,
nx*((c1_1*c2_13_1*c4_1+s1_1*s4_1)*c5_1-
c1_1*s2_13_1*s5_1)+ny*((c2_13_1*c4_1*s1_1-c1_1*s4_1)*c5_1-
s1_1*s2_13_1*s5_1)-nz*(c2_13_1*c4_1*s1_1*s5_1+c4_1*c5_1*s2_13_1));
> theta6_2 := arctan(-nx*(c1_2*c2_23_1*s4_2-c4_2*s1_2)-
ny*(c2_23_1*s1_2*s4_2+c1_2*c4_2)+nz*s2_23_1*s4_2,
nx*((c1_2*c2_23_1*c4_2+s1_2*s4_2)*c5_2-
c1_2*s2_23_1*s5_2)+ny*((c2_23_1*c4_2*s1_2-c1_2*s4_2)*c5_2-
s1_2*s2_23_1*s5_2)-nz*(c2_23_1*c4_2*s1_2*s5_2+c4_2*c5_2*s2_23_1));
> theta6_3 := arctan(-nx*(c1_1*c2_33_2*s4_3-c4_3*s1_1)-
ny*(c2_33_2*s1_1*s4_3+c1_1*c4_3)+nz*s2_33_2*s4_3,
nx*((c1_1*c2_33_2*c4_3+s1_1*s4_3)*c5_3-
c1_1*s2_33_2*s5_3)+ny*((c2_33_2*c4_3*s1_1-c1_1*s4_3)*c5_3-
s1_1*s2_33_2*s5_3)-nz*(c2_33_2*c4_3*s1_1*s5_3+c4_3*c5_3*s2_33_2));
> theta6_4 := arctan(-nx*(c1_2*c2_43_2*s4_4-c4_4*s1_2)-
ny*(c2_43_2*s1_2*s4_4+c1_2*c4_4)+nz*s2_43_2*s4_4,
nx*((c1_2*c2_43_2*c4_4+s1_2*s4_4)*c5_4-
c1_2*s2_43_2*s5_4)+ny*((c2_43_2*c4_4*s1_2-c1_2*s4_4)*c5_4-
s1_2*s2_43_2*s5_4)-nz*(c2_43_2*c4_4*s1_2*s5_4+c4_4*c5_4*s2_43_2));
> theta6_5 := theta6_1+Pi;

```

```
> theta6_6 := theta6_2+Pi;
> theta6_7 := theta6_3+Pi;
> theta6_8 := theta6_4+Pi;
> cost_1_1 := cost(theta1_1);
> cost_1_2 := cost(theta1_2);
> cost_3_1 := cost(theta3_1);
> cost_3_2 := cost(theta3_2);
> cost_2_1 := cost(theta2_1);
> cost_2_2 := cost(theta2_2);
> cost_2_3 := cost(theta2_3);
> cost_2_4 := cost(theta2_4);
> cost_4_1 := cost(theta4_1);
> cost_4_2 := cost(theta4_2);
> cost_4_3 := cost(theta4_3);
> cost_4_4 := cost(theta4_4);
> cost_4_5 := cost(theta4_5);
> cost_4_6 := cost(theta4_6);
> cost_4_7 := cost(theta4_7);
> cost_4_8 := cost(theta4_8);
> cost_5_1 := cost(theta5_1);
> cost_5_2 := cost(theta5_2);
> cost_5_3 := cost(theta5_3);
> cost_5_4 := cost(theta5_4);
> cost_5_5 := cost(theta5_5);
> cost_5_6 := cost(theta5_6);
> cost_5_7 := cost(theta5_7);
> cost_5_8 := cost(theta5_8);
> cost_6_1 := cost(theta6_1);
> cost_6_2 := cost(theta6_2);
> cost_6_3 := cost(theta6_3);
> cost_6_4 := cost(theta6_4);
> cost_6_5 := cost(theta6_5);
> cost_6_6 := cost(theta6_6);
> cost_6_7 := cost(theta6_7);
> cost_6_8 := cost(theta6_8);
> V := {additions = 1, divisions = 4, functions = 8, multiplications = 1};
> flop_theta1_1 := (eval(cost(theta1_1), V))*flop;
> flop_theta1_2 := (eval(cost(theta1_2), V))*flop;
> flop_theta3_1 := (eval(cost(theta3_1), V))*flop;
> flop_theta3_2 := (eval(cost(theta3_2), V))*flop;
> flop_theta2_1 := (eval(cost(theta2_1), V))*flop;
> flop_theta2_2 := (eval(cost(theta2_2), V))*flop;
> flop_theta2_3 := (eval(cost(theta2_3), V))*flop;
> flop_theta2_4 := (eval(cost(theta2_4), V))*flop;
> flop_theta4_1 := (eval(cost(theta4_1), V))*flop;
> flop_theta4_2 := (eval(cost(theta4_2), V))*flop;
> flop_theta4_3 := (eval(cost(theta4_3), V))*flop;
> flop_theta4_4 := (eval(cost(theta4_4), V))*flop;
> flop_theta4_5 := (eval(cost(theta4_5), V))*flop;
> flop_theta4_6 := (eval(cost(theta4_6), V))*flop;
> flop_theta4_7 := (eval(cost(theta4_7), V))*flop;
> flop_theta4_8 := (eval(cost(theta4_8), V))*flop;
> flop_theta5_1 := (eval(cost(theta5_1), V))*flop;
> flop_theta5_2 := (eval(cost(theta5_2), V))*flop;
> flop_theta5_3 := (eval(cost(theta5_3), V))*flop;
```

```
> flop_theta5_4 := (eval(cost(theta5_4), V))*flop;
> flop_theta5_5 := (eval(cost(theta5_5), V))*flop;
> flop_theta5_6 := (eval(cost(theta5_6), V))*flop;
> flop_theta5_7 := (eval(cost(theta5_7), V))*flop;
> flop_theta5_8 := (eval(cost(theta5_8), V))*flop;
> flop_theta6_1 := (eval(cost(theta6_1), V))*flop;
> flop_theta6_2 := (eval(cost(theta6_2), V))*flop;
> flop_theta6_3 := (eval(cost(theta6_3), V))*flop;
> flop_theta6_4 := (eval(cost(theta6_4), V))*flop;
> flop_theta6_5 := (eval(cost(theta6_5), V))*flop;
> flop_theta6_6 := (eval(cost(theta6_6), V))*flop;
> flop_theta6_7 := (eval(cost(theta6_7), V))*flop;
> flop_theta6_8 := (eval(cost(theta6_8), V))*flop;
> total_cost_config_1 := (eval(cost(theta1_1),
V))*flop+(eval(cost(theta3_1), V))*flop+(eval(cost(theta2_1),
V))*flop+(eval(cost(theta4_1), V))*flop+(eval(cost(theta5_1),
V))*flop+(eval(cost(theta6_1), V))*flop;
> total_cost_config_2 := (eval(cost(theta1_1),
V))*flop+(eval(cost(theta3_1), V))*flop+(eval(cost(theta2_1),
V))*flop+(eval(cost(theta4_5), V))*flop+(eval(cost(theta5_5),
V))*flop+(eval(cost(theta6_5), V))*flop;
> total_cost_config_3 := (eval(cost(theta1_1),
V))*flop+(eval(cost(theta3_2), V))*flop+(eval(cost(theta2_3),
V))*flop+(eval(cost(theta4_2), V))*flop+(eval(cost(theta5_2),
V))*flop+(eval(cost(theta6_2), V))*flop;
> total_cost_config_4 := (eval(cost(theta1_1),
V))*flop+(eval(cost(theta3_2), V))*flop+(eval(cost(theta2_3),
V))*flop+(eval(cost(theta4_6), V))*flop+(eval(cost(theta5_6),
V))*flop+(eval(cost(theta6_6), V))*flop;
> total_cost_config_5 := (eval(cost(theta1_2),
V))*flop+(eval(cost(theta3_1), V))*flop+(eval(cost(theta2_2),
V))*flop+(eval(cost(theta4_3), V))*flop+(eval(cost(theta5_3),
V))*flop+(eval(cost(theta6_3), V))*flop;
> total_cost_config_6 := (eval(cost(theta1_2),
V))*flop+(eval(cost(theta3_1), V))*flop+(eval(cost(theta2_2),
V))*flop+(eval(cost(theta4_7), V))*flop+(eval(cost(theta5_7),
V))*flop+(eval(cost(theta6_7), V))*flop;
> total_cost_config_7 := (eval(cost(theta1_2),
V))*flop+(eval(cost(theta3_2), V))*flop+(eval(cost(theta2_4),
V))*flop+(eval(cost(theta4_4), V))*flop+(eval(cost(theta5_4),
V))*flop+(eval(cost(theta6_4), V))*flop;
> total_cost_config_8 := (eval(cost(theta1_2),
V))*flop+(eval(cost(theta3_2), V))*flop+(eval(cost(theta2_4),
V))*flop+(eval(cost(theta4_8), V))*flop+(eval(cost(theta5_8),
V))*flop+(eval(cost(theta6_8), V))*flop;
> theta1_1_rad := evalf(theta1_1, 5);
> theta1_2_rad := evalf(theta1_2, 5);
> theta2_1_rad := evalf(theta2_1, 5);
> theta2_2_rad := evalf(theta2_2, 5);
> theta2_3_rad := evalf(theta2_3, 5);
> theta2_4_rad := evalf(theta2_4, 5);
> theta3_1_rad := evalf(theta3_1, 5);
> theta3_2_rad := evalf(theta3_2, 5);
> theta4_1_rad := evalf(theta4_1, 5);
> theta4_2_rad := evalf(theta4_2, 5);
> theta4_3_rad := evalf(theta4_3, 5);
> theta4_4_rad := evalf(theta4_4, 5);
```

```
> theta4_5_rad := evalf(theta4_5, 5);
> theta4_6_rad := evalf(theta4_6, 5);
> theta4_7_rad := evalf(theta4_7, 5);
> theta4_8_rad := evalf(theta4_8, 5);
> theta5_1_rad := evalf(theta5_1, 5);
> theta5_2_rad := evalf(theta5_2, 5);
> theta5_3_rad := evalf(theta5_3, 5);
> theta5_4_rad := evalf(theta5_4, 5);
> theta5_5_rad := evalf(theta5_5, 5);
> theta5_6_rad := evalf(theta5_6, 5);
> theta5_7_rad := evalf(theta5_7, 5);
> theta5_8_rad := evalf(theta5_8, 5);
> theta6_1_rad := evalf(theta6_1, 5);
> theta6_2_rad := evalf(theta6_2, 5);
> theta6_3_rad := evalf(theta6_3, 5);
> theta6_4_rad := evalf(theta6_4, 5);
> theta6_5_rad := evalf(theta6_5, 5);
> theta6_6_rad := evalf(theta6_6, 5);
> theta6_7_rad := evalf(theta6_7, 5);
> theta6_8_rad := evalf(theta6_8, 5);
> #####
> # Saídas
> #####
> cost_1_1 := 0
> cost_1_2 := multiplications + functions + additions
> cost_3_1 := 0
> cost_3_2 := multiplications + functions + additions
> cost_2_1 := 77 additions + 105 functions + 134 multiplications + 5
divisions
> cost_2_2 := 105 functions + 76 additions + 134 multiplications + 5
divisions
> cost_2_3 := 78 additions + 106 functions + 135 multiplications + 5
divisions
> cost_2_4 := 78 additions + 106 functions + 135 multiplications + 5
divisions
> cost_4_1 := 0
> cost_4_2 := 0
> cost_4_3 := 0
> cost_4_4 := 0
> cost_4_5 := multiplications
> cost_4_6 := multiplications
> cost_4_7 := 0
> cost_4_8 := multiplications
> cost_5_1 := multiplications
> cost_5_2 := 113 functions + 80 additions + 142 multiplications + 5
divisions
> cost_5_3 := 113 functions + 81 additions + 142 multiplications + 5
divisions
> cost_5_4 := 113 functions + 80 additions + 142 multiplications + 5
divisions
> cost_5_5 := multiplications
> cost_5_6 := 113 functions + 81 additions + 142 multiplications + 5
divisions
> cost_5_7 := 113 functions + 80 additions + 142 multiplications + 5
divisions
```

```
> cost_5_8 := 113 functions + 81 additions + 142 multiplications + 5
divisions
> cost_6_1 := 8 additions + 12 functions + 12 multiplications + divisions
> cost_6_2 := 376 functions + 264 additions + 469 multiplications + 21
divisions
> cost_6_3 := 376 functions + 265 additions + 469 multiplications + 21
divisions
> cost_6_4 := 376 functions + 264 additions + 469 multiplications + 21
divisions
> cost_6_5 := 8 additions + 12 functions + 13 multiplications + divisions
> cost_6_6 := 376 functions + 265 additions + 469 multiplications + 21
divisions
> cost_6_7 := 376 functions + 265 additions + 469 multiplications + 21
divisions
> cost_6_8 := 376 functions + 265 additions + 469 multiplications + 21
divisions
> flop_theta1_1 := 0
> flop_theta1_2 := 10 flop
> flop_theta3_1 := 0
> flop_theta3_2 := 10 flop
> flop_theta2_1 := 1071 flop
> flop_theta2_2 := 1070 flop
> flop_theta2_3 := 1081 flop
> flop_theta2_4 := 1081 flop
> flop_theta4_1 := 0
> flop_theta4_2 := 0
> flop_theta4_3 := 0
> flop_theta4_4 := 0
> flop_theta4_5 := flop
> flop_theta4_6 := flop
> flop_theta4_7 := 0
> flop_theta4_8 := flop
> flop_theta5_1 := flop
> flop_theta5_2 := 1146 flop
> flop_theta5_3 := 1147 flop
> flop_theta5_4 := 1146 flop
> flop_theta5_5 := flop
> flop_theta5_6 := 1147 flop
> flop_theta5_7 := 1146 flop
> flop_theta5_8 := 1147 flop
> flop_theta6_1 := 120 flop
> flop_theta6_2 := 3825 flop
> flop_theta6_3 := 3826 flop
> flop_theta6_4 := 3825 flop
> flop_theta6_5 := 121 flop
> flop_theta6_6 := 3826 flop
> flop_theta6_7 := 3826 flop
> flop_theta6_8 := 3826 flop
> total_cost_config_1 := 1192 flop
> total_cost_config_2 := 1194 flop
> total_cost_config_3 := 6062 flop
> total_cost_config_4 := 6065 flop
> total_cost_config_5 := 6053 flop
> total_cost_config_6 := 6052 flop
> total_cost_config_7 := 6072 flop
```

---

```
> total_cost_config_8 := 6075 flop
> total_cost := 34402 flop
> theta1_1_rad := 0.
> theta1_2_rad := -2.6020
> theta2_1_rad := 0.
> theta2_2_rad := 1.6207
> theta2_3_rad := 1.5209
> theta2_4_rad := 3.1416
> theta3_1_rad := 0.
> theta3_2_rad := -3.0486
> theta4_1_rad := 3.1416
> theta4_2_rad := 3.1416
> theta4_3_rad := 0.
> theta4_4_rad := 3.1416
> theta4_5_rad := 6.2832
> theta4_6_rad := 6.2832
> theta4_7_rad := 3.1416
> theta4_8_rad := 6.2832
> theta5_1_rad := 0.
> theta5_2_rad := -1.5277
> theta5_3_rad := -1.6207
> theta5_4_rad := 0.092956
> theta5_5_rad := 0.
> theta5_6_rad := 1.5277
> theta5_7_rad := 1.6207
> theta5_8_rad := -0.092956
> theta6_1_rad := 3.1416
> theta6_2_rad := 3.1416
> theta6_3_rad := -2.6020
> theta6_4_rad := 0.53961
> theta6_5_rad := 0.
> theta6_6_rad := 0.
> theta6_7_rad := 0.53961
> theta6_8_rad := -2.6020
> #####
```



# APÊNDICE E

```
> #####
> # Determinação de uma Base de Gröbner para o PUMA 560 - 1º Caso
> #####
> with(Groebner);
> with(codegen, cost);
> #####
> #Entradas
> #####
> a2 := 432;
> d3 := 125;
> a3 := 20;
> d4 := 430;
> nx := 1;
> ny := 0;
> nz := 0;
> ox := 0;
> oy := -1;
> oz := 0;
> ax := 0;
> ay := 0;
> az := -1;
> x := 452;
> y := 125;
> z := -430;
> r1 := (((c1*c2*c3-c1*s2*s3)*c4+s1*s4)*c5+(-c1*c2*s3-c1*c3*s2)*s5)*c6-
((c1*c2*c3-c1*s2*s3)*s4-s1*c4)*s6-nx;
> r2 := -(((c1*c2*c3-c1*s2*s3)*c4+s1*s4)*c5+(-c1*c2*s3-c1*c3*s2)*s5)*s6-
((c1*c2*c3-c1*s2*s3)*s4-s1*c4)*c6-ox;
> r3 := -((c1*c2*c3-c1*s2*s3)*c4+s1*s4)*s5+(-c1*c2*s3-c1*c3*s2)*c5-ax;
> r4 := (c1*c2*c3-c1*s2*s3)*a3+(-c1*c2*s3-c1*c3*s2)*d4+c1*c2*a2-s1*d3-x;
> r5 := (((c2*c3*s1-s1*s2*s3)*c4-c1*s4)*c5+(-c2*s1*s3-c3*s1*s2)*s5)*c6-
((c2*c3*s1-s1*s2*s3)*s4+c1*c4)*s6-ny;
> r6 := -(((c2*c3*s1-s1*s2*s3)*c4-c1*s4)*c5+(-c2*s1*s3-c3*s1*s2)*s5)*s6-
((c2*c3*s1-s1*s2*s3)*s4+c1*c4)*c6-oy;
> r7 := -((c2*c3*s1-s1*s2*s3)*c4-c1*s4)*s5+(-c2*s1*s3-c3*s1*s2)*c5-ay;
> r8 := (c2*c3*s1-s1*s2*s3)*a3+(-c2*s1*s3-c3*s1*s2)*d4+s1*c2*a2+c1*d3-y;
> r9 := ((-c2*s3-c3*s2)*c4*c5+(-c2*c3+s2*s3)*s5)*c6-(-c2*s3-c3*s2)*s4*s6-
nz;
> r10 := -((-c2*s3-c3*s2)*c4*c5+(-c2*c3+s2*s3)*s5)*s6-(-c2*s3-c3*s2)*s4*c6-
oz;
> r11 := -(-c2*s3-c3*s2)*c4*s5+(-c2*c3+s2*s3)*c5-az;
> r12 := (-c2*s3-c3*s2)*a3+(-c2*c3+s2*s3)*d4-s2*a2-z;
> r13 := c1^2+s1^2-1;
> r14 := c2^2+s2^2-1;
> r15 := c3^2+s3^2-1;
> r16 := c4^2+s4^2-1;
> r17 := c5^2+s5^2-1;
> r18 := c6^2+s6^2-1;
```

```
> GB := Groebner[Basis]([r1, r2, r3, r4, r5, r6, r7, r8, r9, r10, r11, r12,
r13, r14, r15, r16, r17, r18], plex(c6, s6, c5, s5, c4, s4, c2, s2, c3, s3,
c1, s1));
> math_operations := cost(GB);
> V := {additions = 1, divisions = 4, functions = 8, multiplications = 1}:
> total_cost := eval(cost(GB), V) flop;
> #####
> # Saídas
> #####
> GB := [219929*s1^2+113000*s1, 125*c1-452*s1-125, 1853*s3^2+172*s3, 2*c3-
43*s3-2, 407528437*s1*s3+18913894*s1+9730100*s2+104694500*s3, 24325250*c2-
248996875*s3-49703954*s1-24325250, s1*s4, s4*s3, c4^2+s4^2-1,
47504664*c4*s1*s3-47284735*c4*s1+261819000*c4*s3+24325250*s5, -
510675138*s1*s3+12162625*c5-24851977*s1-125388000*s3-12162625, -
c4*s1+s4+s6, -452*c4*s1-125*c4+125*c6]
> math_operations := 25*additions+40*multiplications
> total_cost := 65*flop
> #####
```

## APÊNDICE F

```
> #####
> # Solução da Cinemática Inversa do PUMA 560 (Gröbner -> Solve)- 1º Caso
> #####
> with(linalg);
> with(Units[Natural]);
> with(codegen, cost);
> #####
> #Entradas
> #####
> c1 := cos(theta1);
> c2 := cos(theta2);
> c3 := cos(theta3);
> c4 := cos(theta4);
> c5 := cos(theta5);
> c6 := cos(theta6);
> s1 := sin(theta1);
> s2 := sin(theta2);
> s3 := sin(theta3);
> s4 := sin(theta4);
> s5 := sin(theta5);
> s6 := sin(theta6);
> Eq := 219929*s1^2+113000*s1, 125*c1-452*s1-125, 1853*s3^2+172*s3, 2*c3-
43*s3-2, 407528437*s1*s3+18913894*s1+9730100*s2+104694500*s3, 24325250*c2-
248996875*s3-49703954*s1-24325250, s1*s4, s4*s3, c4^2+s4^2-1,
47504664*c4*s1*s3-47284735*c4*s1+261819000*c4*s3+24325250*s5, -
510675138*s1*s3+12162625*c5-24851977*s1-125388000*s3-12162625, -
c4*s1+s4+s6, -452*c4*s1-125*c4+125*c6;
> sys := {Eq};
> S := evalf(solve(sys, {theta1, theta2, theta3, theta4, theta5, theta6}),
5);
> math_operations := cost(Eq);
> V := {additions = 1, divisions = 4, functions = 8, multiplications = 1};
> total_cost := eval(cost(Eq), V);
> #####
> # Saídas
> #####
> S := {theta1 = -2.6020, theta2 = 3.1416, theta3 = -3.0486, theta4 =
3.1416, theta5 = 0.92956e-1, theta6 = .53961}, {theta1 = -2.6020, theta2 =
3.1416, theta3 = -3.0486, theta4 = 0., theta5 = -0.92956e-1, theta6 = -
2.6020}, {theta1 = 0., theta2 = 0., theta3 = 0., theta4 = 0., theta5 = 0.,
theta6 = 0.}, {theta1 = -2.6020, theta2 = 1.6207, theta3 = 0., theta4 = 0.,
theta5 = -1.6207, theta6 = -2.6020}, {theta1 = -2.6020, theta2 = 1.6207,
theta3 = 0., theta4 = 3.1416, theta5 = 1.6207, theta6 = .53961}, {theta1 =
0., theta2 = 1.5209, theta3 = -3.0486, theta4 = 0., theta5 = 1.5277, theta6
= 0.}, {theta1 = 0., theta2 = 1.5209, theta3 = -3.0486, theta4 = 3.1416,
theta5 = -1.5277, theta6 = 3.1416}, {theta1 = 0., theta2 = 0., theta3 = 0.,
theta4 = 3.1416, theta5 = 0., theta6 = 3.1416}
> math_operations := 25*additions+40*multiplications+43*functions
> total_cost := 409*flop
> #####
```

# APÊNDICE G

```
> #####
> # Solução da Cinemática Inversa do PUMA 560 (Método de Paul) - 2º Caso
> #####
> with(codegen, cost);
> with(linalg);
> with(Units[Natural]);
> #####
> #Entradas
> #####
> c1_1 := cos(theta1_1);
> c1_2 := cos(theta1_2);
> c3_1 := cos(theta3_1);
> c3_2 := cos(theta3_2);
> c2_1 := cos(theta2_1);
> c2_2 := cos(theta2_2);
> c2_3 := cos(theta2_3);
> c2_4 := cos(theta2_4);
> c4_1 := cos(theta4_1);
> c4_2 := cos(theta4_2);
> c4_3 := cos(theta4_3);
> c4_4 := cos(theta4_4);
> c4_5 := cos(theta4_5);
> c4_6 := cos(theta4_6);
> c4_7 := cos(theta4_7);
> c4_8 := cos(theta4_8);
> c5_1 := cos(theta5_1);
> c5_2 := cos(theta5_2);
> c5_3 := cos(theta5_3);
> c5_4 := cos(theta5_4);
> c5_5 := cos(theta5_5);
> c5_6 := cos(theta5_6);
> c5_7 := cos(theta5_7);
> c5_8 := cos(theta5_8);
> c6_1 := cos(theta6_1);
> c6_2 := cos(theta6_2);
> c6_3 := cos(theta6_3);
> c6_4 := cos(theta6_4);
> c6_5 := cos(theta6_5);
> c6_6 := cos(theta6_6);
> c6_7 := cos(theta6_7);
> c6_8 := cos(theta6_8);
> s1_1 := sin(theta1_1);
> s1_2 := sin(theta1_2);
> s3_1 := sin(theta3_1);
> s3_2 := sin(theta3_2);
> s2_1 := sin(theta2_1);
> s2_2 := sin(theta2_2);
> s2_3 := sin(theta2_3);
> s2_4 := sin(theta2_4);
> s4_1 := sin(theta4_1);
```

```
> s4_2 := sin(theta4_2);
> s4_3 := sin(theta4_3);
> s4_4 := sin(theta4_4);
> s4_5 := sin(theta4_5);
> s4_6 := sin(theta4_6);
> s4_7 := sin(theta4_7);
> s4_8 := sin(theta4_8);
> s5_1 := sin(theta5_1);
> s5_2 := sin(theta5_2);
> s5_3 := sin(theta5_3);
> s5_4 := sin(theta5_4);
> s5_5 := sin(theta5_5);
> s5_6 := sin(theta5_6);
> s5_7 := sin(theta5_7);
> s5_8 := sin(theta5_8);
> s6_1 := sin(theta6_1);
> s6_2 := sin(theta6_2);
> s6_3 := sin(theta6_3);
> s6_4 := sin(theta6_4);
> s6_5 := sin(theta6_5);
> s6_6 := sin(theta6_6);
> s6_7 := sin(theta6_7);
> s6_8 := sin(theta6_8);
> c2_13_1 := cos(theta2_1+theta3_1);
> s2_13_1 := sin(theta2_1+theta3_1);
> c2_23_1 := cos(theta2_2+theta3_1);
> s2_23_1 := sin(theta2_2+theta3_1);
> c2_33_2 := cos(theta2_3+theta3_2);
> s2_33_2 := sin(theta2_3+theta3_2);
> c2_43_2 := cos(theta2_4+theta3_2);
> s2_43_2 := sin(theta2_4+theta3_2);
> a2 := 432;
> d3 := 125;
> a3 := 20;
> d4 := 430;
> nx := 1;
> ny := 0;
> nz := 0;
> ox := 0;
> oy := -1;
> oz := 0;
> ax := 0;
> ay := 0;
> az := -1;
> x := 125/2+215*sqrt(3);
> y := 215-125*sqrt(3)*(1/2);
> z := -452;
> theta1_1 := arctan(y, x)-arctan(d3, sqrt(-d3^2+x^2+y^2));
> theta1_2 := arctan(y, x)-arctan(d3, -sqrt(-d3^2+x^2+y^2));
> theta3_1 := arctan(a3, d4)-arctan((-a2^2-a3^2-d3^2-
d4^2+x^2+y^2+z^2)/(2*a2), sqrt(a3^2+d4^2-((-a2^2-a3^2-d3^2-
d4^2+x^2+y^2+z^2)/(2*a2))^2));
> theta3_2 := arctan(a3, d4)-arctan((-a2^2-a3^2-d3^2-
d4^2+x^2+y^2+z^2)/(2*a2), -sqrt(a3^2+d4^2-((-a2^2-a3^2-d3^2-
d4^2+x^2+y^2+z^2)/(2*a2))^2));
```

```

> theta2_1 := arctan((-a2*c3_1-a3)*z-(c1_1*x+s1_1*y)*(-a2*s3_1+d4),
(a2*s3_1-d4)*z-(a2*c3_1+a3)*(c1_1*x+s1_1*y))-theta3_1;
> theta2_2 := arctan((-a2*c3_1-a3)*z-(c1_2*x+s1_2*y)*(-a2*s3_1+d4),
(a2*s3_1-d4)*z-(a2*c3_1+a3)*(c1_2*x+s1_2*y))-theta3_1;
> theta2_3 := arctan((-a2*c3_2-a3)*z-(c1_1*x+s1_1*y)*(-a2*s3_2+d4),
(a2*s3_2-d4)*z-(a2*c3_2+a3)*(c1_1*x+s1_1*y))-theta3_2;
> theta2_4 := arctan((-a2*c3_2-a3)*z-(c1_2*x+s1_2*y)*(-a2*s3_2+d4),
(a2*s3_2-d4)*z-(a2*c3_2+a3)*(c1_2*x+s1_2*y))-theta3_2;
> theta4_1 := arctan(-ax*s1_1+ay*c1_1, -ax*c1_1*c2_13_1-
ay*c2_13_1*s1_1+az*s2_13_1);
> theta4_2 := arctan(-ax*s1_2+ay*c1_2, -ax*c1_2*c2_23_1-
ay*c2_23_1*s1_2+az*s2_23_1);
> theta4_3 := arctan(-ax*s1_1+ay*c1_1, -ax*c1_1*c2_33_2-
ay*c2_33_2*s1_1+az*s2_33_2);
> theta4_4 := arctan(-ax*s1_2+ay*c1_2, -ax*c1_2*c2_43_2-
ay*c2_43_2*s1_2+az*s2_43_2);
> theta4_5 := theta4_1+Pi;
> theta4_6 := theta4_2+Pi;
> theta4_7 := theta4_3+Pi;
> theta4_8 := theta4_4+Pi;
> theta5_1 := arctan(-ax*(c1_1*c2_13_1*c4_1+s1_1*s4_1)-
ay*(c2_13_1*c4_1*s1_1-c1_1*s4_1)+az*s2_13_1*c4_1, -ax*c1_1*s2_13_1-
ay*s1_1*s2_13_1-az*c2_13_1);
> theta5_2 := arctan(-ax*(c1_2*c2_23_1*c4_2+s1_2*s4_2)-
ay*(c2_23_1*c4_2*s1_2-c1_2*s4_2)+az*s2_23_1*c4_2, -ax*c1_2*s2_23_1-
ay*s1_2*s2_23_1-az*c2_23_1);
> theta5_3 := arctan(-ax*(c1_1*c2_33_2*c4_3+s1_1*s4_3)-
ay*(c2_33_2*c4_3*s1_1-c1_1*s4_3)+az*s2_33_2*c4_3, -ax*c1_1*s2_33_2-
ay*s1_1*s2_33_2-az*c2_33_2);
> theta5_4 := arctan(-ax*(c1_2*c2_43_2*c4_4+s1_2*s4_4)-
ay*(c2_43_2*c4_4*s1_2-c1_2*s4_4)+az*s2_43_2*c4_4, -ax*c1_2*s2_43_2-
ay*s1_2*s2_43_2-az*c2_43_2);
> theta5_5 := -theta5_1;
> theta5_6 := -theta5_2;
> theta5_7 := -theta5_3;
> theta5_8 := -theta5_4;
> theta6_1 := arctan(-nx*(c1_1*c2_13_1*s4_1-c4_1*s1_1)-
ny*(c2_13_1*s1_1*s4_1+c1_1*c4_1)+nz*s2_13_1*s4_1,
nx*((c1_1*c2_13_1*c4_1+s1_1*s4_1)*c5_1-
c1_1*s2_13_1*s5_1)+ny*((c2_13_1*c4_1*s1_1-c1_1*s4_1)*c5_1-
s1_1*s2_13_1*s5_1)-nz*(c2_13_1*c4_1*s1_1*s5_1+c4_1*c5_1*s2_13_1));
> theta6_2 := arctan(-nx*(c1_2*c2_23_1*s4_2-c4_2*s1_2)-
ny*(c2_23_1*s1_2*s4_2+c1_2*c4_2)+nz*s2_23_1*s4_2,
nx*((c1_2*c2_23_1*c4_2+s1_2*s4_2)*c5_2-
c1_2*s2_23_1*s5_2)+ny*((c2_23_1*c4_2*s1_2-c1_2*s4_2)*c5_2-
s1_2*s2_23_1*s5_2)-nz*(c2_23_1*c4_2*s1_2*s5_2+c4_2*c5_2*s2_23_1));
> theta6_3 := arctan(-nx*(c1_1*c2_33_2*s4_3-c4_3*s1_1)-
ny*(c2_33_2*s1_1*s4_3+c1_1*c4_3)+nz*s2_33_2*s4_3,
nx*((c1_1*c2_33_2*c4_3+s1_1*s4_3)*c5_3-
c1_1*s2_33_2*s5_3)+ny*((c2_33_2*c4_3*s1_1-c1_1*s4_3)*c5_3-
s1_1*s2_33_2*s5_3)-nz*(c2_33_2*c4_3*s1_1*s5_3+c4_3*c5_3*s2_33_2));
> theta6_4 := arctan(-nx*(c1_2*c2_43_2*s4_4-c4_4*s1_2)-
ny*(c2_43_2*s1_2*s4_4+c1_2*c4_4)+nz*s2_43_2*s4_4,
nx*((c1_2*c2_43_2*c4_4+s1_2*s4_4)*c5_4-
c1_2*s2_43_2*s5_4)+ny*((c2_43_2*c4_4*s1_2-c1_2*s4_4)*c5_4-
s1_2*s2_43_2*s5_4)-nz*(c2_43_2*c4_4*s1_2*s5_4+c4_4*c5_4*s2_43_2));
> theta6_5 := theta6_1+Pi;

```

```
> theta6_6 := theta6_2+Pi;
> theta6_7 := theta6_3+Pi;
> theta6_8 := theta6_4+Pi;
> cost_1_1 := cost(theta1_1);
> cost_1_2 := cost(theta1_2);
> cost_3_1 := cost(theta3_1);
> cost_3_2 := cost(theta3_2);
> cost_2_1 := cost(theta2_1);
> cost_2_2 := cost(theta2_2);
> cost_2_3 := cost(theta2_3);
> cost_2_4 := cost(theta2_4);
> cost_4_1 := cost(theta4_1);
> cost_4_2 := cost(theta4_2);
> cost_4_3 := cost(theta4_3);
> cost_4_4 := cost(theta4_4);
> cost_4_5 := cost(theta4_5);
> cost_4_6 := cost(theta4_6);
> cost_4_7 := cost(theta4_7);
> cost_4_8 := cost(theta4_8);
> cost_5_1 := cost(theta5_1);
> cost_5_2 := cost(theta5_2);
> cost_5_3 := cost(theta5_3);
> cost_5_4 := cost(theta5_4);
> cost_5_5 := cost(theta5_5);
> cost_5_6 := cost(theta5_6);
> cost_5_7 := cost(theta5_7);
> cost_5_8 := cost(theta5_8);
> cost_6_1 := cost(theta6_1);
> cost_6_2 := cost(theta6_2);
> cost_6_3 := cost(theta6_3);
> cost_6_4 := cost(theta6_4);
> cost_6_5 := cost(theta6_5);
> cost_6_6 := cost(theta6_6);
> cost_6_7 := cost(theta6_7);
> cost_6_8 := cost(theta6_8);
> V := {additions = 1, divisions = 4, functions = 8, multiplications = 1};
> flop_theta1_1 := (eval(cost(theta1_1), V))*flop;
> flop_theta1_2 := (eval(cost(theta1_2), V))*flop;
> flop_theta3_1 := (eval(cost(theta3_1), V))*flop;
> flop_theta3_2 := (eval(cost(theta3_2), V))*flop;
> flop_theta2_1 := (eval(cost(theta2_1), V))*flop;
> flop_theta2_2 := (eval(cost(theta2_2), V))*flop;
> flop_theta2_3 := (eval(cost(theta2_3), V))*flop;
> flop_theta2_4 := (eval(cost(theta2_4), V))*flop;
> flop_theta4_1 := (eval(cost(theta4_1), V))*flop;
> flop_theta4_2 := (eval(cost(theta4_2), V))*flop;
> flop_theta4_3 := (eval(cost(theta4_3), V))*flop;
> flop_theta4_4 := (eval(cost(theta4_4), V))*flop;
> flop_theta4_5 := (eval(cost(theta4_5), V))*flop;
> flop_theta4_6 := (eval(cost(theta4_6), V))*flop;
> flop_theta4_7 := (eval(cost(theta4_7), V))*flop;
> flop_theta4_8 := (eval(cost(theta4_8), V))*flop;
> flop_theta5_1 := (eval(cost(theta5_1), V))*flop;
> flop_theta5_2 := (eval(cost(theta5_2), V))*flop;
> flop_theta5_3 := (eval(cost(theta5_3), V))*flop;
```

```
> flop_theta5_4 := (eval(cost(theta5_4), V))*flop;
> flop_theta5_5 := (eval(cost(theta5_5), V))*flop;
> flop_theta5_6 := (eval(cost(theta5_6), V))*flop;
> flop_theta5_7 := (eval(cost(theta5_7), V))*flop;
> flop_theta5_8 := (eval(cost(theta5_8), V))*flop;
> flop_theta6_1 := (eval(cost(theta6_1), V))*flop;
> flop_theta6_2 := (eval(cost(theta6_2), V))*flop;
> flop_theta6_3 := (eval(cost(theta6_3), V))*flop;
> flop_theta6_4 := (eval(cost(theta6_4), V))*flop;
> flop_theta6_5 := (eval(cost(theta6_5), V))*flop;
> flop_theta6_6 := (eval(cost(theta6_6), V))*flop;
> flop_theta6_7 := (eval(cost(theta6_7), V))*flop;
> flop_theta6_8 := (eval(cost(theta6_8), V))*flop;
> total_cost_config_1 := (eval(cost(theta1_1),
V))*flop+(eval(cost(theta3_1), V))*flop+(eval(cost(theta2_1),
V))*flop+(eval(cost(theta4_1), V))*flop+(eval(cost(theta5_1),
V))*flop+(eval(cost(theta6_1), V))*flop;
> total_cost_config_2 := (eval(cost(theta1_1),
V))*flop+(eval(cost(theta3_1), V))*flop+(eval(cost(theta2_1),
V))*flop+(eval(cost(theta4_5), V))*flop+(eval(cost(theta5_5),
V))*flop+(eval(cost(theta6_5), V))*flop;
> total_cost_config_3 := (eval(cost(theta1_1),
V))*flop+(eval(cost(theta3_2), V))*flop+(eval(cost(theta2_3),
V))*flop+(eval(cost(theta4_2), V))*flop+(eval(cost(theta5_2),
V))*flop+(eval(cost(theta6_2), V))*flop;
> total_cost_config_4 := (eval(cost(theta1_1),
V))*flop+(eval(cost(theta3_2), V))*flop+(eval(cost(theta2_3),
V))*flop+(eval(cost(theta4_6), V))*flop+(eval(cost(theta5_6),
V))*flop+(eval(cost(theta6_6), V))*flop;
> total_cost_config_5 := (eval(cost(theta1_2),
V))*flop+(eval(cost(theta3_1), V))*flop+(eval(cost(theta2_2),
V))*flop+(eval(cost(theta4_3), V))*flop+(eval(cost(theta5_3),
V))*flop+(eval(cost(theta6_3), V))*flop;
> total_cost_config_6 := (eval(cost(theta1_2),
V))*flop+(eval(cost(theta3_1), V))*flop+(eval(cost(theta2_2),
V))*flop+(eval(cost(theta4_7), V))*flop+(eval(cost(theta5_7),
V))*flop+(eval(cost(theta6_7), V))*flop;
> total_cost_config_7 := (eval(cost(theta1_2),
V))*flop+(eval(cost(theta3_2), V))*flop+(eval(cost(theta2_4),
V))*flop+(eval(cost(theta4_4), V))*flop+(eval(cost(theta5_4),
V))*flop+(eval(cost(theta6_4), V))*flop;
> total_cost_config_8 := (eval(cost(theta1_2),
V))*flop+(eval(cost(theta3_2), V))*flop+(eval(cost(theta2_4),
V))*flop+(eval(cost(theta4_8), V))*flop+(eval(cost(theta5_8),
V))*flop+(eval(cost(theta6_8), V))*flop;
> theta1_1_rad := evalf(theta1_1, 5);
> theta1_2_rad := evalf(theta1_2, 5);
> theta2_1_rad := evalf(theta2_1, 5);
> theta2_2_rad := evalf(theta2_2, 5);
> theta2_3_rad := evalf(theta2_3, 5);
> theta2_4_rad := evalf(theta2_4, 5);
> theta3_1_rad := evalf(theta3_1, 5);
> theta3_2_rad := evalf(theta3_2, 5);
> theta4_1_rad := evalf(theta4_1, 5);
> theta4_2_rad := evalf(theta4_2, 5);
> theta4_3_rad := evalf(theta4_3, 5);
> theta4_4_rad := evalf(theta4_4, 5);
```



```
> theta4_5_rad := evalf(theta4_5, 5);
> theta4_6_rad := evalf(theta4_6, 5);
> theta4_7_rad := evalf(theta4_7, 5);
> theta4_8_rad := evalf(theta4_8, 5);
> theta5_1_rad := evalf(theta5_1, 5);
> theta5_2_rad := evalf(theta5_2, 5);
> theta5_3_rad := evalf(theta5_3, 5);
> theta5_4_rad := evalf(theta5_4, 5);
> theta5_5_rad := evalf(theta5_5, 5);
> theta5_6_rad := evalf(theta5_6, 5);
> theta5_7_rad := evalf(theta5_7, 5);
> theta5_8_rad := evalf(theta5_8, 5);
> theta6_1_rad := evalf(theta6_1, 5);
> theta6_2_rad := evalf(theta6_2, 5);
> theta6_3_rad := evalf(theta6_3, 5);
> theta6_4_rad := evalf(theta6_4, 5);
> theta6_5_rad := evalf(theta6_5, 5);
> theta6_6_rad := evalf(theta6_6, 5);
> theta6_7_rad := evalf(theta6_7, 5);
> theta6_8_rad := evalf(theta6_8, 5);
> #####
> # Saídas
> #####
> cost_1_1 := 12 functions + 7 additions + 12 multiplications + divisions
> cost_1_2 := 8 additions + 12 functions + 12 multiplications + divisions
> cost_3_1 := 12 functions + 19 multiplications + 10 additions
> cost_3_2 := 11 additions + 12 functions + 19 multiplications
> cost_2_1 := 11 additions + 20 multiplications + 12 functions
> cost_2_2 := 91 additions + 125 functions + 161 multiplications + 5
divisions
> cost_2_3 := 92 additions + 125 functions + 161 multiplications + 5
divisions
> cost_2_4 := 92 additions + 125 functions + 161 multiplications + 5
divisions
> cost_4_1 := 0
> cost_4_2 := 0
> cost_4_3 := 0
> cost_4_4 := 0
> cost_4_5 := multiplications
> cost_4_6 := multiplications
> cost_4_7 := 0
> cost_4_8 := multiplications
> cost_5_1 := multiplications
> cost_5_2 := 113 functions + 80 additions + 142 multiplications + 5
divisions
> cost_5_3 := 113 functions + 81 additions + 142 multiplications + 5
divisions
> cost_5_4 := 113 functions + 80 additions + 142 multiplications + 5
divisions
> cost_5_5 := multiplications
> cost_5_6 := 113 functions + 81 additions + 142 multiplications + 5
divisions
> cost_5_7 := 113 functions + 80 additions + 142 multiplications + 5
divisions
> cost_5_8 := 113 functions + 81 additions + 142 multiplications + 5
divisions
```

---

```
> cost_6_1 := 8 additions + 12 functions + 12 multiplications + divisions
> cost_6_2 := 376 functions + 264 additions + 469 multiplications + 21
divisions
> cost_6_3 := 376 functions + 265 additions + 469 multiplications + 21
divisions
> cost_6_4 := 376 functions + 264 additions + 469 multiplications + 21
divisions
> cost_6_5 := 8 additions + 12 functions + 13 multiplications + divisions
> cost_6_6 := 376 functions + 265 additions + 469 multiplications + 21
divisions
> cost_6_7 := 376 functions + 265 additions + 469 multiplications + 21
divisions
> cost_6_8 := 376 functions + 265 additions + 469 multiplications + 21
divisions
> flop_theta1_1 := 119 flop
> flop_theta1_2 := 120 flop
> flop_theta3_1 := 125 flop
> flop_theta3_2 := 126 flop
> flop_theta2_1 := 127 flop
> flop_theta2_2 := 1272 flop
> flop_theta2_3 := 1273 flop
> flop_theta2_4 := 1273 flop
> flop_theta4_1 := 0
> flop_theta4_2 := 0
> flop_theta4_3 := 0
> flop_theta4_4 := 0
> flop_theta4_5 := flop
> flop_theta4_6 := flop
> flop_theta4_7 := 0
> flop_theta4_8 := flop
> flop_theta5_1 := flop
> flop_theta5_2 := 1146 flop
> flop_theta5_3 := 1147 flop
> flop_theta5_4 := 1146 flop
> flop_theta5_5 := flop
> flop_theta5_6 := 1147 flop
> flop_theta5_7 := 1146 flop
> flop_theta5_8 := 1147 flop
> flop_theta6_1 := 120 flop
> flop_theta6_2 := 3825 flop
> flop_theta6_3 := 3826 flop
> flop_theta6_4 := 3825 flop
> flop_theta6_5 := 121 flop
> flop_theta6_6 := 3826 flop
> flop_theta6_7 := 3826 flop
> flop_theta6_8 := 3826 flop
> total_cost_config_1 := 492 flop
> total_cost_config_2 := 494 flop
> total_cost_config_3 := 6489 flop
> total_cost_config_4 := 6492 flop
> total_cost_config_5 := 6490 flop
> total_cost_config_6 := 6489 flop
> total_cost_config_7 := 6490 flop
> total_cost_config_8 := 6493 flop
> total_cost := 34514 flop
```

---

```
> theta1_1_rad := -0.04221
> theta1_2_rad := -2.6180
> theta2_1_rad := 0.049876
> theta2_2_rad := 1.5708
> theta2_3_rad := 1.5708
> theta2_4_rad := 3.0917
> theta3_1_rad := 0.
> theta3_2_rad := -3.0486
> theta4_1_rad := 3.1416
> theta4_2_rad := 3.1416
> theta4_3_rad := 0.
> theta4_4_rad := 3.1416
> theta4_5_rad := 6.2832
> theta4_6_rad := 6.2832
> theta4_7_rad := 3.1416
> theta4_8_rad := 6.2832
> theta5_1_rad := 0.49871e-1
> theta5_2_rad := -1.4778
> theta5_3_rad := -1.5708
> theta5_4_rad := 0.43080e-1
> theta5_5_rad := -0.49871e-1
> theta5_6_rad := 1.4778
> theta5_7_rad := 1.5708
> theta5_8_rad := -0.43080e-1
> theta6_1_rad := 3.0994
> theta6_2_rad := 3.0994
> theta6_3_rad := -2.6180
> theta6_4_rad := 0.52361
> theta6_5_rad := -0.42211e-1
> theta6_6_rad := -0.42211e-1
> theta6_7_rad := 0.52361
> theta6_8_rad := -2.6180
> #####
```

# APÊNDICE H

```
> #####
> # Determinação de uma Base de Gröbner para o PUMA 560 - 2º Caso
> #####
> with(Groebner);
> with(codegen, cost);
> #####
> #Entradas
> #####
> a2 := 432;
> d3 := 125;
> a3 := 20;
> d4 := 430;
> nx := 1;
> ny := 0;
> nz := 0;
> ox := 0;
> oy := -1;
> oz := 0;
> ax := 0;
> ay := 0;
> az := -1;
> x := 125/2+215*sqrt(3);
> y := 215-125*sqrt(3)*(1/2);
> z := -452;
> r1 := (((c1*c2*c3-c1*s2*s3)*c4+s1*s4)*c5+(-c1*c2*s3-c1*c3*s2)*s5)*c6-
((c1*c2*c3-c1*s2*s3)*s4-s1*c4)*s6-nx;
> r2 := -(((c1*c2*c3-c1*s2*s3)*c4+s1*s4)*c5+(-c1*c2*s3-c1*c3*s2)*s5)*s6-
((c1*c2*c3-c1*s2*s3)*s4-s1*c4)*c6-ox;
> r3 := -((c1*c2*c3-c1*s2*s3)*c4+s1*s4)*s5+(-c1*c2*s3-c1*c3*s2)*c5-ax;
> r4 := (c1*c2*c3-c1*s2*s3)*a3+(-c1*c2*s3-c1*c3*s2)*d4+c1*c2*a2-s1*d3-x;
> r5 := (((c2*c3*s1-s1*s2*s3)*c4-c1*s4)*c5+(-c2*s1*s3-c3*s1*s2)*s5)*c6-
((c2*c3*s1-s1*s2*s3)*s4+c1*c4)*s6-ny;
> r6 := -(((c2*c3*s1-s1*s2*s3)*c4-c1*s4)*c5+(-c2*s1*s3-c3*s1*s2)*s5)*s6-
((c2*c3*s1-s1*s2*s3)*s4+c1*c4)*c6-oy;
> r7 := -((c2*c3*s1-s1*s2*s3)*c4-c1*s4)*s5+(-c2*s1*s3-c3*s1*s2)*c5-ay;
> r8 := (c2*c3*s1-s1*s2*s3)*a3+(-c2*s1*s3-c3*s1*s2)*d4+s1*c2*a2+c1*d3-y;
> r9 := ((-c2*s3-c3*s2)*c4*c5+(-c2*c3+s2*s3)*s5)*c6-(-c2*s3-c3*s2)*s4*s6-
nz;
> r10 := -((-c2*s3-c3*s2)*c4*c5+(-c2*c3+s2*s3)*s5)*s6-(-c2*s3-c3*s2)*s4*c6-
oz;
> r11 := -(-c2*s3-c3*s2)*c4*s5+(-c2*c3+s2*s3)*c5-az;
> r12 := (-c2*s3-c3*s2)*a3+(-c2*c3+s2*s3)*d4-s2*a2-z;
> r13 := c1^2+s1^2-1;
> r14 := c2^2+s2^2-1;
> r15 := c3^2+s3^2-1;
> r16 := c4^2+s4^2-1;
> r17 := c5^2+s5^2-1;
> r18 := c6^2+s6^2-1;
> GB := Groebner[Basis]([r1, r2, r3, r4, r5, r6, r7, r8, r9, r10, r11, r12,
r13, r14, r15, r16, r17, r18], plex(c6, s6, c5, s5, c4, s4, c2, s2, c3, s3,
c1, s1));
```

```
> math_operations := cost(GB);
> V := {additions = 1, divisions = 4, functions = 8, multiplications = 1}:
> total_cost := eval(cost(GB), V) flop;
> #####
> # Saídas
> #####
> GB := [8600*sqrt(3)*s1+32084*s1^2+4300*sqrt(3)+2500*s1-6771, -
8021*sqrt(3)*s1-1250*sqrt(3)+5521*c1-8600*s1-4300, 1853*s3^2+172*s3, 2*c3-
43*s3-2,
18578641250*s1*s3*sqrt(3)+862257500*sqrt(3)*s1+9289320625*sqrt(3)*s3+639105
25900*s1*s3+431128750*sqrt(3)+2966165800*s1+2148795284*s2+9959875000*s3-
665712384, -226593250*sqrt(3)*s1-113296625*sqrt(3)+537198821*c2-
779480780*s1-5780183345*s3-389740390, s4, c4^2-1,
433134000*c4*s3*s1*sqrt(3)-
431128750*s1*c4*sqrt(3)+216567000*c4*s3*sqrt(3)+1489980960*c4*s3*s1-
215564375*sqrt(3)*c4-
1483082900*s1*c4+11821264848*c4*s3+332856192*c4+1074397642*s5, -
4656190500*s1*s3*sqrt(3)-226593250*sqrt(3)*s1-2328095250*sqrt(3)*s3-
16017295320*s1*s3-113296625*sqrt(3)+537198821*c5-779480780*s1-
2226636864*s3-389740390, -c4*s1+s6, -8021*s1*c4*sqrt(3)-1250*sqrt(3)*c4-
8600*s1*c4-4300*c4+5521*c6]
> math_operations := 46*additions+100*multiplications+40*functions
> total_cost := 466*flop
> #####
```

# APÊNDICE I

```
> #####
> # Solução da Cinemática Inversa do PUMA 560 (Gröbner -> Solve)- 2º Caso
> #####
> with(linalg);
> with(Units[Natural]);
> with(codegen, cost);
> #####
> #Entradas
> #####
> c1 := cos(theta1);
> c2 := cos(theta2);
> c3 := cos(theta3);
> c4 := cos(theta4);
> c5 := cos(theta5);
> c6 := cos(theta6);
> s1 := sin(theta1);
> s2 := sin(theta2);
> s3 := sin(theta3);
> s4 := sin(theta4);
> s5 := sin(theta5);
> s6 := sin(theta6);
> Eq := 8600*sqrt(3)*s1+32084*s1^2+4300*sqrt(3)+2500*s1-6771, -
8021*sqrt(3)*s1-1250*sqrt(3)+5521*c1-8600*s1-4300, 1853*s3^2+172*s3, 2*c3-
43*s3-2,
18578641250*s1*s3*sqrt(3)+862257500*sqrt(3)*s1+9289320625*sqrt(3)*s3+639105
25900*s1*s3+431128750*sqrt(3)+2966165800*s1+2148795284*s2+9959875000*s3-
665712384, -226593250*sqrt(3)*s1-113296625*sqrt(3)+537198821*c2-
779480780*s1-5780183345*s3-389740390, s4, c4^2-1,
433134000*c4*s3*s1*sqrt(3)-
431128750*s1*c4*sqrt(3)+216567000*c4*s3*sqrt(3)+1489980960*c4*s3*s1-
215564375*sqrt(3)*c4-
1483082900*s1*c4+11821264848*c4*s3+332856192*c4+1074397642*s5, -
4656190500*s1*s3*sqrt(3)-226593250*sqrt(3)*s1-2328095250*sqrt(3)*s3-
16017295320*s1*s3-113296625*sqrt(3)+537198821*c5-779480780*s1-
2226636864*s3-389740390, -c4*s1+s6, -8021*s1*c4*sqrt(3)-1250*sqrt(3)*c4-
8600*s1*c4-4300*c4+5521*c6;
> sys := {Eq};
> S := evalf(solve(sys, {theta1, theta2, theta3, theta4, theta5, theta6}),
5);
> math_operations := cost(Eq);
> V := {additions = 1, divisions = 4, functions = 8, multiplications = 1};
> total_cost := eval(cost(Eq), V);
> #####
> # Saídas
> #####
> S := {theta1 = -2.6180, theta2 = 1.5708, theta3 = 0., theta4 = 0., theta5
= -1.5708, theta6 = -2.6180}, {theta1 = -2.6180, theta2 = 1.5708, theta3 =
0., theta4 = 3.1416, theta5 = 1.5708, theta6 = .52361}, {theta1 = -2.6180,
theta2 = 3.0917, theta3 = -3.0486, theta4 = 3.1416, theta5 = 0.43080e-1,
theta6 = .52361}, {theta1 = -2.6180, theta2 = 3.0917, theta3 = -3.0486,
theta4 = 0., theta5 = -0.43080e-1, theta6 = -2.6180}, {theta1 = -0.42214e-
```

---

```
1, theta2 = 1.5708, theta3 = -3.0486, theta4 = 0., theta5 = 1.4778, theta6
= -0.42211e-1}, {theta1 = -0.42214e-1, theta2 = 1.5708, theta3 = -3.0486,
theta4 = 3.1416, theta5 = -1.4778, theta6 = 3.0994}, {theta1 = -0.42214e-1,
theta2 = 0.49876e-1, theta3 = -4.7424*10^(-6), theta4 = 0., theta5 = -
0.49871e-1, theta6 = -0.42211e-1}, {theta1 = -0.42214e-1, theta2 =
0.49876e-1, theta3 = -4.7424*10^(-6), theta4 = 3.1416, theta5 = 0.49871e-1,
theta6 = 3.0994}
> math_operations := 46*additions+100*multiplications+101*functions
> total_cost := 954*flop
> #####
```

# APÊNDICE J

```
> #####
> #####
> # Alteração da Ordem Lexicográfica para Análise da Eficiência
> # Computacional - Análise Realizada no 1° e 2° Casos
> # Determinação da Base de Gröbner e Soluções (Gröbner -> Solve)
> #####
> #####

> #####
> # Determinação de uma Base de Gröbner para o PUMA 560 - 1° Caso - 2
> # Ordem lexicográfica (c1, s1, c3, s3, c2, s2, c4, s4, c5, s5, c6, s6)
> #####
> with(Groebner);
> with(codegen, cost);
> #####
> #Entradas
> #####
> a2 := 432;
> d3 := 125;
> a3 := 20;
> d4 := 430;
> nx := 1;
> ny := 0;
> nz := 0;
> ox := 0;
> oy := -1;
> oz := 0;
> ax := 0;
> ay := 0;
> az := -1;
> x := 452;
> y := 125;
> z := -430;
> r1 := (((c1*c2*c3-c1*s2*s3)*c4+s1*s4)*c5+(-c1*c2*s3-c1*c3*s2)*s5)*c6-
((c1*c2*c3-c1*s2*s3)*s4-s1*c4)*s6-nx;
> r2 := -(((c1*c2*c3-c1*s2*s3)*c4+s1*s4)*c5+(-c1*c2*s3-c1*c3*s2)*s5)*s6-
((c1*c2*c3-c1*s2*s3)*s4-s1*c4)*c6-ox;
> r3 := -((c1*c2*c3-c1*s2*s3)*c4+s1*s4)*s5+(-c1*c2*s3-c1*c3*s2)*c5-ax;
> r4 := (c1*c2*c3-c1*s2*s3)*a3+(-c1*c2*s3-c1*c3*s2)*d4+c1*c2*a2-s1*d3-x;
> r5 := (((c2*c3*s1-s1*s2*s3)*c4-c1*s4)*c5+(-c2*s1*s3-c3*s1*s2)*s5)*c6-
((c2*c3*s1-s1*s2*s3)*s4+c1*c4)*s6-ny;
> r6 := -(((c2*c3*s1-s1*s2*s3)*c4-c1*s4)*c5+(-c2*s1*s3-c3*s1*s2)*s5)*s6-
((c2*c3*s1-s1*s2*s3)*s4+c1*c4)*c6-oy;
> r7 := -((c2*c3*s1-s1*s2*s3)*c4-c1*s4)*s5+(-c2*s1*s3-c3*s1*s2)*c5-ay;
> r8 := (c2*c3*s1-s1*s2*s3)*a3+(-c2*s1*s3-c3*s1*s2)*d4+s1*c2*a2+c1*d3-y;
> r9 := ((-c2*s3-c3*s2)*c4*c5+(-c2*c3+s2*s3)*s5)*c6-(-c2*s3-c3*s2)*s4*s6-
nz;
> r10 := -((-c2*s3-c3*s2)*c4*c5+(-c2*c3+s2*s3)*s5)*s6-(-c2*s3-c3*s2)*s4*c6-
oz;
> r11 := -(-c2*s3-c3*s2)*c4*s5+(-c2*c3+s2*s3)*c5-az;
```



```

> r12 := (-c2*s3-c3*s2)*a3+(-c2*c3+s2*s3)*d4-s2*a2-z;
> r13 := c1^2+s1^2-1;
> r14 := c2^2+s2^2-1;
> r15 := c3^2+s3^2-1;
> r16 := c4^2+s4^2-1;
> r17 := c5^2+s5^2-1;
> r18 := c6^2+s6^2-1;
> GB := Groebner[Basis]([r1, r2, r3, r4, r5, r6, r7, r8, r9, r10, r11, r12,
r13, r14, r15, r16, r17, r18], plex(c1, s1, c3, s3, c2, s2, c4, s4, c5, s5,
c6, s6));
> math_operations := cost(GB);
> V := {additions = 1, divisions = 4, functions = 8, multiplications = 1}:
> total_cost := eval(cost(GB), V) flop;
> #####
> # Saídas
> #####
> GB := [c6^2+s6^2-1, 48368765041*s5*s6^3-12769000000*s5*s6,
113000*c6*s5*s6-188679*s5*s6^2,
8720828021042293873*s5^2*s6^2+2300098765968000000*c6*s5-
2302234777057000000*s5^2-3840533938620144000*s5*s6,
287779347132125000*c6*s5^2-
480512561394178875*s5^2*s6+1089092105707183794*s5*s6^2-
287512345746000000*s5, 25943128713538131870062500*s5^3-
55116487554029776576904500*s5^2*s6+107200400293469877668597389*s5*s6^2-
25895011028699959056000000*s5, -
1183339869120*c6*s5+8741785039205*s5^2+1110167046212*s5*s6+7882815906450*c5
-7882815906450, 2546719886125*s5^2*s6-
5410536888481*s5*s6^2+236098810000*s4+236098810000*s6, -
2546719886125*s5^2*s6+5410536888481*s5*s6^2+65292812500*c4-65292812500*c6,
4083486344100*c6*s5-
43708925196025*s5^2+1078062982364*s5*s6+39597400832400*s2,
401042384057562300*c6*s5-
385206757752568325*s5^2+113989410760005823*s5*s6+17026882357932000*c2-
17026882357932000, 9469462635900*c6*s5-
8741785039205*s5^2+1181349205663*s5*s6+7919480166480*s3,
9469462635900*c6*s5-8741785039205*s5^2+1181349205663*s5*s6+368347914720*c3-
368347914720, 4503286800*c6*s5-
4507468825*s5^2+2056917943*s5*s6+813297442*s1, 1017742816800*c6*s5-
1018687954450*s5^2+464863455118*s5*s6+50831090125*c1-50831090125]
> math_operations := 44*additions+109*multiplications
> total_cost := 153*flop
> #####
> #####
> #####
> # Solução da Cin. Inversa do PUMA 560 (Gröbner -> Solve)- 1º Caso - 2
> # Ordem lexicográfica (c1, s1, c3, s3, c2, s2, c4, s4, c5, s5, c6, s6)
> #####
> with(linalg);
> with(Units[Natural]);
> with(codegen, cost);
> #####
> #Entradas
> #####
> c1 := cos(theta1);

```

```

> c2 := cos(theta2);
> c3 := cos(theta3);
> c4 := cos(theta4);
> c5 := cos(theta5);
> c6 := cos(theta6);
> s1 := sin(theta1);
> s2 := sin(theta2);
> s3 := sin(theta3);
> s4 := sin(theta4);
> s5 := sin(theta5);
> s6 := sin(theta6);
> Eq := c6^2+s6^2-1, 48368765041*s5*s6^3-12769000000*s5*s6,
113000*c6*s5*s6-188679*s5*s6^2,
8720828021042293873*s5^2*s6^2+2300098765968000000*c6*s5-
2302234777057000000*s5^2-3840533938620144000*s5*s6,
287779347132125000*c6*s5^2-
480512561394178875*s5^2*s6+1089092105707183794*s5*s6^2-
287512345746000000*s5, 25943128713538131870062500*s5^3-
55116487554029776576904500*s5^2*s6+107200400293469877668597389*s5*s6^2-
25895011028699959056000000*s5, -
1183339869120*c6*s5+8741785039205*s5^2+1110167046212*s5*s6+7882815906450*c5
-7882815906450, 2546719886125*s5^2*s6-
5410536888481*s5*s6^2+236098810000*s4+236098810000*s6, -
2546719886125*s5^2*s6+5410536888481*s5*s6^2+65292812500*c4-65292812500*c6,
4083486344100*c6*s5-
43708925196025*s5^2+1078062982364*s5*s6+39597400832400*s2,
401042384057562300*c6*s5-
385206757752568325*s5^2+113989410760005823*s5*s6+17026882357932000*c2-
17026882357932000, 9469462635900*c6*s5-
8741785039205*s5^2+1181349205663*s5*s6+7919480166480*s3,
9469462635900*c6*s5-8741785039205*s5^2+1181349205663*s5*s6+368347914720*c3-
368347914720, 4503286800*c6*s5-
4507468825*s5^2+2056917943*s5*s6+813297442*s1, 1017742816800*c6*s5-
1018687954450*s5^2+464863455118*s5*s6+50831090125*c1-50831090125;
> sys := {Eq};
> S := evalf(solve(sys, {theta1, theta2, theta3, theta4, theta5, theta6}),
5);
> math_operations := cost(Eq);
> V := {additions = 1, divisions = 4, functions = 8, multiplications = 1};
> total_cost := eval(cost(Eq), V);
> #####
> # Saídas
> #####
> S := {theta1 = -2.6020, theta2 = 3.1416, theta3 = -3.0486, theta4 =
3.1416, theta5 = 0.92956e-1, theta6 = .53961}, {theta1 = -2.6020, theta2 =
3.1416, theta3 = -3.0486, theta4 = 0., theta5 = -0.92956e-1, theta6 = -
2.6020}, {theta1 = 0., theta2 = 0., theta3 = 0., theta4 = 0., theta5 = 0.,
theta6 = 0.}, {theta1 = -2.6020, theta2 = 1.6207, theta3 = 0., theta4 = 0.,
theta5 = -1.6207, theta6 = -2.6020}, {theta1 = -2.6020, theta2 = 1.6207,
theta3 = 0., theta4 = 3.1416, theta5 = 1.6207, theta6 = .53961}, {theta1 =
0., theta2 = 1.5209, theta3 = -3.0486, theta4 = 0., theta5 = 1.5277, theta6
= 0.}, {theta1 = 0., theta2 = 1.5209, theta3 = -3.0486, theta4 = 3.1416,
theta5 = -1.5277, theta6 = 3.1416}, {theta1 = 0., theta2 = 0., theta3 = 0.,
theta4 = 3.1416, theta5 = 0., theta6 = 3.1416}
> math_operations := 44*additions+109*multiplications+85*functions
> total_cost := 833*flop
> #####

```

```

> #####
> # Determinação de uma Base de Gröbner para o PUMA 560 - 2º Caso - 2
> # Ordem lexicográfica (c1, s1, c3, s3, c2, s2, c4, s4, c5, s5, c6, s6)
> #####
> with(Groebner);
> with(codegen, cost);
> #####
> #Entradas
> #####
> a2 := 432;
> d3 := 125;
> a3 := 20;
> d4 := 430;
> nx := 1;
> ny := 0;
> nz := 0;
> ox := 0;
> oy := -1;
> oz := 0;
> ax := 0;
> ay := 0;
> az := -1;
> x := 125/2+215*sqrt(3);
> y := 215-125*sqrt(3)*(1/2);
> z := -452;
> r1 := (((c1*c2*c3-c1*s2*s3)*c4+s1*s4)*c5+(-c1*c2*s3-c1*c3*s2)*s5)*c6-
((c1*c2*c3-c1*s2*s3)*s4-s1*c4)*s6-nx;
> r2 := -(((c1*c2*c3-c1*s2*s3)*c4+s1*s4)*c5+(-c1*c2*s3-c1*c3*s2)*s5)*s6-
((c1*c2*c3-c1*s2*s3)*s4-s1*c4)*c6-ox;
> r3 := -(((c1*c2*c3-c1*s2*s3)*c4+s1*s4)*s5+(-c1*c2*s3-c1*c3*s2)*c5-ax;
> r4 := (c1*c2*c3-c1*s2*s3)*a3+(-c1*c2*s3-c1*c3*s2)*d4+c1*c2*a2-s1*d3-x;
> r5 := (((c2*c3*s1-s1*s2*s3)*c4-c1*s4)*c5+(-c2*s1*s3-c3*s1*s2)*s5)*c6-
((c2*c3*s1-s1*s2*s3)*s4+c1*c4)*s6-ny;
> r6 := -(((c2*c3*s1-s1*s2*s3)*c4-c1*s4)*c5+(-c2*s1*s3-c3*s1*s2)*s5)*s6-
((c2*c3*s1-s1*s2*s3)*s4+c1*c4)*c6-oy;
> r7 := -(((c2*c3*s1-s1*s2*s3)*c4-c1*s4)*s5+(-c2*s1*s3-c3*s1*s2)*c5-ay;
> r8 := (c2*c3*s1-s1*s2*s3)*a3+(-c2*s1*s3-c3*s1*s2)*d4+s1*c2*a2+c1*d3-y;
> r9 := ((-c2*s3-c3*s2)*c4*c5+(-c2*c3+s2*s3)*s5)*c6-(-c2*s3-c3*s2)*s4*s6-
nz;
> r10 := -((-c2*s3-c3*s2)*c4*c5+(-c2*c3+s2*s3)*s5)*s6-(-c2*s3-c3*s2)*s4*c6-
oz;
> r11 := -(-c2*s3-c3*s2)*c4*s5+(-c2*c3+s2*s3)*c5-az;
> r12 := (-c2*s3-c3*s2)*a3+(-c2*c3+s2*s3)*d4-s2*a2-z;
> r13 := c1^2+s1^2-1;
> r14 := c2^2+s2^2-1;
> r15 := c3^2+s3^2-1;
> r16 := c4^2+s4^2-1;
> r17 := c5^2+s5^2-1;
> r18 := c6^2+s6^2-1;
> GB := Groebner[Basis]([r1, r2, r3, r4, r5, r6, r7, r8, r9, r10, r11, r12,
r13, r14, r15, r16, r17, r18], plex(c1, s1, c3, s3, c2, s2, c4, s4, c5, s5,
c6, s6));
> math_operations := cost(GB);
> V := {additions = 1, divisions = 4, functions = 8, multiplications = 1}:
> total_cost := eval(cost(GB), V)flop;

```

```
> #####
> # Saídas
> #####
> GB := [1029383056*s6^4+232922400*s6^2*sqrt(3)-662611528*s6^2-
58230600*sqrt(3)+101316441, -33113421124371848*s6^3*sqrt(3)-
59941592980713600*s6^3+7132677097292405*sqrt(3)*s6+1145678183800557*c6+1498
5398245178400*s6, -50000847174814957273733556*s5*s6^3*sqrt(3)-
83329034622291539548805000*s5*s6^3+12500211793703739318433389*s5*s6*sqrt(3)
+271600894150995002759880*s6^2*sqrt(3)+1721369565654543565045175*s5^2+17241
252250419522556614150*s5*s6+172483214421692292404000*s6^2-
67900223537748750689970*sqrt(3)+31012833316714527147375,
48433939745656366469286264*s5*s6^3*sqrt(3)+80158604144767261635652000*s5*s6
^3-12108484936414091617321566*s5*s6*sqrt(3)-
271013981331775935559020*s6^2*sqrt(3)-16281629689435465926516600*s5*s6-
172110488809132553966000*s6^2+67753495332943983889755*sqrt(3)+1799757773985
406517092800*c5-1835983051175891602706700, s4, -
122189312821784*s6^3*sqrt(3)-
241937186380500*s6^3+30547328205446*sqrt(3)*s6+5187820067925*c4+70859936730
975*s6, -10081112585503797404487960*s5*s6^3*sqrt(3)-
16812491183097501116544800*s5*s6^3+2520278146375949351121990*s5*s6*sqrt(3)+
54202796266355187111804*s6^2*sqrt(3)+3485002391985624573672600*s5*s6+344220
97761826510793200*s6^2-
13550699066588796777951*sqrt(3)+361625748075207262969344*s2-
11171070621288537935844,
862948876070339984225935752*s5*s6^3*sqrt(3)+1724396876870665502898380000*s5
*s6^3-
215737219017584996056483938*s5*s6*sqrt(3)+131983507067696425096968012*s6^2*
sqrt(3)-512246824845795205441981800*s5*s6+83817616362589665915959600*s6^2-
32995876766924106274242003*sqrt(3)+38874767918084780769204480*c2+1961939872
3416998379703500,
401003818684458834060852168*s5*s6^3*sqrt(3)+801310195456268591461420000*s5*
s6^3-
100250954671114708515213042*s5*s6*sqrt(3)+30624579890490680718169260*s6^2*s
qrt(3)-238036039640672238201196200*s5*s6+19448485235431978598158000*s6^2-
7656144972622670179542315*sqrt(3)+194373839590423903846022400*s3+1399212407
9444550518381100,
401003818684458834060852168*s5*s6^3*sqrt(3)+801310195456268591461420000*s5*
s6^3-
100250954671114708515213042*s5*s6*sqrt(3)+30624579890490680718169260*s6^2*s
qrt(3)-238036039640672238201196200*s5*s6+19448485235431978598158000*s6^2-
7656144972622670179542315*sqrt(3)+9040643701880181574233600*c3+495148037756
4368944147500, 2759224*s6^2*sqrt(3)-802100*s6^2-
689806*sqrt(3)+2156300*s1+1278675,
802100*s6^2*sqrt(3)+2759224*s6^2+75525*sqrt(3)+552100*c1-689806]
> math_operations := 68*additions+207*multiplications+68*functions
> total_cost := 819*flop
> #####
> #####
> #####
> # Solução da Cin. Inversa do PUMA 560 (Gröbner -> Solve)- 2º Caso - 2
> # Ordem lexicográfica (c1, s1, c3, s3, c2, s2, c4, s4, c5, s5, c6, s6)
> #####
> with(linalg);
> with(Units[Natural]);
> with(codegen, cost);
> #####
```

```

> #Entradas
> #####
> c1 := cos(theta1);
> c2 := cos(theta2);
> c3 := cos(theta3);
> c4 := cos(theta4);
> c5 := cos(theta5);
> c6 := cos(theta6);
> s1 := sin(theta1);
> s2 := sin(theta2);
> s3 := sin(theta3);
> s4 := sin(theta4);
> s5 := sin(theta5);
> s6 := sin(theta6);
> Eq := 1029383056*s6^4+232922400*s6^2*sqrt(3)-662611528*s6^2-
58230600*sqrt(3)+101316441, -33113421124371848*s6^3*sqrt(3)-
59941592980713600*s6^3+7132677097292405*sqrt(3)*s6+1145678183800557*c6+1498
5398245178400*s6, -50000847174814957273733556*s5*s6^3*sqrt(3)-
83329034622291539548805000*s5*s6^3+12500211793703739318433389*s5*s6*sqrt(3)
+271600894150995002759880*s6^2*sqrt(3)+1721369565654543565045175*s5^2+17241
252250419522556614150*s5*s6+172483214421692292404000*s6^2-
67900223537748750689970*sqrt(3)+31012833316714527147375,
48433939745656366469286264*s5*s6^3*sqrt(3)+80158604144767261635652000*s5*s6
^3-12108484936414091617321566*s5*s6*sqrt(3)-
271013981331775935559020*s6^2*sqrt(3)-16281629689435465926516600*s5*s6-
172110488809132553966000*s6^2+67753495332943983889755*sqrt(3)+1799757773985
406517092800*c5-1835983051175891602706700, s4, -
122189312821784*s6^3*sqrt(3)-
241937186380500*s6^3+30547328205446*sqrt(3)*s6+5187820067925*c4+70859936730
975*s6, -10081112585503797404487960*s5*s6^3*sqrt(3)-
16812491183097501116544800*s5*s6^3+2520278146375949351121990*s5*s6*sqrt(3)+
54202796266355187111804*s6^2*sqrt(3)+3485002391985624573672600*s5*s6+344220
97761826510793200*s6^2-
13550699066588796777951*sqrt(3)+361625748075207262969344*s2-
11171070621288537935844,
862948876070339984225935752*s5*s6^3*sqrt(3)+1724396876870665502898380000*s5
*s6^3-
215737219017584996056483938*s5*s6*sqrt(3)+131983507067696425096968012*s6^2*
sqrt(3)-512246824845795205441981800*s5*s6+83817616362589665915959600*s6^2-
32995876766924106274242003*sqrt(3)+38874767918084780769204480*c2+1961939872
3416998379703500,
401003818684458834060852168*s5*s6^3*sqrt(3)+801310195456268591461420000*s5*
s6^3-
100250954671114708515213042*s5*s6*sqrt(3)+30624579890490680718169260*s6^2*s
qrt(3)-238036039640672238201196200*s5*s6+19448485235431978598158000*s6^2-
7656144972622670179542315*sqrt(3)+194373839590423903846022400*s3+1399212407
9444550518381100,
401003818684458834060852168*s5*s6^3*sqrt(3)+801310195456268591461420000*s5*
s6^3-
100250954671114708515213042*s5*s6*sqrt(3)+30624579890490680718169260*s6^2*s
qrt(3)-238036039640672238201196200*s5*s6+19448485235431978598158000*s6^2-
7656144972622670179542315*sqrt(3)+9040643701880181574233600*c3+495148037756
4368944147500, 2759224*s6^2*sqrt(3)-802100*s6^2-
689806*sqrt(3)+2156300*s1+1278675,
802100*s6^2*sqrt(3)+2759224*s6^2+75525*sqrt(3)+552100*c1-689806;
> sys := {Eq};
> S := evalf(solve(sys, {theta1, theta2, theta3, theta4, theta5, theta6}),
5);

```

```
> math_operations := cost(Eq);
> V := {additions = 1, divisions = 4, functions = 8, multiplications = 1};
> total_cost := eval(cost(Eq), V);
> #####
> # Saídas
> #####
> S := {theta1 = -2.6180, theta2 = 1.5708, theta3 = 0., theta4 = 0., theta5
= -1.5708, theta6 = -2.6180}, {theta1 = -2.6180, theta2 = 1.5708, theta3 =
0., theta4 = 3.1416, theta5 = 1.5708, theta6 = .52361}, {theta1 = -2.6180,
theta2 = 3.0917, theta3 = -3.0486, theta4 = 3.1416, theta5 = 0.43080e-1,
theta6 = .52361}, {theta1 = -2.6180, theta2 = 3.0917, theta3 = -3.0486,
theta4 = 0., theta5 = -0.43080e-1, theta6 = -2.6180}, {theta1 = -0.42214e-
1, theta2 = 1.5708, theta3 = -3.0486, theta4 = 0., theta5 = 1.4778, theta6
= -0.42211e-1}, {theta1 = -0.42214e-1, theta2 = 1.5708, theta3 = -3.0486,
theta4 = 3.1416, theta5 = -1.4778, theta6 = 3.0994}, {theta1 = -0.42214e-1,
theta2 = 0.49876e-1, theta3 = -4.7424*10^(-6), theta4 = 0., theta5 = -
0.49871e-1, theta6 = -0.42211e-1}, {theta1 = -0.42214e-1, theta2 =
0.49876e-1, theta3 = -4.7424*10^(-6), theta4 = 3.1416, theta5 = 0.49871e-1,
theta6 = 3.0994}
> math_operations := 68*additions+207*multiplications+154*functions
> total_cost := 1507*flop
> #####
```

# APÊNDICE K

```
> #####
> # Determinação de uma Base de Gröbner para um manipular 6R com pulso
> # não esférico
> #####
> with(Groebner);
> with(codegen, cost);
> #####
> #Entradas
> #####
> a2 := 432;
> a3 := 20;
> d3 := 125;
> d4 := 430;
> d5 := 20;
> nx := 1;
> ny := 0;
> nz := 0;
> ox := 0;
> oy := -1;
> oz := 0;
> ax := 0;
> ay := 0;
> az := -1;
> x := 452;
> y := 105;
> z := -430;
> r1 := (((c1*c2*c3-c1*s2*s3)*c4+s1*s4)*c5+(-c1*c2*s3-c1*c3*s2)*s5)*c6-
((c1*c2*c3-c1*s2*s3)*s4-s1*c4)*s6-nx;
> r2 := -(((c1*c2*c3-c1*s2*s3)*c4+s1*s4)*c5+(-c1*c2*s3-c1*c3*s2)*s5)*s6-
((c1*c2*c3-c1*s2*s3)*s4-s1*c4)*c6-ox;
> r3 := -((c1*c2*c3-c1*s2*s3)*c4+s1*s4)*s5+(-c1*c2*s3-c1*c3*s2)*c5-ax;
> r4 := -(c1*c2*c3-c1*s2*s3)*s4+s1*c4)*d5+(c1*c2*c3-c1*s2*s3)*a3+(-
c1*c2*s3-c1*c3*s2)*d4+c1*c2*a2-s1*d3-x;
> r5 := (((c2*c3*s1-s1*s2*s3)*c4-c1*s4)*c5+(-c2*s1*s3-c3*s1*s2)*s5)*c6-
((c2*c3*s1-s1*s2*s3)*s4+c1*c4)*s6-ny;
> r6 := -(((c2*c3*s1-s1*s2*s3)*c4-c1*s4)*c5+(-c2*s1*s3-c3*s1*s2)*s5)*s6-
((c2*c3*s1-s1*s2*s3)*s4+c1*c4)*c6-oy;
> r7 := -((c2*c3*s1-s1*s2*s3)*c4-c1*s4)*s5+(-c2*s1*s3-c3*s1*s2)*c5-ay;
> r8 := -(c2*c3*s1-s1*s2*s3)*s4-c1*c4)*d5+(c2*c3*s1-s1*s2*s3)*a3+(-
c2*s1*s3-c3*s1*s2)*d4+s1*c2*a2+c1*d3-y;
> r9 := ((-c2*s3-c3*s2)*c4*c5+(-c2*c3+s2*s3)*s5)*c6-(-c2*s3-c3*s2)*s4*s6-
nz;
> r10 := -((-c2*s3-c3*s2)*c4*c5+(-c2*c3+s2*s3)*s5)*s6-(-c2*s3-c3*s2)*s4*c6-
oz;
> r11 := -(-c2*s3-c3*s2)*c4*s5+(-c2*c3+s2*s3)*c5-az;
r12 := -(-c2*s3-c3*s2)*s4*d5+(-c2*s3-c3*s2)*a3+(-c2*c3+s2*s3)*d4-s2*a2-z;;
> r13 := c1^2+s1^2-1;
> r14 := c2^2+s2^2-1;
> r15 := c3^2+s3^2-1;
> r16 := c4^2+s4^2-1;
```

```
> r17 := c5^2+s5^2-1;
> r18 := c6^2+s6^2-1;
> GB := Groebner[Basis]([r1, r2, r3, r4, r5, r6, r7, r8, r9, r10, r11, r12,
r13, r14, r15, r16, r17, r18], plex(c6, s6, c5, s5, c4, s4, c2, s2, c3, s3,
c1, s1));
> math_operations := cost(GB);
> V := {additions = 1, divisions = 4, functions = 8, multiplications = 1}:
> total_cost := eval(cost(GB), V) flop;
> #####
> # Saídas
> #####
> GB :=
[1175262138064162505664198980608584518451937095726344321*s1^11+291012009866
0049753263028747695997083011770034930582000*s1^10+2362053508278532447802867
82923617524667297708528815600*s1^9-
4734749428568907931203608683602748172679010491784560000*s1^8-
3795196768655369223929571145636498266957223725005600000*s1^7+76209076484127
9462235536831923509016134687212288000000*s1^6+22383029987510793324995255080
95909448387768222016000000*s1^5+1126569760199489107367796576802269937236141
47840000000*s1^4+24479503443709599747939978792596886789046927360000000*s1
^3+2251636012078674409085238869177208852971520000000000*s1^2+7245140171869
1702961182334983631667200000000000000*s1, -
283052426792554008658514169501696018630425161770570947535393698853216259827
674663198215607942306033565665918109653664765859648675790576135722397283024
405830749239177219873979934251437028982370662008653153907091410148499576064
4785025069263199251*s1^10-
635479207058847345758669374875124482703358127674277234515918851427921084345
429598178184533827302630982812222472200151174320068493886540653779568418693
286567428669598051746688082991196085579479541247367805453294883373895270374
9607748938467892000*s1^9+93182198003350327362631982166935012957631186381000
199074086507125147704377236852958900924500982170466281337044161486262626708
340049955854200964129226220926508043036202216253708891212940959795831238674
1268733671553929319577581979741978478730150*s1^8+11249885525822857744335602
203026983938352224551742598592303054125857937875502993373175533516742207884
713882685494203582195337183856193301668385300622995367304747882402539432734
275299539981149891953883820647940253709628596534010973971231679360000*s1^7+
651189623397073113727918835706579360211982590007860953985179941286277672038
052762545641886201099179693917206491192464399742771486800025365475394333832
404347099777940851347396636643720591607703216320825174573117505480077797880
9034063342097350000*s1^6-
346064862810178634708480929912601878277188980644895027892219952459970051484
259343415360066489816450310304244684574417132032875415422327570613074297212
013484756540628391301471441934258484838223977699182047668631451017981620420
7486388659328000000*s1^5-
463354336579047181166230550950592588450377946160221632194240915097545657529
596531129446810612426471000949652277889482460313834144769430876716981118153
119735530636430564722936124794035183712649713796568431535597531020999145293
2472822466608500000*s1^4-
158835585885749367032601484036841178783780881037544688567243980538256589769
400212202299020071550805930112270847361855389242464378865883751808178296982
294395257022712306076920300975360099246102806868393852027825527439153632417
7859470950400000000*s1^3-
180221227765503853142259295293707410256694886409625875888383046294881275793
813865469901377363438143089135441704922822694343204217821422113069334122946
961777756921087534256663140622791462768659432645442991308011005425227449268
000076742850000000*s1^2+131201282638551012386923693699872286259953436365277
026412222048585319953556366958877995772053618963906823940678846987235990016
242205007202440399767363387752006342221472679742174016656567534724316215312
6595418807603370345424949398437500000000*c1-
```



653391069948016616533671522145717128721438108120886386243125664643333353735  
616636765157568497574607674668105421920268860697388094831930889932986885996  
247285028628182857821442758079833703104323721795336914799565713673095465266  
0531200000000000\*s1-  
131201282638551012386923693699872286259953436365277026412222048585319953556  
366958877995772053618963906823940678846987235990016242205007202440399767363  
387752006342221472679742174016656567534724316215312659541880760337034542494  
9398437500000000,  
876725743464079477511344180357144697327104025283766499818432699107580918755  
00182961850731222655603753507151640218312767989515873017\*s1^10+149237571151  
279833229391548026652794137547854433391117053438502563519700868088446658202  
634473273824093689712043408074945811101112000\*s1^9-  
743383246631069181362993652300839528590812037717224506528097433585269758029  
95632416317268658335610764981779941494168185641548471800\*s1^8-  
239966155772204989830731220471781300526894182608524098848559402943128291412  
310784047529402418332144022459709781695828914153240576000\*s1^7-  
104175988377018752376212163303255753175298651383390178752228138574427095124  
568259019089262407478854125891077929050118100350634000000\*s1^6+353773212924  
734114967140956393315186250790799954208925749232648249255237999926276787083  
42426473545024359267119168664495122326400000\*s1^5+2216198180955176999683025  
766045282902661584368744049832856165444756223105917040396688805286943666491  
3300183771850836833060531200000\*s1^4\*s3+38844017338442678954007192990361014  
833745225422830829080106948126763168998935199804318169697645759830000681440  
848135003610624000000\*s1^4+232602570436806004731533524572274954140648991699  
285865919357536845899262030312280116319675867750571005570152298960619529728  
00000000\*s1^3\*s3+1058015380294093309533322693804877836190826111721956546498  
8812182990512575047338629382210672478056286473434928227458308505600000000\*s  
1^3+697622117748909436544873100299715687705507468562803218242011561541899245  
88898402231608664970485377876292684828235463893227520000000\*s1^2\*s3+1070800  
765265894095066700827283496847307590202520530192689523160081059881307434858  
248398031842218605149387504312961033830400000000\*s1^2+453692162149355274892  
114301001516194934600432425265819681186303169929224826805296308691647964070  
050120232128350400829440000000000\*s1\*s3+36187276459371972874397712979813379  
593048311748142591975945366745538906210685681091952873882515716035823793453  
190348800000000000\*s1,  
626081198172040227469794483219851288076758513747635496229318228041211344381  
663430954191716729258046886677736223538810601113096202264671344469348114034  
543425197103189544441845284216782163820554707066291027156413583733746603\*s1  
^10+13768703652547816590591011879058547809844073302538842635198569605234194  
546613846469260745405647386216278027021939984710706397726233397190822036511  
443027081070915382573329604268944468836145789890992642728162066744455385453  
15\*s1^9-  
33349993997384085338982813737918022143056448107607555822218627677506437758  
009704023699776566081753441029984039017106756784534028224491509875783106513  
067919719964349609572172444699228976166637753153498032127688168738580500\*s1  
^8-  
247783264931835917973995753435462079343126305113867375764016345922029166766  
488333893574426869832409585372292015072626834678333842914770714596865768413  
3497374690122107984957064859130023191371517017316803878494621226624420000\*s  
1^7-  
117731857853565546757437977508844893546770536476116123187313082021489978113  
110698397906312974330859850607041946324551034099415946549968396001065691392  
0244545173481871058804405356662080106670401311088249550812000099302400000\*s  
1^6+85223909663778781139136912862622020423503951524887821065835654894811006  
471917497846145976401805170757597924866962752840117124484825298128955142247  
912921476871436229627677458014175178747212573074974156715040211734778200000  
0\*s1^5+91701866144980251253973002045778435205506901473231842384243854909260  
713007448820585980012234123387927095634580940678691588297118282059779308249  
829381229205342994941524772595230745862038632830244902537062500537993284800  
0000\*s1^4-

338483872126896271619073921355358639784894605137602262889721295523352969911  
 919928193859561565429374959122097700188837070435091583539164288308914651037  
 71309047111908000365796497018065221677090310800967905468140000000000000\*s1^  
 3\*s3+2950877555412530379558151441607379746778824487301023289765067040493038  
 824011993875464693477264434887928042256723107882957659578445252376843788821  
 851422554912622550318269588680248628455983414076834490870264527746662400000  
 00\*s1^3-  
 206049654056785492357407546643788855672036673376260998841701152270252066818  
 935044455930744720852660206668514536085491332763500526033714246160677532789  
 4683572531070548178809451968442703610490457829549606903280000000000000\*s1^  
 2\*s3+1738201133811633623191628878153651965041249570127625957347156346733665  
 453022520851435369942800626678832079855658215080605198618944703835940705595  
 848096655908675750435719892489461631428515755192423593844931033600000000000  
 0\*s1\*s3^2+37678484658660510780950512682434884842364623496041297731747082182  
 059724806710225412595154639742019488559124876248506681866098409133243625931  
 614426720968369498205823409263722476564367833338229913947836003098705920000  
 00000\*s1^2-  
 892201493408564695449081676085625790693289534135447714487598808070013994553  
 854498446602373258418004010434087489951660822898027148261212764607596098133  
 377622036583468997587767256761453178922820526019191681536000000000000\*s1\*s3  
 +15745639723184635366870703421806136953056191024241998497634597479589392665  
 576878972907268045297434151055468017632643985236713081075427665131352533252  
 6300522188902813444155770683439642403662704982821628817408000000000000\*s1,  
 466423985101030474686309172727568303566867338512131804388383032464879463122  
 223707783032130634404524834196718315829528415984627546643144990139916631113  
 941728827273497678515297651343510355251874415392243398917734682245249208913  
 703843562\*s1^10+56631016012465763438878025205095821025306501266303658581011  
 014507323164449817818205441938148896683520453218622439848710671498700579508  
 016837496855902293044079436062685045799802156175332954331985442711837563300  
 2346865216411742620613375\*s1^9-  
 631833837926023573651779624595736342677542414087494664685537887621655534743  
 008828661693868861708820634131891081424063573540264262764667128242592400851  
 686471779784168497633179260920700330881757068120638657612879774028003701623  
 495050550\*s1^8-  
 960382439593470930566956825430058637054197783365690663749962269925008134692  
 965655424574986812548125020450269529562102612442754299934551012273943872339  
 821080270645873671425034989217443502945376608376827516829246957088464542887  
 927320000\*s1^7-  
 403638773282148063707662589708175627260363233235763685177147916012473320351  
 660346024825286664618604837145251302341160817841999666523706518520716304677  
 017631693103845435220258755865385831255637748818555125700924234602666747512  
 85700000\*s1^6+3548203484618158017164212769668510843244758514425274136491578  
 931411287423536058144922013801823975960364584629104169649963502051930355828  
 694528951774907169708824775675214329577062023996046251665393254071245006919  
 98737846010935336000000\*s1^5+1477597190382769942635052555759295688768510323  
 976401715526341409049705328451053385466382921949222308798316913770977917824  
 491795408417413880378800714810261950216300526199161412878420206618532413709  
 00943944602107376917286431680352000000\*s1^4-  
 705106846733007465622142284194188294670382143898494850350552156796306798003  
 651561129057972166934189426331257785062152720809138429096501995770539412197  
 913985701487063277768697139277752987861344792246833453894271021446496000000  
 00000\*s1^3\*s3+926836950017763527448506499073624572599580372209023793230647  
 299912709380896139315801878075688476857183166353207565873671253287732876794  
 074260538880610160305947504889185388591354419993604369388473121326830995197  
 8096455634124800000000\*s1^3-  
 545053270227821251440163924973178789800477338922234242019506189784285831845  
 758996376081191396324091848684359596242796003556383772290502274940738265000  
 864204310498686645293760365991487466119581635886749168426315057893000000000  
 00000\*s1^2\*s3+165449083673586260473358406322698730818043936679230443603324  
 271426090687153150786963192213269836996312924738877696368111476613910051521



361727528396683814900233625600000000\*s1^2+103236019238542523090322649169934  
 276112715730101749960784236932927350756686247722189278787921721332060961458  
 597462772028697591273525693236733572763758279831062416476447863882915974285  
 220341785270601967165615799145240533197342950006695078825466178893228500351  
 1999996786391040000000000000\*s1\*s3+1001192501970958137482317518351043446272  
 593548672032770396577521505409712117465318156052072714590101350202393548080  
 942090812260145560284542234203979481150380278896804438522754813374929020849  
 880800830679707965141909834794977480190576909123782663908862407418566503821  
 60244736000000000000\*s3^2+432247167607537242302133844943785621704303744704  
 601304002839728658568683051253240434345201387842461457183979945204939920480  
 198659713020849318598588011289964502491874555217595170372338487145118532468  
 723025565090948660461943866245257165298988737791198017234135565600260096000  
 000000000\*c3+1933861786047083097098702095162630440155806291971473666197624  
 288709579363475291193789710851684921310126705779562555434211672545465992840  
 769670812252411613310185909541939006291614257979907013355379385440820996771  
 2881418066515949392582093697005873180577989471439696591087288320000000000\*s  
 1-  
 432247167607537242302133844943785621704303744704601304002839728658568683051  
 253240434345201387842461457183979945204939920480198659713020849318598588011  
 289964502491874555217595170372338487145118532468723025565090948660461943866  
 24525716529898873779119801723413556560026009600000000000000,  
 155591583513456782655249078265564499018303173862432252234212138160584943804  
 905586365557582552234823213313664325448950548266802618108936761256435861588  
 036000601314821236462093607259195958158745880611528099394151676995579690316  
 924932022550334171168543896448847\*s1^10+27068956720022079402390696893388079  
 561431127894461270659287074361253715887016545447523594747759061603640495953  
 413760787024925842533372188365422826197353822776826281994332316782079995280  
 1941991399633258621983066885081110207468038783203841987328889559003074000\*s  
 1^9-  
 191237543432370413669298844880036975063712445632958578780083662831434729272  
 503070178261382983399375605153512242592311687428850016590586959676712435322  
 086673181401863903039252111504819634903932550090723705559206220764005436198  
 391520039233405803118039961705800\*s1^8-  
 551171702195413598251704324642986713190997967388509425899531511099924353924  
 100520688908505193768869463908394358737615163118410880626385777591565200842  
 577175139257825165232738934378111013081440663353747434288259333334209858905  
 425549262321351051647755887920000\*s1^7-  
 125436648513084459198531217734683312281770810817064721913277680695473542781  
 647679934998186877720151985682946155904287871165301189125263481463484173981  
 708451537851925812843248246230359905520307518456324775706043958346072799804  
 838580122894920333210222299200000\*s1^6+285262904611430237659659848545303562  
 117998492707979627649830361566121166599079285051337102476116600177775392706  
 84190042515388923442232322889025748406226527874828352084123107228179186123  
 194230466812388185847137404922611116018466122431773868385287196816000000\*s1  
 ^5+205407971288481922444186905025819404188305301210046589716297366088362956  
 336753707887665427901083080787176825405398368746479028979573485350718292922  
 334197491543373988968478999155836654015865772869946522629073946616089611949  
 083554298226893396178414299712000000\*s1^4-  
 205413323741625377874704567201900970335118472244983024787787949634867421287  
 115103696900882987746404959604610506374855699889072739085775588522456568476  
 958744867911645822236939216166917814302834927389695993519059816961342024523  
 86964656960182889111040000000000\*s1^3\*s3+4485954845589300077774672123794057  
 115585121287655398671478669834931701281646759599341011417316760443809203888  
 032128645154318886601311208525817665628229273261255492377359729965516453082  
 496965783461041953615506456795405015041875188002066077769501818880000000\*s  
 1^3-  
 159948411382686397812368940080392735173778161726414150057439161645916514275  
 446427887265878239153201303386601665672312879416541328923641718530405422299  
 661857971608932497647648666962743105738828131403686633789731468536020419588  
 90862190374145760320000000000000\*s1^2\*s3+1688367784594310763239197295208234

417954098282828213434402077012393061390722878184238551168974617365329422737  
183272223081294945828296468706103656623720754633394164856578892269822780292  
170036142396677380490986432805201567939965420926847861787898675200000000\*s1  
^2-  
231910183339138720926918533485704714190096336308228359250637451924458545905  
117760876958838986575474160362021835791012413386325116001832112739788276309  
301445154501504035304469351393261952547831945757084708511185244203133590159  
9213525756996592384000000000000\*s1\*s3-  
175018491377841080183152443399687398760772415846362003461387359156455349211  
718972313335724814751815511613641237114765282152808025386336446842650929002  
898508037494638285158249030373365985320100462334796957800471461053314217333  
0332780417550592000000000000\*s3^2-  
653330835624693318738066693566485607316095527837610735788446434357152303241  
573742318354406494576008618619327463610470561955812068826308081736837380387  
645002293793959266647519377981897334511226787962663010106192701392329337924  
36965352717359513600000000000\*s1+151607271756371167223408872873381148174602  
113443481063086097420699749174707770252442048513106944156059884073008310954  
954811642499082740498715688344085866300422540392743303460498258722781654823  
260242985407650066191945635685588102599210855168000000000000\*s2+670828636  
090137908068180853422040478648681917891509128699546109290925551804293152398  
444748260814849822495898266862632543414347341074073003166762584450735842577  
612357271254249992295233547145235664791970830310027397989538431805763713322  
368000000000000000\*s3, -  
407735534661312917927910981334532463965509286806969149268139095866903691058  
110416790942514727889588115453392511114333051838946744633341595816333054704  
996376506067421353592128475133566459562886132338819426438128918340029319170  
476177131062520860957241551608548641221244305165417754123458839538438071341  
54389973\*s1^10-  
450895717596113050884114494323014230068493000027826898900762611418912741458  
595147924346864769934273464658915266067630727207968206287261525472269198592  
553464864900636080319099841688492368560281380811056013754665983847039550602  
984322636554096420853201916651333699985053872929293693094408691138805708320  
26416000\*s1^9+9210448896127059246780058214542623986659572492637316660224083  
103900363594442620146003007850526711100186601121997231022011586149349186806  
741210274501278336297701908573576921427451583842829609170458956060686162125  
362509444073914084738023824213403536709521578698184031207573835644835937106  
6204973130948612717200\*s1^8+10981482357832984809671142365684635207255407020  
847859389455578332465379392591154685909607163886075730542069678457070626209  
205820840502092107206317839140770572756619281241880600171703368299665278593  
630385098604850480668421355627857293054273076235395623848519771587780215218  
6745218860506581204531811769767280000\*s1^7-  
467430294237015707485881709535668880170755152427021641379946148225455179106  
565741571419875825691161665939026864651933265690625646719958010709509041676  
156515114987543441449238833138266635577878196305338033053091832564031464841  
939902756827474512500399077482826402144732707525699515060038956153417832272  
47200000\*s1^6-  
824989326327097346880206594980514104491572005999883574078524423440184777239  
293329783665619485522816003555623174567376263159581073611702942159300373353  
246953560665497382248379473923800688509794272393391582107450643962245596390  
553070876612112692146265523285052049472619627586423753849689844597878809261  
44000000\*s1^5-  
195428900036703633258409775904443400006442860072040802362583439850472845513  
508123049863043096849991330068751474217796957427115324524838856188429408229  
597151924018004066662360642145662855897615031492349814328698402145312059216  
674576082967356487885472881783266281703509825336016453199078400484958407710  
08000000\*s1^4+5609620027009032411816345766106525904264874015744078732788291  
616914445384071210986273099079819637096823673002606074650792210158756431202  
839831698897889202830636102485718048643660321615416733569262270840907405833  
730957678332601550606945322890666648257853348230550329459203570478001481692  
5354240000000000000000\*s1^3\*s3+512880558630488048169266784218102623583800600

225395089667702794651743811150724721984623710282089599658375332167256783566  
 549759740732123106019643984185540412006069847242577318444774605971603628798  
 687635001680031280407108105155310028722892014465227545222765086329358331917  
 2483087175947252240976965260800000000\*s1^3+43362814509933511745205033751765  
 185213137117605554728983992914973629681656753177167611183971734756734055152  
 470013812026321296125489409910070646650248273081773096584935934050722048781  
 282218545398272424535076996180906163422751057035121673704415847646410432395  
 32216494319450510203388395194092000000000000000000\*s1^2\*s3+2169808333416525  
 102157203511673338450274088096716064322245880418359090234206996064091368269  
 579399079719002388593142515073421963419616289258613410451507578377110857947  
 422072639996664712648577842354618870372400018164259421090461835803573594265  
 271484853102914905381196071844045917035124766807324909363200000000\*s1^2+821  
 449241359189778809666940097680906754205544805104350692773938033149941932752  
 069288479445107803228273557576306875837217671998834967532154040518286257792  
 644105401431938347362346843354633631343166580159279183678824634617062540704  
 05489221312722964023238534952407254298946069151862172312135040000000000000  
 000\*s1\*s3+76192866910134857572594664387449997144978497270295698249402130110  
 833833813844758339333189104664818942585072037423356113403693463319304776429  
 004177218268148708286892893729729704841996710143581467936432928950794629020  
 623749968642564021052599524371411884139637898096368929198504840190514972160  
 0000000000000000\*s3^2+69092476337036924371258658183277335745133964995054863  
 710947418061070952355224221103105996127787330920276316557309396364609182025  
 554398759335204775593397702650150536911596365736560742909457011515630886199  
 749278552784182115046402920046516084843168691580094725120980750394630896826  
 80004435968000000000000000\*c2+137296200227835604835225599149603965671535445  
 747107633210871048982043257692299099449030691849613147831560453757562344766  
 554065044812043864066711023808452648426346099185791984925107439352340250139  
 251810491638343274675937218077726591913980412867448431706663620242520162013  
 72727637525145054248714240000000000\*s1-  
 690924763370369243712586581832773357451339649950548637109474180610709523552  
 242211031059961277873309202763165573093963646091820255543987593352047755933  
 977026501505369115963657365607429094570115156308861997492785527841821150464  
 0292004651608484316869158009472512098075039463089682680004435968000000000  
 0000, -  
 427066771541665855288021460301220298353283718755254469584030604911180878677  
 970329920934574015985408168062131299692047891875639389639278512846692680131  
 257992190089490093902003350259372070654033461247011571862258797144283549349  
 119140212140398473869667555834027482761497303363705752983977609027763106172  
 75256392659\*s1^10-  
 786940313480442508109098028411100094826159370060526825340898754482918983910  
 089571701947898498643911204236796758674556021775010795304121196776682983651  
 492305393163713048392063099605840283706599904141574748700983887329293100448  
 952728022998272139294554975467973536222109025250863693976761821093280509450  
 53219488000\*s1^9+4868876870017733317092930785758842135311261205043302656816  
 120314660001079705603266574341924608886745114862997944261463088538749538008  
 774918042093262475045448452660506263275283978333061144617955344551252491611  
 657534602166872205515943578184014612456112667974123369533121128106599410424  
 7468102870599045869601487600\*s1^8+16020166004709279039295960697003252812441  
 283023208290901178428371133157385559677278119323865268164962537095606326397  
 317876319811000651760578670940787341643557615468324684277285450730687012095  
 844419082277459361744738494107180677214263630166983210343721463542232979034  
 0184148464363692765270537110392383513012240000\*s1^7+42007344506149288714779  
 357973694889926354009942547230463375609853104498116465173376057541244023657  
 075722978385365834315458325442001510025545197240595144216166352179181374741  
 828232811413878538453581239587190549601010494466578015711371250969189699930  
 076343910368783424115762404778214150986179983783527899202400000\*s1^6-  
 816813001027970080036656549440952868809898717171088344070781346735243745535  
 562975818002677886666401313216761988816540184449859215006412211856102341390  
 900521770139501308049631916059623407427759024018749701797549961999238019072  
 930038308407500812950204897317778894184218640035323215928792911002569949650

17152000000\*s1^5-  
596762419402515396134163009994378722511181166496168194397176380025828283553  
306258933930197475325728491446823360671020271665740375023003396274913160875  
679958204961046288451245895967598576412015928233868255854402081308656978639  
551721876012073485048787428258102514687120168553880259612934175249953745107  
20064000000\*s1^4+4758937661558096350012374118327479815605614577522794201355  
944090455616262010524705726902379430223748121449842667260506692750351377770  
310759477168670119026882321181978905964504959653228294626553444528156929666  
414234888014172934830045332290244052431180459038299391704534652085772350579  
724445663015526400000000000\*s1^3\*s3-  
122916748067262870493440673770961677759581565456924653240018796282837931497  
254390600009855380500657995838862992231485080425076911912283365142137587777  
533372216304819595088974606243994031185163206990056173349296798236650280313  
295197276994217005373130426780924263277766569507441938135342980193682056164  
7360000000\*s1^3+3678697132584737919540524750618506930883404592740332628107  
429928038369091846889464971434857185352600547177562263373290375666230383578  
140983965176311792727605507268803128690275530846976077407567627768570078427  
077579970087276953817741388948903598393659341688252075090305380581992908471  
283990827731200000000000000\*s1^2\*s3-  
151233265251426141509390105922841383186177158491589281840982562489478315347  
655264366594915430599624631485855198898515358453296403813140425904386550873  
229568001185694024445980862145865516819342354938738333614328499229402693260  
657551556242827759622469558904135707564651957514757112621838048440345211494  
400000000\*s1^2+696878881803143632619525510990579848524958098396873483389724  
192695735304330749127930340489763670063514181137359186643195938335092372322  
661376727918723673249422808550926662940364673872261641058628546070522410454  
427025625162676175522839664594264899160816112419152309485214537582790152154  
2305216614400000000000000\*s1\*s3+67583961139672397020785083148477960374045755  
757795527252094546720827855521532915255994209848568358993622714576400825262  
834416277344294066921314192723108450348498131818536074066649103918406818241  
327839147152020103697315138032995806342486043799067075081424291081072615115  
5121925326403533561221160960000000000000\*s3^2+60898843662282492014765373663  
689939693383335217455674013441979873302748510681045878929715715186050206836  
727650986526503657009032585902118656408106827502365704620914907295816488005  
304518057564777046180585521208410168093281398543287244824948740121056275958  
19169546481379907037947992058879593606840320000000000\*s1+62733088591600929  
776443790078457685830198974583598654545926940291324291147888081079498133264  
726161613076669763307835646020073392893786181923723913374930671667251368984  
286048243192908116384094643866100017863720765439493814040341493205113866883  
10597363197505702074738154273123106105850392473288704000000000000\*s3+30529  
022441218368065194437032152103784722153867125960222608717712832234710212545  
917240693621507563575700532658285036296297451737282505764674335926836206571  
483201570608201220721278202485578822974835785107745910622285369513292672647  
42539994383200247640143488993688395398074345008570030552754012160000000000  
00\*s4,  
283052426792554008658514169501696018630425161770570947535393698853216259827  
674663198215607942306033565665918109653664765859648675790576135722397283024  
405830749239177219873979934251437028982370662008653153907091410148499576064  
4785025069263199251\*s1^10+6354792070588473457586693748751244827033581276742  
772345159188514279210843454295981781845338273026309828122224722001511743200  
684938865406537795684186932865674286695980517466880829911960855794795412473  
678054532948833738952703749607748938467892000\*s1^9-  
931821980033503273626319821669350129576311863810001990740865071251477043772  
368529589009245009821704662813370441614862626267083400499558542009641292262  
209265080430362022162537088912129409597958312386741268733671553929319577581  
979741978478730150\*s1^8-  
112498855258228577443356022030269839383522245517425985923030541258579378755  
029933731755335167422078847138826854942035821953371838561933016683853006229  
953673047478824025394327342752995399811498919538838206479402537096285965340  
10973971231679360000\*s1^7-

651189623397073113727918835706579360211982590007860953985179941286277672038  
052762545641886201099179693917206491192464399742771486800025365475394333832  
404347099777940851347396636643720591607703216320825174573117505480077797880  
9034063342097350000\*s1^6+34606486281017863470848092991260187827718898064489  
502789221995245997005148425934341536006648981645031030424468457441713203287  
541542232757061307429721201348475654062839130147144193425848483822397769918  
20476686314510179816204207486388659328000000\*s1^5+4633543365790471811662305  
509505925884503779461602216321942409150975456575295965311294468106124264710  
009496522778894824603138341447694308767169811181531197355306364305647229361  
247940351837126497137965684315355975310209991452932472822466608500000\*s1^4+  
158835585885749367032601484036841178783780881037544688567243980538256589769  
400212202299020071550805930112270847361855389242464378865883751808178296982  
294395257022712306076920300975360099246102806868393852027825527439153632417  
785947095040000000\*s1^3+18022122776550385314225929529370741025669488640962  
587588838304629488127579381386546990137736343814308913544170492282269434320  
421782142211306933412294696177775692108753425666314062279146276865943264544  
2991308011005425227449268000076742850000000\*s1^2+24990720502581145216556894  
038070911668562559307671814554708961635299038772641325500570623248308374077  
490274415018473759236193569943810895702933289021597667048827089804319950890  
2888869652447093935648214589603582400641970557133218750000000000\*c4+8860078  
658968273463948571688531452501192426776750337730670313168557507747392268045  
226148308580535352338790364250276190195941341410180464704669455410814000972  
513595328019055263755098828857389142723037194229546948822224200614497874500  
00000000\*s1-  
249907205025811452165568940380709116685625593076718145547089616352990387726  
413255005706232483083740774902744150184737592361935699438108957029332890215  
976670488270898043199508902888869652447093935648214589603582400641970557133  
21875000000000, -  
930296262535146430028646528397046380074714876771894974677142216597744967513  
135570354703678054045464592230138922014818719367298449812314631223027889681  
621909185472711246558023435643636611722193873173590210650153504324419355327  
6782803953183319479405156839\*s1^10-  
151051847475036794751965660417562546832998107438528830420090921150734538256  
020614069315136574185208126770640277678121052007752080875815525262358057479  
415973689527815512042276955332154780681519309392006778099430571680359905567  
83331838541924262634397254000\*s1^9+1254865826426699406251394196675548977218  
468208192990467316648729058829085608011529003476387566217693388917533202087  
910100114952138684007198753964152293642255570959619731360825113497891189902  
0804302343774263558798664157599027280606818672616500471992265850\*s1^8+30162  
053432312879056316273163619288967494919421555365200066795116395195920287688  
615305814135595549786642351111751717843191780067309557102088862171223829774  
434091913406683310751103496239121802358255537916502896504675880367144497989  
845933470262966034040000\*s1^7+382310595898468736926125250684809674758060643  
230339690844370323488215827028302018588545068604663545824832076495417367328  
464482289241324720299959670262589934971173804635303744529950889738386285081  
7370342532599357048510258727538124567405038610396335400000\*s1^6-  
152075138312597115128907580269128420138229264388252354725139494615749831113  
725641992401778892059481330159871973338526837432809461791242623625543384895  
068341724276540317129219639489060686958339258314258632406500423738499680247  
67849504296906077782892000000\*s1^5-  
838678125668055914520944481481168684885286765713255920514301625498112555281  
249711429330361190851592039346153867899307695139493262313225100615709981690  
144836430480236650511002242471894168293043116576201554676871341157335182498  
3549880715497343862844000000\*s1^4+43928132724473449810622268749483010044612  
68185216343691371887821884629722221977097162450715318793577577451013888512  
151076644261008377116431619440727240357763402151422154741147211674645960914  
0599961646567151342975751041237349939929652548234240000000000\*s1^3\*s3-  
997994566205262381872529921736115553316878085892661800147026121692223845366  
798152935607564528996959460072884486192893168614987447632330520678162358151  
730231375932597274164713287274052233413088311366741453138350709419471694661



184940221856882841600000000\*s1^3+345681773856510250889761277112217759443178  
292908970988935969987545396006561188652546211449966336718742838271851235767  
749018730822040067562188090484094511617394040860074987906880042703245570765  
266491468409439000661564235581861466543115636408000000000000\*s1^2\*s3+153574  
615948195534743972389431111110189126261644805190074471407708585109488307874  
621084995255311542098622173307090439103430539022051337538283570441181616811  
716282585059258157000095871472515603337472843761283140410726840786471289828  
550431408537600000000\*s1^2+675282101104467873111145594932912453109043098553  
257070813636637391720745547139839779038356445910051089298266076626711241156  
979118810826644773289103699121240522712063667622049667194175168349488986033  
17803105942851732241031422957540488997504000000000000\*s1\*s3-  
235180901804889114088516022054978229602929931351939482201670762474184813682  
569300238294681640921384659243614376027652491515759520979955450481936784233  
658008793709425437826865612176200395940221712744289458469834800928056872827  
20277803296000000000000000\*s3^2+1555554423006504001687516164505831840027878  
902065719089046935785145786088696732740517593333880440683983037216728294123  
900771842823287221930273862400552715685183849618730449750965321918364624128  
726034191880773041750637596028668943563489976320000000000\*s1+2127363539397  
900342710932938970530059139583249127407384882281530520905555404735155597604  
909918017494151678324313517862188417591990721161743657809464431509014516741  
090712222780102416174039572628958114072400560340585590502426445339471424000  
0000000000\*s3+222585259214780313635495761207101756187752691806849106010831  
308285983636815495437576416069278459237814018195043914368914158507310140269  
700956789323593296776518881243750445531621826877468955284326173046464132702  
3020108581242484475558064000000000000\*s5, -  
406583458198262405582262355455873447846404411648779766818896263688403351627  
495446863666598321363397916020255107723859547673835727874056117567818629736  
664853204422528336383049510257979527823041905698656499285758513952923427344  
6958082506682248549659536225137\*s1^10-  
716602506930806311824179541814700397792500378967079338831779966278454201884  
753154441036491768582752488811448370619242609700523281520263154759725799095  
290993762337430855218411679784323920779115763797250134373788115486258895130  
7043370479692323478959076766500\*s1^9+48446502362780081777608351804575343207  
536798094508063504118065931272472906430494786188051410757898265160702311152  
793223562020141009173778299088530233469486231032831605368184616318142340733  
89094570129882545452600960693630350818633103206179033691648494036800\*s1^8+1  
451727909225439043994113626496340735291035837284794863849089774800226072258  
484035489212215383779739951043961664893850388066399467128089759217686173790  
791429426721916145709327250914804202402646534078667965212998949193944030479  
7442315162656859825383292320000\*s1^7+35583441539064212995467541149386822113  
108036314259153286485936444335609757933176219001670054475253683628411330863  
296684616968900664100768206369573503485140853179869090644689363701623849709  
54441755980953214731103124473371340799633940518186230524817543200000\*s1^6-  
740854768967767519311920848948815639451074801338233552357566702472428976707  
893256642553916662249153562260803802760158765144087040509731533039011780148  
980638983511749912876615390656128828909841172291650680497825289851266509313  
6583141711364517129269936000000\*s1^5-  
547458686501210050646191850279519545800301788125647689547193996724283049664  
664327404173106465224995353469641181036301766001234044760936684886339596606  
811875999976203063296151261965013288277314504480061898647270032104088952293  
1722471038227549790941552000000\*s1^4+53034940237033562200822169329989614077  
238419128580502148534379610160952617536745314745318051521444515779325744597  
372453758911448265461856202160373229331513785620321757827266989319232945815  
744264885227392531949343243892177494057646363806822964992000000000\*s1^3\*s3  
-  
122939193697301233991748133558359723492784573749411298919787944181736926558  
636322423360133219896411901765779312264247648842691896150322683361838862132  
608570377971950811414146567762055935893235719973529286537130438099497258173  
3936859636072659814604800000000\*s1^3+41304333863288254521494754356609613940  
455521809341901939101879944837177830502870297642837548903580060682209937848

871201402474522684660716278414867375064076775275868371553773134410402771356  
871933893230674675670695724189386592540162045475355536000000000000\*s1^2\*s3  
-  
533204146090982684219950535621183831942658319699507728246033700127435082236  
349903738681875538672203745022359859203977359741756354905735103748705644801  
706823360753333717172134587931493540115281032928917209469317031831597129950  
38599453257849017139200000000\*s1^2+5941264676885032418487921614044881747935  
058261458089908293941325394021110024105004839942418628162414349300406464242  
723008288247550368272460571685373273647255566584399542290863235213059437667  
384827794601912990869499129000784280362368111630643200000000000\*s1\*s3+4449  
622662148502038554723137280188104087434301178695003255610826011576674874211  
160508535376646232597752889183994443185139478170136940757123118243957700809  
526376982329283684297382373711491188994805121956554249274433558836033890676  
5603836032000000000000000\*s3^2+40063376877605458398516356523816306363740725  
651854017408445114860430276010534174379351561354465331308843505035382073888  
009467966203034915466903876571363390598385182448144351041206520892398951574  
425076035699071080713264100788399777323419200000000000000\*c5+1274958870863  
377602963893553180396439902118888519589044077283294043602439959447736678809  
007543569578473480811598880473894337492439913985103497195452726365096641533  
787470616228565069196600429529998106925757345692443524804520077617670159088  
025600000000000\*s1-  
400633768776054583985163565238163063637407256518540174084451148604302760105  
341743793515613544653313088435050353820738880094679662030349154669038765713  
633905983851824481443510412065208923989515744250760356990710807132641007883  
9977732341920000000000000000,  
999516056001039152079906156806521001473633173463700341451869364053819274467  
417604163013953236435068570047155368354032701671223063334623139623865889653  
923609563115190773027357589848601121249092873570359256225188562550490456085  
886064663404293056444608924865826097975288864849486468285947165263562152558  
2679987\*s1^10+446168863689234202230277754894707391475784630635071485383397  
628199913666848257099636592890275380745042556635142473959943821587327691941  
370716588587679117557414127721322322930942049792401433318191849270065201067  
140959497786595411363450581876101538685938939032290120224111747833591842611  
985436813072485696000\*s1^9-  
339821321014610550696683677013363322362779324267827457021523674051298312412  
696164337211431240075910697445938005091477365745864725441438210081821574642  
028869308706322614907277042490200447765871291610372234863361126714637947866  
796395002717542208755958889664612302937528067570541591950291719431324581651  
66466800\*s1^8-  
199635581678659200500177263058911601513476718983752082818206243263141328211  
652294763918803845202672999498135620497850927534176536892092747922679604351  
109768950830099830686228257052504291552315277805630639031902594117226088630  
451880133943658034409574020733593644244723505473167352504852628668146249364  
82320000\*s1^7+3136427776845530047998138161974462839472211134722915708107952  
375660715737477154007933010013418748495744631082890050555437852643331026092  
194005978143682973161824213998340309183819137395961205920335870558916661641  
924627418653322622588570382765802712089650875232745864137056901053352586307  
0107234101433456800000\*s1^6+25549535100794903002821145141427381229179079820  
222434975870174085497934974265744002095810214927524849316746247608578904810  
271392111941376987279639116694215428775218646438053957127837773002467250639  
926059666100097699177992333657880944822447908731701093696523651820509448972  
825545508385155588148219331136000000\*s1^5-  
122843845393400286528805985389638824338995655920624634486678417719013583085  
932844607782955268747184449605125221265674527221001043097123405634757364237  
060607639441440632981735525219556429811831553211690368287031735515183032660  
009746186052342831135341672202429811697906407195312635867191307038897322284  
8000000\*s1^4-  
234722902444091208855033600009056519766623053175772748143392836661091569165  
260930141756093216309455077315838967149284499701514586094705805155773734934  
152958176454807694026540533723756396172434629464271891324839752963365494060

528302144701098760399554007616900041567023961357871535617515037507563520000  
000000\*s1^3\*s3-  
504221721552818608948480689801857008168720230945887756145753617054194416878  
378093135122246600734304538854152093449055486997333603468063467986065569739  
828227686579459312559798333603543213461035993888945158259390344437090142659  
994106934188988828738672110903989594443460380864874417788982486005271128320  
000000\*s1^3-  
181442693638970741068670109656457940334486729247970614258259603476184053129  
231158421715550870259226327293379832521486619949590707708747393610466382126  
320480820971977300025590088770867283606392578724514425710133006686810698139  
04392704073461838926368770222259746235233512889871766426589582564416000000  
000000\*s1^2\*s3-  
123505224895709453997905288401862625012338961015685346078360099727650875024  
813693761008935200876249892893498439638610895235196673450975129814280421086  
400867618000236477237561952793414137530865278677924899105782126786981354783  
945548568061953149432859123874494083341495123420550627117125587087776174080  
000000\*s1^2-  
343718378809929643144740899223185206524512696202702862171264199515527951985  
493767605519847474380673309441445013671361227277681532641686164289564646999  
733471061958611922130610234889467039597629798226171569338965964663928076527  
09779967326545765418323933836620560607647265711390054852090610824192000000  
000000\*s1\*s3-  
333341275837994421381579667753156145809092693024565319795258945710821335985  
514164981457482475033686253704590388866832367060106271030701694485424888344  
135098135247920417246833712168671743503060660365015041065444722438223956945  
63355706025031252585838115242296442855944659676339555918935336112128000000  
000000\*s3^2-  
775602673412167419785738686388645396238810768435179730779732347148941724260  
159850258114215164750898027789304504452753356503518026896409588797608314945  
358475859775865101826663939845736153153389332377953881550743535328166216315  
49018239657299293360806937176164561074970412846529390179223881218457600000  
00000\*s1-  
309415539363923585955918525923059131565914426336887398838556603681928061465  
237109426933011255832671536088448714976228640768150451253538539943297791663  
201494221600875940455776570388621370116170715503413853552382581000402161870  
74458615414492042335478444801268149871680957713601746454705222942720000000  
00000\*s3+150576899001392615043147258524141610107443282412008079538143394556  
666110876708843088903661808394021893100752389286450630520245482306846920816  
505815925675145766756670892642130716855789038859290658276855148369591867584  
175040014617305062094642583286350743434250686283374169751808671377333772288  
000000000000\*s6, -  
290538680283773904796977003140003362761413787446210007248818010973278030240  
024482794938344080521785221801040122272910532589653358426680968803755327280  
519037658514208530929881572184260537419393206606255489331789163278364317087  
885124705088984746649922365551726519012924251686912990588609131089\*s1^10-  
529532827070422354077074788320443900043911305936345543584380341750580065613  
047188050709478142462268756185941201816103172826591396199550697255652082892  
801680740623565156765124238258668334998269360271549268726585586584327950673  
427061622781836176663950092344704963956852028685944009336199701600\*s1^9+283  
077015178500252580214435231915373062922746850609684541802402212643816603421  
713845913408658808000750946515311549747171221692033864758346759203162081928  
593734193133918835350437320859200722116878477688256275557999883772254066058  
346255471928844371511564011333583624121013456844917013023411600\*s1^8+973910  
611438293502293071374937938635949183237006485231690625068071379074611843934  
925833947225601671948837211768966077788291541407965230566852758888393002030  
143883411383683662196750579792207041790015603181285981513376448134514538775  
635528945679498602459138536012030109707802837332651246640000\*s1^7+296164922  
399405914427225056111775668773754937367573901289223142712291012808905750947  
383689512187665871895995096262403050632135850409166811173501605412577125636  
486350077542614294328958224315055929730595232862659237718680433841643171521

```
713267437128656575375092288747222448031554045914554400000*s1^6-
370065279364038118294469952002545271267870856019753050654615040752038877448
277822646613713156043317462666161557233135408691347448889051790347612284640
924892740478431322230740093469517808445469777808358712695371943442745240961
675522618488987567408135373266033508547477226304171001189312000000*s1^5-
295538270610922739156731781498035580628178234306314901846629265775717808514
325710730733503204143648740106416439266220260092404765008372013923982172294
280651392649007510458713030824233280324196365581738987719771716126767359330
465633062811900590385368915363610171811109092627065285863744000000*s1^4-
824065931638023901335662212798719134949465704271037819067055652745991589891
136201504501131737109492875407392982635659278757414369156007482150339931344
369328959092624977824751682737041193237394984260094530142092754148613425724
27402508983513783776231459578100259391719208891650621043200000000*s1^3-
118337359219600233062921607242477930859367889710056927326453429575933979981
107040267883314602108965634814798137614357565899831105209028018588476759590
698647550849727714690513197841454904766272130192143110461572163401858848948
48180578397453268355366493072731401889305125006465632870400000000*s1^2+3817
981655108875623472341233371934370932954257342627926346514033885795955532319
449713116651418162875965903615378547466498707046535394940981769310003485742
551694636920536982674813553234225335804262997415640039655603307650123357994
4408876208795894296797736753210761007598576640000000000000000*c6-
143788534792163517844679107533203980300712584362265787013814626035167078578
375029288157145772271570811787666957942256691854527126493003637324175559814
104584599490987259708833854251421094164040692106570609246403857290373337253
79311976788737018175008492127803245928448905797388288000000000000*s1-
381798165510887562347234123337193437093295425734262792634651403388579595553
231944971311665141816287596590361537854746649870704653539494098176931000348
574255169463692053698267481355323422533580426299741564003965560330765012335
799444088762087958942967977367532107610075985766400000000000000]
> math_operations := 188*additions+912*multiplications
> total_cost := 1100*flop
> #####
```

# APÊNDICE L

```
> #####
> # Solução da Cinemática Inversa para um manipular 6R com pulso
> # não esférico (Gröbner -> Solve)
> #####
> with(linalg);
> with(Units[Natural]);
> with(codegen, cost);
> #####
> #Entradas
> #####
> c1 := cos(theta1);
> c2 := cos(theta2);
> c3 := cos(theta3);
> c4 := cos(theta4);
> c5 := cos(theta5);
> c6 := cos(theta6);
> s1 := sin(theta1);
> s2 := sin(theta2);
> s3 := sin(theta3);
> s4 := sin(theta4);
> s5 := sin(theta5);
> s6 := sin(theta6);
> Eq :=
1175262138064162505664198980608584518451937095726344321*s1^11+2910120098660
049753263028747695997083011770034930582000*s1^10+23620535082785324478028678
2923617524667297708528815600*s1^9-
4734749428568907931203608683602748172679010491784560000*s1^8-
3795196768655369223929571145636498266957223725005600000*s1^7+76209076484127
9462235536831923509016134687212288000000*s1^6+22383029987510793324995255080
95909448387768222016000000*s1^5+1126569760199489107367796576802269937236141
47840000000*s1^4+24479503443709599747939978792596886789046927360000000*s1
^3+2251636012078674409085238869177208852971520000000000*s1^2+7245140171869
17029611823349836316672000000000000000*s1, -
283052426792554008658514169501696018630425161770570947535393698853216259827
674663198215607942306033565665918109653664765859648675790576135722397283024
405830749239177219873979934251437028982370662008653153907091410148499576064
4785025069263199251*s1^10-
635479207058847345758669374875124482703358127674277234515918851427921084345
429598178184533827302630982812222472200151174320068493886540653779568418693
286567428669598051746688082991196085579479541247367805453294883373895270374
9607748938467892000*s1^9+93182198003350327362631982166935012957631186381000
199074086507125147704377236852958900924500982170466281337044161486262626708
340049955854200964129226220926508043036202216253708891212940959795831238674
1268733671553929319577581979741978478730150*s1^8+11249885525822857744335602
203026983938352224551742598592303054125857937875502993373175533516742207884
713882685494203582195337183856193301668385300622995367304747882402539432734
275299539981149891953883820647940253709628596534010973971231679360000*s1^7+
651189623397073113727918835706579360211982590007860953985179941286277672038
052762545641886201099179693917206491192464399742771486800025365475394333832
404347099777940851347396636643720591607703216320825174573117505480077797880
9034063342097350000*s1^6-
346064862810178634708480929912601878277188980644895027892219952459970051484
```

259343415360066489816450310304244684574417132032875415422327570613074297212  
013484756540628391301471441934258484838223977699182047668631451017981620420  
7486388659328000000\*s1^5-  
463354336579047181166230550950592588450377946160221632194240915097545657529  
596531129446810612426471000949652277889482460313834144769430876716981118153  
119735530636430564722936124794035183712649713796568431535597531020999145293  
2472822466608500000\*s1^4-  
15883558585749367032601484036841178783780881037544688567243980538256589769  
400212202299020071550805930112270847361855389242464378865883751808178296982  
294395257022712306076920300975360099246102806868393852027825527439153632417  
785947095040000000\*s1^3-  
180221227765503853142259295293707410256694886409625875888383046294881275793  
813865469901377363438143089135441704922822694343204217821422113069334122946  
961777756921087534256663140622791462768659432645442991308011005425227449268  
00007674285000000\*s1^2+131201282638551012386923693699872286259953436365277  
026412222048585319953556366958877995772053618963906823940678846987235990016  
242205007202440399767363387752006342221472679742174016656567534724316215312  
6595418807603370345424949398437500000000\*c1-  
653391069948016616533671522145717128721438108120886386243125664643333353735  
616636765157568497574607674668105421920268860697388094831930889932986885996  
247285028628182857821442758079833703104323721795336914799565713673095465266  
053120000000000\*s1-  
131201282638551012386923693699872286259953436365277026412222048585319953556  
366958877995772053618963906823940678846987235990016242205007202440399767363  
387752006342221472679742174016656567534724316215312659541880760337034542494  
9398437500000000,  
876725743464079477511344180357144697327104025283766499818432699107580918755  
00182961850731222655603753507151640218312767989515873017\*s1^10+149237571151  
27983322939154802665279413754785443391117053438502563519700868088446658202  
634473273824093689712043408074945811101112000\*s1^9-  
743383246631069181362993652300839528590812037717224506528097433585269758029  
95632416317268658335610764981779941494168185641548471800\*s1^8-  
239966155772204989830731220471781300526894182608524098848559402943128291412  
310784047529402418332144022459709781695828914153240576000\*s1^7-  
104175988377018752376212163303255753175298651383390178752228138574427095124  
568259019089262407478854125891077929050118100350634000000\*s1^6+353773212924  
734114967140956393315186250790799954208925749232648249255237999926276787083  
42426473545024359267119168664495122326400000\*s1^5+2216198180955176999683025  
766045282902661584368744049832856165444756223105917040396688805286943666491  
3300183771850836833060531200000\*s1^4\*s3+38844017338442678954007192990361014  
833745225422830829080106948126763168998935199804318169697645759830000681440  
848135003610624000000\*s1^4+232602570436806004731533524572274954140648991699  
285865919357536845899262030312280116319675867750571005570152298960619529728  
00000000\*s1^3\*s3+1058015380294093309533322693804877836190826111721956546498  
8812182990512575047338629382210672478056286473434928227458308505600000000\*s  
1^3+69762117748909436544873100299715687705507468562803218242011561541899245  
88898402231608664970485377876292684828235463893227520000000\*s1^2\*s3+1070800  
765265894095066700827283496847307590202520530192689523160081059881307434858  
248398031842218605149387504312961033830400000000\*s1^2+453692162149355274892  
114301001516194934600432425265819681186303169929224826805296308691647964070  
050120232128350400829440000000000\*s1\*s3+36187276459371972874397712979813379  
593048311748142591975945366745538906210685681091952873882515716035823793453  
190348800000000000\*s1,  
626081198172040227469794483219851288076758513747635496229318228041211344381  
663430954191716729258046886677736223538810601113096202264671344469348114034  
543425197103189544441845284216782163820554707066291027156413583733746603\*s1  
^10+13768703652547816590591011879058547809844073302538842635198569605234194  
546613846469260745405647386216278027021939984710706397726233397190822036511  
443027081070915382573329604268944468836145789890992642728162066744455385453  
15\*s1^9-

333499939973840853389828137379180221430564481076075558222218627677506437758  
009704023699776566081753441029984039017106756784534028224491509875783106513  
067919719964349609572172444699228976166637753153498032127688168738580500\*s1  
^8-  
247783264931835917973995753435462079343126305113867375764016345922029166766  
488333893574426869832409585372292015072626834678333842914770714596865768413  
3497374690122107984957064859130023191371517017316803878494621226624420000\*s  
1^7-  
117731857853565546757437977508844893546770536476116123187313082021489978113  
110698397906312974330859850607041946324551034099415946549968396001065691392  
0244545173481871058804405356662080106670401311088249550812000099302400000\*s  
1^6+85223909663778781139136912862622020423503951524887821065835654894811006  
471917497846145976401805170757597924866962752840117124484825298128955142247  
912921476871436229627677458014175178747212573074974156715040211734778200000  
0\*s1^5+91701866144980251253973002045778435205506901473231842384243854909260  
713007448820585980012234123387927095634580940678691588297118282059779308249  
829381229205342994941524772595230745862038632830244902537062500537993284800  
0000\*s1^4-  
338483872126896271619073921355358639784894605137602262889721295523352969911  
919928193859561565429374959122097700188837070435091583539164288308914651037  
71309047111908000365796497018065221677090310800967905468140000000000000\*s1^  
3\*s3+2950877555412530379558151441607379746778824487301023289765067040493038  
824011993875464693477264434887928042256723107882957659578445252376843788821  
851422554912622550318269588680248628455983414076834490870264527746662400000  
00\*s1^3-  
206049654056785492357407546643788855672036673376260998841701152270252066818  
935044455930744720852660206668514536085491332763500526033714246160677532789  
4683572531070548178809451968442703610490457829549606903280000000000000\*s1^  
2\*s3+1738201133811633623191628878153651965041249570127625957347156346733665  
453022520851435369942800626678832079855658215080605198618944703835940705595  
848096655908675750435719892489461631428515755192423593844931033600000000000  
0\*s1\*s3^2+37678484658660510780950512682434884842364623496041297731747082182  
059724806710225412595154639742019488559124876248506681866098409133243625931  
614426720968369498205823409263722476564367833338229913947836003098705920000  
000000\*s1^2-  
892201493408564695449081676085625790693289534135447714487598808070013994553  
854498446602373258418004010434087489951660822898027148261212764607596098133  
377622036583468997587767256761453178922820526019191681536000000000000\*s1\*s3  
+15745639723184635366870703421806136953056191024241998497634597479589392665  
576878972907268045297434151055468017632643985236713081075427665131352533252  
63005221889028134441557706834396424036627049828216288174080000000000000\*s1,  
466423985101030474686309172727568303566867338512131804388383032464879463122  
223707783032130634404524834196718315829528415984627546643144990139916631113  
941728827273497678515297651343510355251874415392243398917734682245249208913  
703843562\*s1^10+56631016012465763438878025205095821025306501266303658581011  
014507323164449817818205441938148896683520453218622439848710671498700579508  
016837496855902293044079436062685045799802156175332954331985442711837563300  
2346865216411742620613375\*s1^9-  
631833837926023573651779624595736342677542414087494664685537887621655534743  
008828661693868861708820634131891081424063573540264262764667128242592400851  
686471779784168497633179260920700330881757068120638657612879774028003701623  
495050550\*s1^8-  
960382439593470930566956825430058637054197783365690663749962269925008134692  
965655424574986812548125020450269529562102612442754299934551012273943872339  
821080270645873671425034989217443502945376608376827516829246957088464542887  
927320000\*s1^7-  
403638773282148063707662589708175627260363233235763685177147916012473320351  
660346024825286664618604837145251302341160817841999666523706518520716304677  
017631693103845435220258755865385831255637748818555125700924234602666747512  
85700000\*s1^6+3548203484618158017164212769668510843244758514425274136491578

931411287423536058144922013801823975960364584629104169649963502051930355828  
694528951774907169708824775675214329577062023996046251665393254071245006919  
98737846010935336000000\*s1^5+1477597190382769942635052555759295688768510323  
976401715526341409049705328451053385466382921949222308798316913770977917824  
491795408417413880378800714810261950216300526199161412878420206618532413709  
00943944602107376917286431680352000000\*s1^4-  
705106846733007465622142284194188294670382143898494850350552156796306798003  
651561129057972166934189426331257785062152720809138429096501995770539412197  
91398570148706327776869713927775298786134479224683345389427102144649600000  
000000\*s1^3\*s3+926836950017763527448506499073624572599580372209023793230647  
299912709380896139315801878075688476857183166353207565873671253287732876794  
074260538880610160305947504889185388591354419993604369388473121326830995197  
8096455634124800000000\*s1^3-  
545053270227821251440163924973178789800477338922234242019506189784285831845  
758996376081191396324091848684359596242796003556383772290502274940738265000  
86420431049868664529376036599148746611958163588674916842631505789300000000  
000000\*s1^2\*s3+165449083673586260473358406322698730818043936679230443603324  
271426090687153150786963192213269836996312924738877696368111476613910051521  
963598246893716145527868933348455113362883953166573721244970709456245141812  
827117670400000000000000\*s3^3-  
355213520900993224578903890563525297784090518026826930566577338474077171678  
036587445624863489852639459016198003598207508494122187880234157376600649853  
961034136097308332663134074656731949955607119880801590251013915483678638080  
000000\*s1^2-  
103252890844163920047007555771226680444429668505441484153691553117456896303  
393500278375691781024472247469811973247459284696770887643482584976047030945  
99229953266664324107734991443145943033028376677053901327592029248160000000  
000000\*s1\*s3+15357389310230349056350591412576460712738023264343030922704681  
427570209492899047683577474734167276506110661137055464282338897783339914612  
918995394343862401939264185609433080634668075904306559166196452495501560607  
8058496000000000000000\*s3^2-  
290921068132161435245066194365933784777067770341153039724583066778371366170  
405906777817898078084026599361269737006988160684682697783719272240642838018  
12585766882630872266907808503624000238955994547737188448862978296053760000  
000000\*s1, -  
380153282861503569480423826199226702073417459123045023025731245116908848103  
032549606067097070692555091015804292047337007609540680326602968417633609070  
089567459451509102424076358964333147838273084671174173926652874210639511946  
29085740616419761660522406099535870839990319893100732014523209\*s1^10-  
279719566407211427148493149172857569627583335339707404107475706278554255079  
694399324381142510291243497067729402104797342693508513028635387463911904407  
530063027150580963556037605245923522399160246647811064651354065401308313954  
08992924768281628414076131122780300932128419642843739949488000\*s1^9+1098550  
158330710406293070455474623507356042354740115766181235622062985828841864889  
145524972907494164423892690897584275488259535270719945040046100422401521006  
074001517414615098204199249939100692640843142111064311907344204429845681854  
65812471881701335115595244176062545873814800817147707600\*s1^8+8675993582519  
875028519674552859931860065037826663011266062920313445794565677529066506703  
392666657131523373544242657440207139841209691509923439484535171489729136777  
559145610741427413512738715392148936455049857706929003813496349085632998483  
8586510988785252829684185879663205517737640240000\*s1^7-  
860531721036652518012811430250930770733706220341693748260717864995386716830  
021523524756626841740633664581370102239945839685058722549884682110312102102  
627790859852091932588919329515200468551932003276208054843028749143277767997  
74636142482366840036649105208219388967207134578309489477600000\*s1^6-  
869213320810946119749365617418581009585465879234417124399842548733154495526  
938885586392080511238811699436612481379744204472098661767341130031735278266  
527098684451684137543698324408545733233354435347397673060519367388407969770  
57150710159320762588919945404983074563255856405362767552000000\*s1^5-  
444275659142459089155905005823768459641225077070742748867339600212656040439



651929609198688164199386212760624653498946250071559032012973017742808405238  
957675351562681272766209717233561444730810780477850550605390110949568864977  
5248985433212892574017657288113467523851807736375900864000000\*s1^4+70499163  
170569185516307870929046235092310904571952045286781706633004608480242714011  
148426034134315075307276539532029122691549066578761625601530771662943004604  
268766546972088423279396107312428554454214833133733359904887863562883490383  
14250115860641962570185715941544160524902400000000000\*s1^3\*s3+1287673257965  
386453478278483434526297017768790475593908173973012376860219654556162332739  
174941399915848041723557337697806859493765722282995348616647452027473033937  
930787429860183632757800436755146728960537570819334254747559829622956207079  
4929043493241685486134093140071237812006400000000\*s1^3+54496420808396500242  
834307825537521905928481836133205273727127175338704773965362830086181638729  
603419953276728163948313905297593485843738093734573621245666042656082379437  
048245855639382423985949442049517484557807436775542032615463487194459874637  
0676281539842542000581953920000000000000\*s1^2\*s3+3383855764541497987225236  
742764759296020575008528430308052268878809814817581087688294260175773169386  
803995805891339465497620357506573467770718501984433659086745344773091442033  
589021599933218717699964995229187474247818713414687274039309763142789759113  
361727528396683814900233625600000000\*s1^2+103236019238542523090322649169934  
276112715730101749960784236932927350756686247722189278787921721332060961458  
597462772028697591273525693236733572763758279831062416476447863882915974285  
220341785270601967165615799145240533197342950006695078825466178893228500351  
1999996786391040000000000000\*s1\*s3+1001192501970958137482317518351043446272  
593548672032770396577521505409712117465318156052072714590101350202393548080  
942090812260145560284542234203979481150380278896804438522754813374929020849  
880800830679707965141909834794977480190576909123782663908862407418566503821  
602447360000000000000\*s3^2+432247167607537242302133844943785621704303744704  
601304002839728658568683051253240434345201387842461457183979945204939920480  
198659713020849318598588011289964502491874555217595170372338487145118532468  
723025565090948660461943866245257165298988737791198017234135565600260096000  
000000000\*c3+1933861786047083097098702095162630440155806291971473666197624  
288709579363475291193789710851684921310126705779562555434211672545465992840  
769670812252411613310185909541939006291614257979907013355379385440820996771  
2881418066515949392582093697005873180577989471439696591087288320000000000\*s  
s1-  
432247167607537242302133844943785621704303744704601304002839728658568683051  
253240434345201387842461457183979945204939920480198659713020849318598588011  
289964502491874555217595170372338487145118532468723025565090948660461943866  
2452571652989887377911980172341355656002600960000000000000,  
155591583513456782655249078265564499018303173862432252234212138160584943804  
905586365557582552234823213313664325448950548266802618108936761256435861588  
036000601314821236462093607259195958158745880611528099394151676995579690316  
924932022550334171168543896448847\*s1^10+27068956720022079402390696893388079  
561431127894461270659287074361253715887016545447523594747759061603640495953  
413760787024925842533372188365422826197353822776826281994332316782079995280  
1941991399633258621983066885081110207468038783203841987328889559003074000\*s  
1^9-  
191237543432370413669298844880036975063712445632958578780083662831434729272  
503070178261382983399375605153512242592311687428850016590586959676712435322  
086673181401863903039252111504819634903932550090723705559206220764005436198  
391520039233405803118039961705800\*s1^8-  
551171702195413598251704324642986713190997967388509425899531511099924353924  
100520688908505193768869463908394358737615163118410880626385777591565200842  
577175139257825165232738934378111013081440663353747434288259333334209858905  
425549262321351051647755887920000\*s1^7-  
125436648513084459198531217734683312281770810817064721913277680695473542781  
647679934998186877720151985682946155904287871165301189125263481463484173981  
708451537851925812843248246230359905520307518456324775706043958346072799804  
838580122894920333210222299200000\*s1^6+285262904611430237659659848545303562  
117998492707979627649830361566121166599079285051337102476116600177775392706

841900425153889234422323222889025748406226527874828352084123107228179186123  
194230466812388185847137404922611116018466122431773868385287196816000000\*s1  
^5+205407971288481922444186905025819404188305301210046589716297366088362956  
336753707887665427901083080787176825450398368746479028979573485350718292922  
334197491543373988968478999155836654015865772869946522629073946616089611949  
083554298226893396178414299712000000\*s1^4-  
205413323741625377874704567201900970335118472244983024787787949634867421287  
115103696900882987746404959604610506374855699889072739085775588522456568476  
958744867911645822236939216166917814302834927389695993519059816961342024523  
86964656960182889111040000000000\*s1^3\*s3+4485954845589300077774672123794057  
115585121287655398671478669834931701281646759599341011417316760443809203888  
032128645154318886601311208525817665628229273261255492377359729965516453082  
496965783461041953615506456795405015041875188002066077769501818880000000\*s  
1^3-  
159948411382686397812368940080392735173778161726414150057439161645916514275  
446427887265878239153201303386601665672312879416541328923641718530405422299  
661857971608932497647648666962743105738828131403686633789731468536020419588  
90862190374145760320000000000000\*s1^2\*s3+1688367784594310763239197295208234  
417954098282828213434402077012393061390722878184238551168974617365329422737  
183272223081294945828296468706103656623720754633394164856578892269822780292  
17003614239667738049098643280520156793996542092684786178789867520000000\*s1  
^2-  
231910183339138720926918533485704714190096336308228359250637451924458545905  
117760876958838986575474160362021835791012413386325116001832112739788276309  
301445154501504035304469351393261952547831945757084708511185244203133590159  
9213525756996592384000000000000\*s1\*s3-  
175018491377841080183152443399687398760772415846362003461387359156455349211  
718972313335724814751815511613641237114765282152808025386336446842650929002  
898508037494638285158249030373365985320100462334796957800471461053314217333  
033278041755059200000000000000\*s3^2-  
653330835624693318738066693566485607316095527837610735788446434357152303241  
573742318354406494576008618619327463610470561955812068826308081736837380387  
645002293793959266647519377981897334511226787962663010106192701392329337924  
36965352717359513600000000000\*s1+151607271756371167223408872873381148174602  
113443481063086097420699749174707770252442048513106944156059884073008310954  
954811642499082740498715688344085866300422540392743303460498258722781654823  
260242985407650066191945635685588102599210855168000000000000\*s2+670828636  
090137908068180853422040478648681917891509128699546109290925551804293152398  
444748260814849822495898266862632543414347341074073003166762584450735842577  
612357271254249992295233547145235664791970830310027397989538431805763713322  
368000000000000000\*s3, -  
407735534661312917927910981334532463965509286806969149268139095866903691058  
110416790942514727889588115453392511114333051838946744633341595816333054704  
996376506067421353592128475133566459562886132338819426438128918340029319170  
476177131062520860957241551608548641221244305165417754123458839538438071341  
54389973\*s1^10-  
450895717596113050884114494323014230068493000027826898900762611418912741458  
595147924346864769934273464658915266067630727207968206287261525472269198592  
553464864900636080319099841688492368560281380811056013754665983847039550602  
984322636554096420853201916651333699985053872929293693094408691138805708320  
26416000\*s1^9+9210448896127059246780058214542623986659572492637316660224083  
103900363594442620146003007850526711100186601121997231022011586149349186806  
741210274501278336297701908573576921427451583842829609170458956060686162125  
362509444073914084738023824213403536709521578698184031207573835644835937106  
6204973130948612717200\*s1^8+10981482357832984809671142365684635207255407020  
847859389455578332465379392591154685909607163886075730542069678457070626209  
205820840502092107206317839140770572756619281241880600171703368299665278593  
630385098604850480668421355627857293054273076235395623848519771587780215218  
6745218860506581204531811769767280000\*s1^7-  
467430294237015707485881709535668880170755152427021641379946148225455179106

565741571419875825691161665939026864651933265690625646719958010709509041676  
 156515114987543441449238833138266635577878196305338033053091832564031464841  
 939902756827474512500399077482826402144732707525699515060038956153417832272  
 47200000\*s1^6-  
 824989326327097346880206594980514104491572005999883574078524423440184777239  
 293329783665619485522816003555623174567376263159581073611702942159300373353  
 246953560665497382248379473923800688509794272393391582107450643962245596390  
 553070876612112692146265523285052049472619627586423753849689844597878809261  
 44000000\*s1^5-  
 195428900036703633258409775904443400006442860072040802362583439850472845513  
 508123049863043096849991330068751474217796957427115324524838856188429408229  
 597151924018004066662360642145662855897615031492349814328698402145312059216  
 674576082967356487885472881783266281703509825336016453199078400484958407710  
 08000000\*s1^4+5609620027009032411816345766106525904264874015744078732788291  
 616914445384071210986273099079819637096823673002606074650792210158756431202  
 839831698897889202830636102485718048643660321615416733569262270840907405833  
 730957678332601550606945322890666648257853348230550329459203570478001481692  
 5354240000000000000000\*s1^3\*s3+512880558630488048169266784218102623583800600  
 225395089667702794651743811150724721984623710282089599658375332167256783566  
 549759740732123106019643984185540412006069847242577318444774605971603628798  
 687635001680031280407108105155310028722892014465227545222765086329358331917  
 2483087175947252240976965260800000000\*s1^3+43362814509933511745205033751765  
 185213137117605554728983992914973629681656753177167611183971734756734055152  
 470013812026321296125489409910070646650248273081773096584935934050722048781  
 282218545398272424535076996180906163422751057035121673704415847646410432395  
 322164943194505102033883951940920000000000000000000\*s1^2\*s3+2169808333416525  
 102157203511673338450274088096716064322245880418359090234206996064091368269  
 579399079719002388593142515073421963419616289258613410451507578377110857947  
 422072639996664712648577842354618870372400018164259421090461835803573594265  
 271484853102914905381196071844045917035124766807324909363200000000\*s1^2+821  
 449241359189778809666940097680906754205544805104350692773938033149941932752  
 069288479445107803228273557576306875837217671998834967532154040518286257792  
 644105401431938347362346843354633631343166580159279183678824634617062540704  
 05489221312722964023238534952407254298946069151862172312135040000000000000  
 000\*s1\*s3+76192866910134857572594664387449997144978497270295698249402130110  
 833833813844758339333189104664818942585072037423356113403693463319304776429  
 004177218268148708286892893729729704841996710143581467936432928950794629020  
 623749968642564021052599524371411884139637898096368929198504840190514972160  
 0000000000000000\*s3^2+69092476337036924371258658183277335745133964995054863  
 710947418061070952355224221103105996127787330920276316557309396364609182025  
 554398759335204775593397702650150536911596365736560742909457011515630886199  
 749278552784182115046402920046516084843168691580094725120980750394630896826  
 8000443596800000000000000000\*c2+137296200227835604835225599149603965671535445  
 747107633210871048982043257692299099449030691849613147831560453757562344766  
 554065044812043864066711023808452648426346099185791984925107439352340250139  
 251810491638343274675937218077726591913980412867448431706663620242520162013  
 7272763752514505424871424000000000000\*s1-  
 690924763370369243712586581832773357451339649950548637109474180610709523552  
 242211031059961277873309202763165573093963646091820255543987593352047755933  
 977026501505369115963657365607429094570115156308861997492785527841821150464  
 02920046516084843168691580094725120980750394630896826800044359680000000000  
 0000, -  
 427066771541665855288021460301220298353283718755254469584030604911180878677  
 970329920934574015985408168062131299692047891875639389639278512846692680131  
 257992190089490093902003350259372070654033461247011571862258797144283549349  
 119140212140398473869667555834027482761497303363705752983977609027763106172  
 75256392659\*s1^10-  
 786940313480442508109098028411100094826159370060526825340898754482918983910  
 089571701947898498643911204236796758674556021775010795304121196776682983651  
 492305393163713048392063099605840283706599904141574748700983887329293100448

952728022998272139294554975467973536222109025250863693976761821093280509450  
53219488000\*s1^9+4868876870017733317092930785758842135311261205043302656816  
120314660001079705603266574341924608886745114862997944261463088538749538008  
774918042093262475045448452660506263275283978333061144617955344551252491611  
657534602166872205515943578184014612456112667974123369533121128106599410424  
7468102870599045869601487600\*s1^8+16020166004709279039295960697003252812441  
283023208290901178428371133157385559677278119323865268164962537095606326397  
317876319811000651760578670940787341643557615468324684277285450730687012095  
844419082277459361744738494107180677214263630166983210343721463542232979034  
0184148464363692765270537110392383513012240000\*s1^7+42007344506149288714779  
357973694889926354009942547230463375609853104498116465173376057541244023657  
075722978385365834315458325442001510025545197240595144216166352179181374741  
828232811413878538453581239587190549601010494466578015711371250969189699930  
076343910368783424115762404778214150986179983783527899202400000\*s1^6-  
816813001027970080036656549440952868809898717171088344070781346735243745535  
562975818002677886666401313216761988816540184449859215006412211856102341390  
900521770139501308049631916059623407427759024018749701797549961999238019072  
930038308407500812950204897317778894184218640035323215928792911002569949650  
17152000000\*s1^5-  
596762419402515396134163009994378722511181166496168194397176380025828283553  
306258933930197475325728491446823360671020271665740375023003396274913160875  
679958204961046288451245895967598576412015928233868255854402081308656978639  
551721876012073485048787428258102514687120168553880259612934175249953745107  
20064000000\*s1^4+4758937661558096350012374118327479815605614577522794201355  
944090455616262010524705726902379430223748121449842667260506692750351377770  
310759477168670119026882321181978905964504959653228294626553444528156929666  
414234888014172934830045332290244052431180459038299391704534652085772350579  
7244456630155264000000000000\*s1^3\*s3-  
122916748067262870493440673770961677759581565456924653240018796282837931497  
254390600009855380500657995838862992231485080425076911912283365142137587777  
533372216304819595088974606243994031185163206990056173349296798236650280313  
295197276994217005373130426780924263277766569507441938135342980193682056164  
73600000000\*s1^3+3678697132584737919540524750618506930883404592740332628107  
429928038369091846889464971434857185352600547177562263373290375666230383578  
140983965176311792727605507268803128690275530846976077407567627768570078427  
077579970087276953817741388948903598393659341688252075090305380581992908471  
2839908277312000000000000000\*s1^2\*s3-  
151233265251426141509390105922841383186177158491589281840982562489478315347  
655264366594915430599624631485855198898515358453296403813140425904386550873  
229568001185694024445980862145865516819342354938738333614328499229402693260  
657551556242827759622469558904135707564651957514757112621838048440345211494  
400000000\*s1^2+696878881803143632619525510990579848524958098396873483389724  
192695735304330749127930340489763670063514181137359186643195938335092372322  
661376727918723673249422808550926662940364673872261641058628546070522410454  
427025625162676175522839664594264899160816112419152309485214537582790152154  
23052166144000000000000000\*s1\*s3+67583961139672397020785083148477960374045755  
757795527252094546720827855521532915255994209848568358993622714576400825262  
834416277344294066921314192723108450348498131818536074066649103918406818241  
327839147152020103697315138032995806342486043799067075081424291081072615115  
51219253264035335612211609600000000000000\*s3^2+60898843662282492014765373663  
689939693383335217455674013441979873302748510681045878929715715186050206836  
727650986526503657009032585902118656408106827502365704620914907295816488005  
304518057564777046180585521208410168093281398543287244824948740121056275958  
1916954648137990703794799205887959360684032000000000000\*s1+62733088591600929  
776443790078457685830198974583598654545926940291324291147888081079498133264  
726161613076669763307835646020073392893786181923723913374930671667251368984  
286048243192908116384094643866100017863720765439493814040341493205113866883  
1059736319750570207473815427312310610585039247328870400000000000000\*s3+30529  
022441218368065194437032152103784722153867125960222608717712832234710212545  
917240693621507563575700532658285036296297451737282505764674335926836206571

483201570608201220721278202485578822974835785107745910622285369513292672647  
425399943832002476401434889936883953980743450085700305527540121600000000000  
00\*s4,  
283052426792554008658514169501696018630425161770570947535393698853216259827  
674663198215607942306033565665918109653664765859648675790576135722397283024  
405830749239177219873979934251437028982370662008653153907091410148499576064  
4785025069263199251\*s1^10+6354792070588473457586693748751244827033581276742  
772345159188514279210843454295981781845338273026309828122224722001511743200  
684938865406537795684186932865674286695980517466880829911960855794795412473  
678054532948833738952703749607748938467892000\*s1^9-  
931821980033503273626319821669350129576311863810001990740865071251477043772  
368529589009245009821704662813370441614862626267083400499558542009641292262  
209265080430362022162537088912129409597958312386741268733671553929319577581  
979741978478730150\*s1^8-  
112498855258228577443356022030269839383522245517425985923030541258579378755  
029933731755335167422078847138826854942035821953371838561933016683853006229  
953673047478824025394327342752995399811498919538838206479402537096285965340  
10973971231679360000\*s1^7-  
651189623397073113727918835706579360211982590007860953985179941286277672038  
052762545641886201099179693917206491192464399742771486800025365475394333832  
404347099777940851347396636643720591607703216320825174573117505480077797880  
9034063342097350000\*s1^6+34606486281017863470848092991260187827718898064489  
502789221995245997005148425934341536006648981645031030424468457441713203287  
541542232757061307429721201348475654062839130147144193425848483822397769918  
20476686314510179816204207486388659328000000\*s1^5+4633543365790471811662305  
509505925884503779461602216321942409150975456575295965311294468106124264710  
009496522778894824603138341447694308767169811181531197355306364305647229361  
247940351837126497137965684315355975310209991452932472822466608500000\*s1^4+  
158835585885749367032601484036841178783780881037544688567243980538256589769  
400212202299020071550805930112270847361855389242464378865883751808178296982  
2943952207022712306076920300975360099246102806868393852027825527439153632417  
785947095040000000\*s1^3+18022122776550385314225929529370741025669488640962  
587588838304629488127579381386546990137736343814308913544170492282269434320  
421782142211306933412294696177775692108753425666314062279146276865943264544  
2991308011005425227449268000076742850000000\*s1^2+24990720502581145216556894  
038070911668562559307671814554708961635299038772641325500570623248308374077  
490274415018473759236193569943810895702933289021597667048827089804319950890  
2888869652447093935648214589603582400641970557133218750000000000\*c4+8860078  
658968273463948571688531452501192426776750337730670313168557507747392268045  
226148308580535352338790364250276190195941341410180464704669455410814000972  
513595328019055263755098828857389142723037194229546948822224200614497874500  
00000000\*s1-  
249907205025811452165568940380709116685625593076718145547089616352990387726  
413255005706232483083740774902744150184737592361935699438108957029332890215  
976670488270898043199508902888869652447093935648214589603582400641970557133  
21875000000000, -  
930296262535146430028646528397046380074714876771894974677142216597744967513  
135570354703678054045464592230138922014818719367298449812314631223027889681  
621909185472711246558023435643636611722193873173590210650153504324419355327  
6782803953183319479405156839\*s1^10-  
151051847475036794751965660417562546832998107438528830420090921150734538256  
020614069315136574185208126770640277678121052007752080875815525262358057479  
415973689527815512042276955332154780681519309392006778099430571680359905567  
83331838541924262634397254000\*s1^9+1254865826426699406251394196675548977218  
468208192990467316648729058829085608011529003476387566217693388917533202087  
910100114952138684007198753964152293642255570959619731360825113497891189902  
0804302343774263558798664157599027280606818672616500471992265850\*s1^8+30162  
053432312879056316273163619288967494919421555365200066795116395195920287688  
615305814135595549786642351111751717843191780067309557102088862171223829774  
434091913406683310751103496239121802358255537916502896504675880367144497989

845933470262966034040000\*s1^7+382310595898468736926125250684809674758060643  
230339690844370323488215827028302018588545068604663545824832076495417367328  
464482289241324720299959670262589934971173804635303744529950889738386285081  
7370342532599357048510258727538124567405038610396335400000\*s1^6-  
152075138312597115128907580269128420138229264388252354725139494615749831113  
725641992401778892059481330159871973338526837432809461791242623625543384895  
068341724276540317129219639489060686958339258314258632406500423738499680247  
67849504296906077782892000000\*s1^5-  
838678125668055914520944481481168684885286765713255920514301625498112555281  
249711429330361190851592039346153867899307695139493262313225100615709981690  
144836430480236650511002242471894168293043116576201554676871341157335182498  
3549880715497343862844000000\*s1^4+43928132724473449810622268749483010044612  
681852163436913718878218846297222221977097162450715318793577577451013888512  
151076644261008377116431619440727240357763402151422154741147211674645960914  
0599961646567151342975751041237349939929652548234240000000000\*s1^3\*s3-  
997994566205262381872529921736115553316878085892661800147026121692223845366  
798152935607564528996959460072884486192893168614987447632330520678162358151  
730231375932597274164713287274052233413088311366741453138350709419471694661  
18494022185688284160000000\*s1^3+345681773856510250889761277112217759443178  
292908970988935969987545396006561188652546211449966336718742838271851235767  
749018730822040067562188090484094511617394040860074987906880042703245570765  
266491468409439000661564235581861466543115636408000000000000\*s1^2\*s3+153574  
615948195534743972389431111110189126261644805190074471407708585109488307874  
621084995255311542098622173307090439103430539022051337538283570441181616811  
716282585059258157000095871472515603337472843761283140410726840786471289828  
55043140853760000000\*s1^2+675282101104467873111145594932912453109043098553  
257070813636637391720745547139839779038356445910051089298266076626711241156  
979118810826644773289103699121240522712063667622049667194175168349488986033  
17803105942851732241031422957540488997504000000000000\*s1\*s3-  
235180901804889114088516022054978229602929931351939482201670762474184813682  
569300238294681640921384659243614376027652491515759520979955450481936784233  
658008793709425437826865612176200395940221712744289458469834800928056872827  
20277803296000000000000000\*s3^2+1555554423006504001687516164505831840027878  
902065719089046935785145786088696732740517593333880440683983037216728294123  
900771842823287221930273862400552715685183849618730449750965321918364624128  
726034191880773041750637596028668943563489976320000000000\*s1+2127363539397  
900342710932938970530059139583249127407384882281530520905555404735155597604  
909918017494151678324313517862188417591990721161743657809464431509014516741  
090712222780102416174039572628958114072400560340585590502426445339471424000  
0000000000\*s3+222585259214780313635495761207101756187752691806849106010831  
308285983636815495437576416069278459237814018195043914368914158507310140269  
700956789323593296776518881243750445531621826877468955284326173046464132702  
30201085812424844755580640000000000000\*s5, -  
406583458198262405582262355455873447846404411648779766818896263688403351627  
495446863666598321363397916020255107723859547673835727874056117567818629736  
664853204422528336383049510257979527823041905698656499285758513952923427344  
6958082506682248549659536225137\*s1^10-  
716602506930806311824179541814700397792500378967079338831779966278454201884  
753154441036491768582752488811448370619242609700523281520263154759725799095  
290993762337430855218411679784323920779115763797250134373788115486258895130  
7043370479692323478959076766500\*s1^9+48446502362780081777608351804575343207  
536798094508063504118065931272472906430494786188051410757898265160702311152  
793223562020141009173778299088530233469486231032831605368184616318142340733  
89094570129882545452600960693630350818633103206179033691648494036800\*s1^8+1  
451727909225439043994113626496340735291035837284794863849089774800226072258  
484035489212215383779739951043961664893850388066399467128089759217686173790  
791429426721916145709327250914804202402646534078667965212998949193944030479  
7442315162656859825383292320000\*s1^7+35583441539064212995467541149386822113  
108036314259153286485936444335609757933176219001670054475253683628411330863  
296684616968900664100768206369573503485140853179869090644689363701623849709

54441755980953214731103124473371340799633940518186230524817543200000\*s1^6-  
740854768967767519311920848948815639451074801338233552357566702472428976707  
893256642553916662249153562260803802760158765144087040509731533039011780148  
980638983511749912876615390656128828909841172291650680497825289851266509313  
6583141711364517129269936000000\*s1^5-  
547458686501210050646191850279519545800301788125647689547193996724283049664  
664327404173106465224995353469641181036301766001234044760936684886339596606  
811875999976203063296151261965013288277314504480061898647270032104088952293  
1722471038227549790941552000000\*s1^4+53034940237033562200822169329989614077  
238419128580502148534379610160952617536745314745318051521444515779325744597  
372453758911448265461856202160373229331513785620321757827266989319232945815  
7442648852273925319493432438921774940576463638068229649920000000000\*s1^3\*s3  
-  
122939193697301233991748133558359723492784573749411298919787944181736926558  
636322423360133219896411901765779312264247648842691896150322683361838862132  
608570377971950811414146567762055935893235719973529286537130438099497258173  
3936859636072659814604800000000\*s1^3+41304333863288254521494754356609613940  
455521809341901939101879944837177830502870297642837548903580060682209937848  
871201402474522684660716278414867375064076775275868371553773134410402771356  
871933893230674675670695724189386592540162045475355536000000000000\*s1^2\*s3  
-  
533204146090982684219950535621183831942658319699507728246033700127435082236  
349903738681875538672203745022359859203977359741756354905735103748705644801  
706823360753333717172134587931493540115281032928917209469317031831597129950  
38599453257849017139200000000\*s1^2+5941264676885032418487921614044881747935  
058261458089908293941325394021110024105004839942418628162414349300406464242  
723008288247550368272460571685373273647255566584399542290863235213059437667  
384827794601912990869499129000784280362368111630643200000000000\*s1\*s3+4449  
622662148502038554723137280188104087434301178695003255610826011576674874211  
160508535376646232597752889183994443185139478170136940757123118243957700809  
526376982329283684297382373711491188994805121956554249274433558836033890676  
56038360320000000000000000\*s3^2+40063376877605458398516356523816306363740725  
651854017408445114860430276010534174379351561354465331308843505035382073888  
009467966203034915466903876571363390598385182448144351041206520892398951574  
425076035699071080713264100788399777323419200000000000000\*c5+1274958870863  
377602963893553180396439902118888519589044077283294043602439959447736678809  
007543569578473480811598880473894337492439913985103497195452726365096641533  
787470616228565069196600429529998106925757345692443524804520077617670159088  
0256000000000000\*s1-  
400633768776054583985163565238163063637407256518540174084451148604302760105  
341743793515613544653313088435050353820738880094679662030349154669038765713  
633905983851824481443510412065208923989515744250760356990710807132641007883  
9977732341920000000000000000,  
999516056001039152079906156806521001473633173463700341451869364053819274467  
417604163013953236435068570047155368354032701671223063334623139623865889653  
923609563115190773027357589848601121249092873570359256225188562550490456085  
886064663404293056444608924865826097975288864849486468285947165263562152558  
2679987\*s1^10+446168863689234202230277754894707391475784630635071485383397  
628199913666848257099636592890275380745042556635142473959943821587327691941  
370716588587679117557414127721322322930942049792401433318191849270065201067  
140959497786595411363450581876101538685938939032290120224111747833591842611  
985436813072485696000\*s1^9-  
339821321014610550696683677013363322362779324267827457021523674051298312412  
696164337211431240075910697445938005091477365745864725441438210081821574642  
028869308706322614907277042490200447765871291610372234863361126714637947866  
796395002717542208755958889664612302937528067570541591950291719431324581651  
66466800\*s1^8-  
199635581678659200500177263058911601513476718983752082818206243263141328211  
652294763918803845202672999498135620497850927534176536892092747922679604351  
109768950830099830686228257052504291552315277805630639031902594117226088630

451880133943658034409574020733593644244723505473167352504852628668146249364  
82320000\*s1^7+3136427776845530047998138161974462839472211134722915708107952  
375660715737477154007933010013418748495744631082890050555437852643331026092  
194005978143682973161824213998340309183819137395961205920335870558916661641  
924627418653322622588570382765802712089650875232745864137056901053352586307  
0107234101433456800000\*s1^6+25549535100794903002821145141427381229179079820  
222434975870174085497934974265744002095810214927524849316746247608578904810  
271392111941376987279639116694215428775218646438053957127837773002467250639  
926059666100097699177992333657880944822447908731701093696523651820509448972  
825545508385155588148219331136000000\*s1^5-  
122843845393400286528805985389638824338995655920624634486678417719013583085  
932844607782955268747184449605125221265674527221001043097123405634757364237  
060607639441440632981735525219556429811831553211690368287031735515183032660  
009746186052342831135341672202429811697906407195312635867191307038897322284  
8000000\*s1^4-  
234722902444091208855033600009056519766623053175772748143392836661091569165  
260930141756093216309455077315838967149284499701514586094705805155773734934  
152958176454807694026540533723756396172434629464271891324839752963365494060  
528302144701098760399554007616900041567023961357871535617515037507563520000  
0000000\*s1^3\*s3-  
504221721552818608948480689801857008168720230945887756145753617054194416878  
378093135122246600734304538854152093449055486997333603468063467986065569739  
828227686579459312559798333603543213461035993888945158259390344437090142659  
994106934188988828738672110903989594443460380864874417788982486005271128320  
0000000\*s1^3-  
181442693638970741068670109656457940334486729247970614258259603476184053129  
231158421715550870259226327293379832521486619949590707708747393610466382126  
320480820971977300025590088770867283606392578724514425710133006686810698139  
043927040734618389263687702222597462352335128898717664265895825644160000000  
0000000\*s1^2\*s3-  
123505224895709453997905288401862625012338961015685346078360099727650875024  
813693761008935200876249892893498439638610895235196673450975129814280421086  
400867618000236477237561952793414137530865278677924899105782126786981354783  
945548568061953149432859123874494083341495123420550627117125587087776174080  
0000000\*s1^2-  
343718378809929643144740899223185206524512696202702862171264199515527951985  
493767605519847474380673309441445013671361227277681532641686164289564646999  
733471061958611922130610234889467039597629798226171569338965964663928076527  
097799673265457654183239338366205606076472657113900548520906108241920000000  
0000000\*s1\*s3-  
333341275837994421381579667753156145809092693024565319795258945710821335985  
514164981457482475033686253704590388866832367060106271030701694485424888344  
135098135247920417246833712168671743503060660365015041065444722438223956945  
633557060250312525858381152422964428559446596763395559189353361121280000000  
0000000\*s3^2-  
775602673412167419785738686388645396238810768435179730779732347148941724260  
159850258114215164750898027789304504452753356503518026896409588797608314945  
358475859775865101826663939845736153153389332377953881550743535328166216315  
490182396572992933608069371761645610749704128465293901792238812184576000000  
00000\*s1-  
309415539363923585955918525923059131565914426336887398838556603681928061465  
237109426933011255832671536088448714976228640768150451253538539943297791663  
201494221600875940455776570388621370116170715503413853552382581000402161870  
744586154144920423354784448012681498716809577136017464547052229427200000000  
00000\*s3+150576899001392615043147258524141610107443282412008079538143394556  
666110876708843088903661808394021893100752389286450630520245482306846920816  
505815925675145766756670892642130716855789038859290658276855148369591867584  
175040014617305062094642583286350743434250686283374169751808671377333772288  
000000000000\*s6, -  
290538680283773904796977003140003362761413787446210007248818010973278030240



```

024482794938344080521785221801040122272910532589653358426680968803755327280
519037658514208530929881572184260537419393206606255489331789163278364317087
885124705088984746649922365551726519012924251686912990588609131089*s1^10-
529532827070422354077074788320443900043911305936345543584380341750580065613
047188050709478142462268756185941201816103172826591396199550697255652082892
801680740623565156765124238258668334998269360271549268726585586584327950673
427061622781836176663950092344704963956852028685944009336199701600*s1^9+283
077015178500252580214435231915373062922746850609684541802402212643816603421
713845913408658808000750946515311549747171221692033864758346759203162081928
593734193133918835350437320859200722116878477688256275557999883772254066058
346255471928844371511564011333583624121013456844917013023411600*s1^8+973910
611438293502293071374937938635949183237006485231690625068071379074611843934
925833947225601671948837211768966077788291541407965230566852758888393002030
143883411383683662196750579792207041790015603181285981513376448134514538775
635528945679498602459138536012030109707802837332651246640000*s1^7+296164922
399405914427225056111775668773754937367573901289223142712291012808905750947
383689512187665871895995096262403050632135850409166811173501605412577125636
486350077542614294328958224315055929730595232862659237718680433841643171521
713267437128656575375092288747222448031554045914554400000*s1^6-
370065279364038118294469952002545271267870856019753050654615040752038877448
277822646613713156043317462666161557233135408691347448889051790347612284640
924892740478431322230740093469517808445469777808358712695371943442745240961
675522618488987567408135373266033508547477226304171001189312000000*s1^5-
295538270610922739156731781498035580628178234306314901846629265775717808514
325710730733503204143648740106416439266220260092404765008372013923982172294
280651392649007510458713030824233280324196365581738987719771716126767359330
465633062811900590385368915363610171811109092627065285863744000000*s1^4-
824065931638023901335662212798719134949465704271037819067055652745991589891
136201504501131737109492875407392982635659278757414369156007482150339931344
369328959092624977824751682737041193237394984260094530142092754148613425724
27402508983513783776231459578100259391719208891650621043200000000*s1^3-
118337359219600233062921607242477930859367889710056927326453429575933979981
107040267883314602108965634814798137614357565899831105209028018588476759590
698647550849727714690513197841454904766272130192143110461572163401858848948
48180578397453268355366493072731401889305125006465632870400000000*s1^2+3817
981655108875623472341233371934370932954257342627926346514033885795955532319
449713116651418162875965903615378547466498707046535394940981769310003485742
551694636920536982674813553234225335804262997415640039655603307650123357994
44088762087958942967977367532107610075985766400000000000000*c6-
143788534792163517844679107533203980300712584362265787013814626035167078578
375029288157145772271570811787666957942256691854527126493003637324175559814
104584599490987259708833854251421094164040692106570609246403857290373337253
79311976788737018175008492127803245928448905797388288000000000000*s1-
381798165510887562347234123337193437093295425734262792634651403388579595553
231944971311665141816287596590361537854746649870704653539494098176931000348
574255169463692053698267481355323422533580426299741564003965560330765012335
799444088762087958942967977367532107610075985766400000000000000;
> sys := {Eq};
> S := evalf(solve(sys, {thetal, theta2, theta3, theta4, theta5, theta6}),
5);
> math_operations := cost(Eq);
> V := {additions = 1, divisions = 4, functions = 8, multiplications = 1};
> total_cost := eval(cost(Eq), V);
> #####
> # Saídas
> #####
>S := {thetal = 0., theta2 = 0., theta3 = 0., theta4 = 0., theta5 = 0.,
theta6 = 0.}, {thetal = 0., theta2 = 1.5209, theta3 = -3.0486, theta4 = 0.,
theta5 = 1.5277, theta6 = 0.}, {thetal = -0.83106e-1, theta2 = 0., theta3 =

```

---

```
0., theta4 = 2.6020, theta5 = 0., theta6 = -2.6851}, {theta1 = -2.6851,
theta2 = 3.1416, theta3 = -3.0486, theta4 = 0., theta5 = -0.92956e-1,
theta6 = -2.6851}, {theta1 = -2.6851, theta2 = 1.6207, theta3 = 0., theta4
= 0., theta5 = -1.6207, theta6 = -2.6851}, {theta1 = -2.6188-0.60089e-1*I,
theta2 = 3.1416, theta3 = 3.1416, theta4 = 1.8665-1.1349*I, theta5 = 0.,
theta6 = 1.8005+1.0748*I}, {theta1 = -0.89541e-1, theta2 = 1.5470, theta3 =
-3.0755, theta4 = 3.1416, theta5 = -1.5287, theta6 = 3.0521}, {theta1 =
1.8314-.28856*I, theta2 = 1.5716-1.3116*I, theta3 = 1.5700+1.3116*I, theta4
= -0.92287e-1-4.0819*I, theta5 = 3.1416, theta6 = 1.9365-4.3538*I}
> math_operations := 188*additions+912*multiplications+228*functions
> total_cost := 2924*flop
> #####
```

# APÊNDICE M

```
> #####
> # Teste de Soluções para um manipular 6R com pulso não esférico
> #####

> #####
> #Determinação da Matriz de Transformação Homogênea
> #####
> # Entradas
> #####
> with(Units[Natural]);
> theta1 := 0*'deg';
> theta2 := 0*'deg';
> theta3 := 0*'deg';
> theta4 := 0*'deg';
> theta5 := 0*'deg';
> theta6 := 0*'deg';
> a2 := 432;
> a3 := 20;
> d3 := 125;
> d4 := 430;
> d5 := 20;
> Matrix(4, 4, [[cos(theta1), -sin(theta1), 0, 0], [sin(theta1),
cos(theta1), 0, 0], [0, 0, 1, 0], [0, 0, 0, 1]]) . Matrix(4, 4,
[[cos(theta2), -sin(theta2), 0, 0], [0, 0, 1, 0], [-sin(theta2), -
cos(theta2), 0, 0], [0, 0, 0, 1]]) . Matrix(4, 4, [[cos(theta3), -
sin(theta3), 0, a2], [sin(theta3), cos(theta3), 0, 0], [0, 0, 1, d3], [0,
0, 0, 1]]) . Matrix(4, 4, [[cos(theta4), -sin(theta4), 0, a3], [0, 0, 1,
d4], [-sin(theta4), -cos(theta4), 0, 0], [0, 0, 0, 1]]) . Matrix(4, 4,
[[cos(theta5), -sin(theta5), 0, 0], [0, 0, -1, d5], [sin(theta5),
cos(theta5), 0, 0], [0, 0, 0, 1]]) . Matrix(4, 4, [[cos(theta6), -
sin(theta6), 0, 0], [0, 0, 1, 0], [-sin(theta6), -cos(theta6), 0, 0], [0,
0, 0, 1]]);
> #####
> # Saídas
> #####
> Matrix(4, 4, [[1, 0, 0, 452], [0, -1, 0, 105], [0, 0, -1, -430], [0, 0,
0, 1]])
> #####

> #####
> #Configuração de Juntas 1 {theta1 = 0., theta2 = 0., theta3 = 0.,
> #theta4 = 0., theta5 = 0., theta6 = 0.}
> #####
> # Entradas
> #####
> with(Units[Natural]);
> theta1 := 0;
> theta2 := 0;
> theta3 := 0;
> theta4 := 0;
```

```
> theta5 := 0;
> theta6 := 0;
> a2 := 432;
> a3 := 20;
> d3 := 125;
> d4 := 430;
> d5 := 20;
> Matrix(4, 4, [[cos(theta1), -sin(theta1), 0, 0], [sin(theta1),
cos(theta1), 0, 0], [0, 0, 1, 0], [0, 0, 0, 1]]) . Matrix(4, 4,
[[cos(theta2), -sin(theta2), 0, 0], [0, 0, 1, 0], [-sin(theta2), -
cos(theta2), 0, 0], [0, 0, 0, 1]]) . Matrix(4, 4, [[cos(theta3), -
sin(theta3), 0, a2], [sin(theta3), cos(theta3), 0, 0], [0, 0, 1, d3], [0,
0, 0, 1]]) . Matrix(4, 4, [[cos(theta4), -sin(theta4), 0, a3], [0, 0, 1,
d4], [-sin(theta4), -cos(theta4), 0, 0], [0, 0, 0, 1]]) . Matrix(4, 4,
[[cos(theta5), -sin(theta5), 0, 0], [0, 0, -1, d5], [sin(theta5),
cos(theta5), 0, 0], [0, 0, 0, 1]]) . Matrix(4, 4, [[cos(theta6), -
sin(theta6), 0, 0], [0, 0, 1, 0], [-sin(theta6), -cos(theta6), 0, 0], [0,
0, 0, 1]]);
> #####
> # Saídas
> #####
> Matrix(4, 4, [[1, 0, 0, 452], [0, -1, 0, 105], [0, 0, -1, -430], [0, 0,
0, 1]])
> #####

> #####
> #Configuração de Juntas 2 {theta1 = 0., theta2 = 1.5209,
> #theta3 = -3.0486, theta4 = 0., theta5 = 1.5277, theta6 = 0.}
> #####
> # Entradas
> #####
> with(Units[Natural]);
> theta1 := 0;
> theta2 := 1.5209;
> theta3 := -3.0486;
> theta4 := 0;
> theta5 := 1.5277;
> theta6 := 0;
> a2 := 432;
> d3 := 125;
> a3 := 20;
> d4 := 430;
> Matrix(4, 4, [[cos(theta1), -sin(theta1), 0, 0], [sin(theta1),
cos(theta1), 0, 0], [0, 0, 1, 0], [0, 0, 0, 1]]) . Matrix(4, 4,
[[cos(theta2), -sin(theta2), 0, 0], [0, 0, 1, 0], [-sin(theta2), -
cos(theta2), 0, 0], [0, 0, 0, 1]]) . Matrix(4, 4, [[cos(theta3), -
sin(theta3), 0, a2], [sin(theta3), cos(theta3), 0, 0], [0, 0, 1, d3], [0,
0, 0, 1]]) . Matrix(4, 4, [[cos(theta4), -sin(theta4), 0, a3], [0, 0, 1,
d4], [-sin(theta4), -cos(theta4), 0, 0], [0, 0, 0, 1]]) . Matrix(4, 4,
[[cos(theta5), -sin(theta5), 0, 0], [0, 0, -1, d5], [sin(theta5),
cos(theta5), 0, 0], [0, 0, 0, 1]]) . Matrix(4, 4, [[cos(theta6), -
sin(theta6), 0, 0], [0, 0, 1, 0], [-sin(theta6), -cos(theta6), 0, 0], [0,
0, 0, 1]]);
> #####
> # Saídas
> #####
```

```
> Matrix(4, 4, [[.999999999951847, 0, -0.5214571830e-11, 452.008673623161],
[0, -1, 0, 105], [-0.5214571830e-11, 0, -.999999999951847, -
430.006603302526], [0, 0, 0, 1]])
> #####

> #####
> #Configuração de Juntas 3 {theta1 = -0.83106e-1, theta2 = 0.,
> #theta3 = 0., theta4 = 2.6020, theta5 = 0., theta6 = -2.6851}
> #####
> # Entradas
> #####
> with(Units[Natural]);
> theta1 := -0.83106e-1;
> theta2 := 0;
> theta3 := 0;
> theta4 := 2.6020;
> theta5 := 0;
> theta6 := -2.6851;
> a2 := 432;
> d3 := 125;
> a3 := 20;
> d4 := 430;
> Matrix(4, 4, [[cos(theta1), -sin(theta1), 0, 0], [sin(theta1),
cos(theta1), 0, 0], [0, 0, 1, 0], [0, 0, 0, 1]]) . Matrix(4, 4,
[[cos(theta2), -sin(theta2), 0, 0], [0, 0, 1, 0], [-sin(theta2), -
cos(theta2), 0, 0], [0, 0, 0, 1]]) . Matrix(4, 4, [[cos(theta3), -
sin(theta3), 0, a2], [sin(theta3), cos(theta3), 0, 0], [0, 0, 1, d3], [0,
0, 0, 1]]) . Matrix(4, 4, [[cos(theta4), -sin(theta4), 0, a3], [0, 0, 1,
d4], [-sin(theta4), -cos(theta4), 0, 0], [0, 0, 0, 1]]) . Matrix(4, 4,
[[cos(theta5), -sin(theta5), 0, 0], [0, 0, -1, d5], [sin(theta5),
cos(theta5), 0, 0], [0, 0, 0, 1]]) . Matrix(4, 4, [[cos(theta6), -
sin(theta6), 0, 0], [0, 0, 1, 0], [-sin(theta6), -cos(theta6), 0, 0], [0,
0, 0, 1]]);
> #####
> # Saídas
> #####
> Matrix(4, 4, [[1.00000000003393, -0.5999917413e-5, 0, 452.000356528767],
[-0.5999917413e-5, -1.00000000003393, 0, 105.000032524324], [0, 0, -1, -
430], [0, 0, 0, 1]])
> #####

> #####
> #Configuração de Juntas 4 {theta1 = -2.6851, theta2 = 3.1416,
> #theta3 = -3.0486, theta4 = 0., theta5 = -0.92956e-1, theta6 = -2.6851}
> #####
> # Entradas
> #####
> with(Units[Natural]);
> theta1 := -2.6851;
> theta2 := 3.1416;
> theta3 := -3.0486;
> theta4 := 0;
> theta5 := -0.92956e-1;
> theta6 := -2.6851;
> a2 := 432;
> d3 := 125;
```

```

> a3 := 20;
> d4 := 430;
> Matrix(4, 4, [[cos(theta1), -sin(theta1), 0, 0], [sin(theta1),
cos(theta1), 0, 0], [0, 0, 1, 0], [0, 0, 0, 1]]) . Matrix(4, 4,
[[cos(theta2), -sin(theta2), 0, 0], [0, 0, 1, 0], [-sin(theta2), -
cos(theta2), 0, 0], [0, 0, 0, 1]]) . Matrix(4, 4, [[cos(theta3), -
sin(theta3), 0, a2], [sin(theta3), cos(theta3), 0, 0], [0, 0, 1, d3], [0,
0, 0, 1]]) . Matrix(4, 4, [[cos(theta4), -sin(theta4), 0, a3], [0, 0, 1,
d4], [-sin(theta4), -cos(theta4), 0, 0], [0, 0, 0, 1]]) . Matrix(4, 4,
[[cos(theta5), -sin(theta5), 0, 0], [0, 0, -1, d5], [sin(theta5),
cos(theta5), 0, 0], [0, 0, 0, 1]]) . Matrix(4, 4, [[cos(theta6), -
sin(theta6), 0, 0], [0, 0, 1, 0], [-sin(theta6), -cos(theta6), 0, 0], [0,
0, 0, 1]]);
> #####
> # Saídas
> #####
> Matrix(4, 4, [[.999999999322015, 0.3781262525e-9, 0.394945821308723e-4,
452.018192355595], [-0.3781263080e-9, -.99999999906298,
0.193953167710315e-4, 105.002646157679], [0.394945821308744e-4, -
0.193953167710246e-4, -.999999999044330, -429.995951209629], [0, 0, 0, 1]])
> #####

> #####
> #Configuração de Juntas 5 {theta1 = -2.6851, theta2 = 1.6207,
> #theta3 = 0., theta4 = 0., theta5 = -1.6207, theta6 = -2.6851}
> #####
> # Entradas
> #####
> with(Units[Natural]);
> theta1 := -2.6851;
> theta2 := 1.6207;
> theta3 := 0;
> theta4 := 0;
> theta5 := -1.6207;
> theta6 := -2.6851;
> a2 := 432;
> d3 := 125;
> a3 := 20;
> d4 := 430;
> Matrix(4, 4, [[cos(theta1), -sin(theta1), 0, 0], [sin(theta1),
cos(theta1), 0, 0], [0, 0, 1, 0], [0, 0, 0, 1]]) . Matrix(4, 4,
[[cos(theta2), -sin(theta2), 0, 0], [0, 0, 1, 0], [-sin(theta2), -
cos(theta2), 0, 0], [0, 0, 0, 1]]) . Matrix(4, 4, [[cos(theta3), -
sin(theta3), 0, a2], [sin(theta3), cos(theta3), 0, 0], [0, 0, 1, d3], [0,
0, 0, 1]]) . Matrix(4, 4, [[cos(theta4), -sin(theta4), 0, a3], [0, 0, 1,
d4], [-sin(theta4), -cos(theta4), 0, 0], [0, 0, 0, 1]]) . Matrix(4, 4,
[[cos(theta5), -sin(theta5), 0, 0], [0, 0, -1, d5], [sin(theta5),
cos(theta5), 0, 0], [0, 0, 0, 1]]) . Matrix(4, 4, [[cos(theta6), -
sin(theta6), 0, 0], [0, 0, 1, 0], [-sin(theta6), -cos(theta6), 0, 0], [0,
0, 0, 1]]);
> #####
> # Saídas
> #####
> Matrix(4, 4, [[1.00000000010102, -0.4432232359e-11, 0, 452.011884201337],
[0.4432232359e-11, -1.000000000009417, 0.3469446952e-17, 104.999548298594],
[0, 0, -1.00000000001120, -429.987617754300], [0, 0, 0, 1]])
> #####

```

```

> #####
> #Configuração de Juntas 6 {theta1 = -2.6188-0.60089e-1*I,
> #theta2 = 3.1416, theta3 = 3.1416, theta4 = 1.8665-1.1349*I,
> #theta5 = 0., theta6 = 1.8005+1.0748*I}
> #####
> # Entradas
> #####
> with(Units[Natural]);
> theta1 := -2.6188-0.60089e-1*I;
> theta2 := 3.1416;
> theta3 := 3.1416;
> theta4 := 1.8665-1.1349*I;
> theta5 := 0;
> theta6 := 1.8005+1.0748*I;
> a2 := 432;
> a3 := 20;
> d3 := 125;
> d4 := 430;
> d5 := 20;;
> a2 := 432;
> d3 := 125;
> a3 := 20;
> d4 := 430;
> Matrix(4, 4, [[cos(theta1), -sin(theta1), 0, 0], [sin(theta1),
cos(theta1), 0, 0], [0, 0, 1, 0], [0, 0, 0, 1]]) . Matrix(4, 4,
[[cos(theta2), -sin(theta2), 0, 0], [0, 0, 1, 0], [-sin(theta2), -
cos(theta2), 0, 0], [0, 0, 0, 1]]) . Matrix(4, 4, [[cos(theta3), -
sin(theta3), 0, a2], [sin(theta3), cos(theta3), 0, 0], [0, 0, 1, d3], [0,
0, 0, 1]]) . Matrix(4, 4, [[cos(theta4), -sin(theta4), 0, a3], [0, 0, 1,
d4], [-sin(theta4), -cos(theta4), 0, 0], [0, 0, 0, 1]]) . Matrix(4, 4,
[[cos(theta5), -sin(theta5), 0, 0], [0, 0, -1, d5], [sin(theta5),
cos(theta5), 0, 0], [0, 0, 0, 1]]) . Matrix(4, 4, [[cos(theta6), -
sin(theta6), 0, 0], [0, 0, 1, 0], [-sin(theta6), -cos(theta6), 0, 0], [0,
0, 0, 1]]);
> #####
> # Saídas
> #####
> Matrix(4, 4, {(1, 1) = .9999965820697341+0.2933761116593464e-7*I, (1, 2)
= -0.26146896861089734e-2+0.10998657640204357e-4*I, (1, 3) =
0.12753263139150141e-4+0.44108726997463617e-6*I, (1, 4) =
452.01223111894507+0.30055763125415247e-1*I, (2, 1) = -0.2614689733213318e-
2+0.10998660885608302e-4*I, (2, 2) = -.9999965820968673-
0.29343245418547714e-7*I, (2, 3) = 0.73493986987558495e-5-
0.7654098317405894e-6*I, (2, 4) = 104.95180771978026+0.6147737040222978e-
3*I, (3, 1) = 0.12734011557032434e-4+0.44316827928320046e-6*I, (3, 2) = -
0.7382724278313082e-5+0.7643939623875249e-6*I, (3, 3) = -
.999999999460303+.0*I, (3, 4) = -429.996637768013+0.11943352280505518e-
3*I, (4, 1) = .0+.0*I, (4, 2) = .0+.0*I, (4, 3) = .0+.0*I, (4, 4) =
1.0+.0*I}, datatype = complex[8])
> #####

> #####
> #Configuração de Juntas 7 {theta1 = -0.89541e-1, theta2 = 1.5470,
> #theta3 = -3.0755, theta4 = 3.1416, theta5 = -1.5287, theta6 = 3.0521}
> #####

```

```

> # Entradas
> #####
> with(Units[Natural]);
> theta1 := -0.89541e-1;
> theta2 := 1.5470;
> theta3 := -3.0755;
> theta4 := 3.1416;
> theta5 := -1.5287;
> theta6 := 3.0521;
> a2 := 432;
> d3 := 125;
> a3 := 20;
> d4 := 430;
> Matrix(4, 4, [[cos(theta1), -sin(theta1), 0, 0], [sin(theta1),
cos(theta1), 0, 0], [0, 0, 1, 0], [0, 0, 0, 1]]) . Matrix(4, 4,
[[cos(theta2), -sin(theta2), 0, 0], [0, 0, 1, 0], [-sin(theta2), -
cos(theta2), 0, 0], [0, 0, 0, 1]]) . Matrix(4, 4, [[cos(theta3), -
sin(theta3), 0, a2], [sin(theta3), cos(theta3), 0, 0], [0, 0, 1, d3], [0,
0, 0, 1]]) . Matrix(4, 4, [[cos(theta4), -sin(theta4), 0, a3], [0, 0, 1,
d4], [-sin(theta4), -cos(theta4), 0, 0], [0, 0, 0, 1]]) . Matrix(4, 4,
[[cos(theta5), -sin(theta5), 0, 0], [0, 0, -1, d5], [sin(theta5),
cos(theta5), 0, 0], [0, 0, 0, 1]]) . Matrix(4, 4, [[cos(theta6), -
sin(theta6), 0, 0], [0, 0, 1, 0], [-sin(theta6), -cos(theta6), 0, 0], [0,
0, 0, 1]]);
> #####
> # Saídas
> #####
> Matrix(4, 4, [[.999999979197256, -0.486588026861190e-4, -
0.198542439551065e-3, 451.940596102804], [-0.486538021897270e-4, -
.999999998585281, 0.251947775171076e-4, 105.007512796025], [-
0.198543662927508e-3, -0.251851169417926e-4, -.999999980031386, -
430.077431024437], [0, 0, 0, 1]])
> #####

> #####
> #Configuração de Juntas 8 {theta1 = 1.8314-.28856*I,
> #theta2 = 1.5716-1.3116*I, theta3 = 1.5700+1.3116*I,
> #theta4 = -0.92287e-1-4.0819*I, theta5 = 3.1416,
> #theta6 = 1.9365-4.3538*I}
> #####
> # Entradas
> #####
> with(Units[Natural]);
> theta1 := 1.8314-.28856*I;
> theta2 := 1.5716-1.3116*I;
> theta3 := 1.5700+1.3116*I;
> theta4 := -0.92287e-1-4.0819*I;
> theta5 := 3.1416;
> theta6 := 1.9365-4.3538*I;;
> a2 := 432;
> d3 := 125;
> a3 := 20;
> d4 := 430;
> Matrix(4, 4, [[cos(theta1), -sin(theta1), 0, 0], [sin(theta1),
cos(theta1), 0, 0], [0, 0, 1, 0], [0, 0, 0, 1]]) . Matrix(4, 4,
[[cos(theta2), -sin(theta2), 0, 0], [0, 0, 1, 0], [-sin(theta2), -

```



```
cos(theta2), 0, 0], [0, 0, 0, 1])) . Matrix(4, 4, [[cos(theta3), -
sin(theta3), 0, a2], [sin(theta3), cos(theta3), 0, 0], [0, 0, 1, d3], [0,
0, 0, 1]]) . Matrix(4, 4, [[cos(theta4), -sin(theta4), 0, a3], [0, 0, 1,
d4], [-sin(theta4), -cos(theta4), 0, 0], [0, 0, 0, 1]]) . Matrix(4, 4,
[[cos(theta5), -sin(theta5), 0, 0], [0, 0, -1, d5], [sin(theta5),
cos(theta5), 0, 0], [0, 0, 0, 1]]) . Matrix(4, 4, [[cos(theta6), -
sin(theta6), 0, 0], [0, 0, 1, 0], [-sin(theta6), -cos(theta6), 0, 0], [0,
0, 0, 1]]);
> #####
> # Saídas
> #####
> Matrix(4, 4, {(1, 1) = .9807184853771105-0.3267537986175739e-2*I, (1, 2)
= -.19613514883894823-0.16337303466343656e-1*I, (1, 3) =
0.5064143354213364e-4-0.28840333903338856e-3*I, (1, 4) = 449.2354392429275-
82.45454915002816*I, (2, 1) = -.19613514646465546-0.16337306160949083e-1*I,
(2, 2) = -.980718482270504+0.3267540000194913e-2*I, (2, 3) = -
0.29381387527746365e-3-0.4920689604871094e-4*I, (2, 4) =
22.05238648781841+2.9191554638131265*I, (3, 1) = 0.10554595294921896e-3-
0.26855662835287893e-3*I, (3, 2) = 0.2736651843433706e-
3+0.10303676349405176e-3*I, (3, 3) = -.9999999974490915-
0.14696262804504432e-9*I, (3, 4) = -430.0025315818767-.5932998702367621*I,
(4, 1) = .0+.0*I, (4, 2) = .0+.0*I, (4, 3) = .0+.0*I, (4, 4) = 1.0+.0*I},
datatype = complex[8])
> #####
```

Salvador, 29/07/2021

## DECLARAÇÃO

Declaro para os devidos fins que o texto final apresentado para a conclusão do meu curso de Doutorado em Mecatrônica da Universidade Federal da Bahia é de minha autoria. Declaro também que quaisquer informações utilizadas neste texto, mas que sejam provenientes de outros trabalhos têm fonte claramente expressa e, quando for o caso, foram devidamente autorizadas pelo(s) respectivo(s) autor(es).



---

Nome: Sérgio Ricardo Xavier da Silva  
CPF: 924.749.265-34