UNIVERSIDADE FEDERAL DE MINAS GERAIS
PROGRAMA DE PÓS GRADUAÇÃO EM ENGENHARIA MECÂNICA (PPGMEC)

ANÁLISE DA EFICIÊNCIA COMPUTACIONAL PARA CINEMÁTICA DIRETA E INVERSA DE MANIPULADORES ROBÓTICOS UTILIZANDO MATRIZES DE TRANSFORMAÇÃO HOMOGÊNEAS E QUATÉRNIOS DUAIS

TAMES FERNANDES MARIANO

Belo Horizonte, 13 de setembro de 2018

Tames Fernandes Mariano

ANÁLISE DA EFICIÊNCIA COMPUTACIONAL PARA CINEMÁTICA DIRETA E INVERSA DE MANIPULADORES ROBÓTICOS UTILIZANDO MATRIZES DE TRANSFORMAÇÃO HOMOGÊNEAS E QUATÉRNIOS DUAIS

Dissertação Apresentada ao Programa de Pós-Graduação em Engenharia Mecânica – PPGMEC, da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para a Obtenção do Título de Mestre em Engenharia Mecânica.

Área de concentração: Controle e Automação

Orientador: Prof. Dr. Eduardo José Lima
(Universidade Federal de Minas Gerais)

Belo Horizonte
Escola de Engenharia da UFMG
2018

AGRADECIMENTOS

Aos meus anjos na Terra: pai e mãe.

RESUMO

O presente trabalho propõe um estudo detalhado da implementação de uma ferramenta matemática denominada quatérnios duais e uma análise à fundo sobre sua evolução e potencial aplicação dessa ferramenta na área da robótica. Em paralelo à essa visão é apresentado um comparativo com a forma tradicional de calcular rotação e translação na cinemática de robôs manipuladores. Os quatérnios duais têm sido massivamente empregados na robótica devido ao fato de serem computacionalmente mais eficientes na representação de informações rotacionais do que a representação com matrizes de transformação homogêneas. Assim, este trabalho tem como objetivo fornecer uma explicação detalhada através de um passo-a-passo para o uso da álgebra quaterniônica, de forma simplificada, utilizou-se de exemplos para melhor compreensão da implementação em questão. Embora haja uma grande quantidade de literatura sobre os aspectos teóricos de quatérnios duais, em poucas delas existem exemplos práticos de como realmente funciona o seu uso. Assim, ao dar uma clara noção da introdução à teoria de quatérnios duais, este documento também demonstra sua aplicação a um manipulador robótico do tipo serial com 4 Graus de Liberdade. Nesta dissertação foi possível fazer uma comparação entre a cinemática direta e inversa calculada usando a álgebra de matrizes versus a álgebra quaterniônica, verificou-se que os quatérnios são computacionalmente mais eficientes embora não aloquem menor área da memória de programa para o microcontrolador Atmel Atmega 328/P. Foram realizadas simulações e testes experimentais para comprovar os resultados.

Palavras-chave: Quatérnios Duais Unitários, Matrizes de Transformações Homogêneas, Cinemática Direta, Cinemática Inversa.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE GRÁFICOS

LISTA DE SÍMBOLOS

$\{A\}$	Sistema de coordenadas
A_P	Posição do ponto P em relação ao sistema de coordenadas $\{A\}$
$\vec{\boldsymbol{\imath}}, \vec{\boldsymbol{\jmath}}, \vec{\boldsymbol{k}}$	Vetores unitários
R_B^A	Matriz rotação que define a orientação do sistema $\{B\}$ em relação ao $\{A\}$
P_B^A	Posição da origem do sistema $\{B\}$ em relação ao $\{A\}$
T_B^A	Matriz de rotação que define o sistema $\{B\}$ em relação ao $\{A\}$

LISTA DE ABREVIAÇÕES E SIGLAS

DH	Convenção de Denavit-Hartenberg.
FLOP	Operação Ponto Flutuante (“*Floating Point Operation”*)
GDL	Graus de Liberdade
MCU	Unidade de Comando Controlado ou Microcontrolador
MTH ou MTHs	Matriz/Matrizes de Transformação Homogênea
nsa	Convenção ferramenta: normal, deslizamento(“sliding”) e apontamento
QD ou QDs	Quatérnio Dual/ Quatérnios Duais
QDU ou QDUs	Quatérnio Dual Unitário/Quatérnios Duais Unitários
UFMG	Universidade Federal de Minas Gerais
3D	Espaço com três dimensões

1 INTRODUÇÃO

Os quatérnios simples são uma representação de rotações 3D com diversas vantagens em comparação com matrizes de rotação. Contudo, os objetos rígidos não só giram, como também realizam translação. A rotação quando associada à translação é chamada de transformação rígida, qualquer deslocamento de um objeto rígido no espaço 3D pode ser descrito por uma transformação rígida. Os quatérnios duais são tidos como uma melhor representação de transformações rígidas por tratarem componentes de rotação e translação de forma independente, desta forma ele unifica a translação e a rotação em uma única variável de estado. Esta variável de estado única oferece uma forma robusta, inequívoca e computacionalmente eficiente para representar transformação rígida. Quando os movimentos se tornam mais complexos em termos de memória computacional usada, problemas passam a empregar os sistemas robóticos e os quatérnios duais são os mais indicados para solucionar tais problemas.

Levar em consideração complexidade temporal e espacial de um algoritmo é muito importante em programação. Afinal, é necessário saber se o algoritmo levará microssegundos, horaso ou dias para apresentar uma solução. Também é preciso saber qual é a quantidade de memória que um algoritmo requer para produzir uma solução. Enfim, a análise de algoritmos serve para classificar algoritmos como adequados ou menos adequados para solução de problemas. Assim, considera-se um algoritmo eficiente quando não se conhece nenhum outro funcionalmente equivalente a ele ou que possua menor complexidade.

Este trabalho se propõe a utilizar os Quatérnios Duais Unitários (QDU) como uma alternativa para cálculo da cinemática direta e inversa, ao mesmo tempo que se pretende comparar os resultados de testes de eficiência de execução de algoritmos obtidas com os resultados devido ao uso das tradicionais Matrizes de Transformação Homogênea (MTH).

Uma transformação 3D completamente rígida pode ser representada por uma componente translacional e outra rotacional através de uma matriz 4x4 homogênea, nota-se que as MTHs possuem a limitação de serem difíceis para realizar a interpolação entre as transformações (SHOEMAKE, 1985). Alternativamente, surgem os Quatérnios Duais como a melhor solução para rotações, neles as transformações podem ser gerenciadas com a vantagem de encapsular a translação e a rotação em um único estado que pode ser interpolado sem esforço utilizando oito números escalares ao invés de dezesseis como nas MTHs. Nota-se que os QDUs permitem fácil a incorporação a sistemas existentes, isso porque uma implementação com eles consegue ir de encontro às implementações com MTHs realizando poucas interferências nas montagens existentes.

A importância deste trabalho resume-se na otimização do cálculo de tarefas com manipuladores robóticos, resultando em tarefas operacionais mais ágeis já que o desempenho computacional é o foco do trabalho, bem como a simulação, modelagem e implementação de sistemas para a verificação da maneira mais eficiente de execução de algoritmos de cinemática direta e inversa.

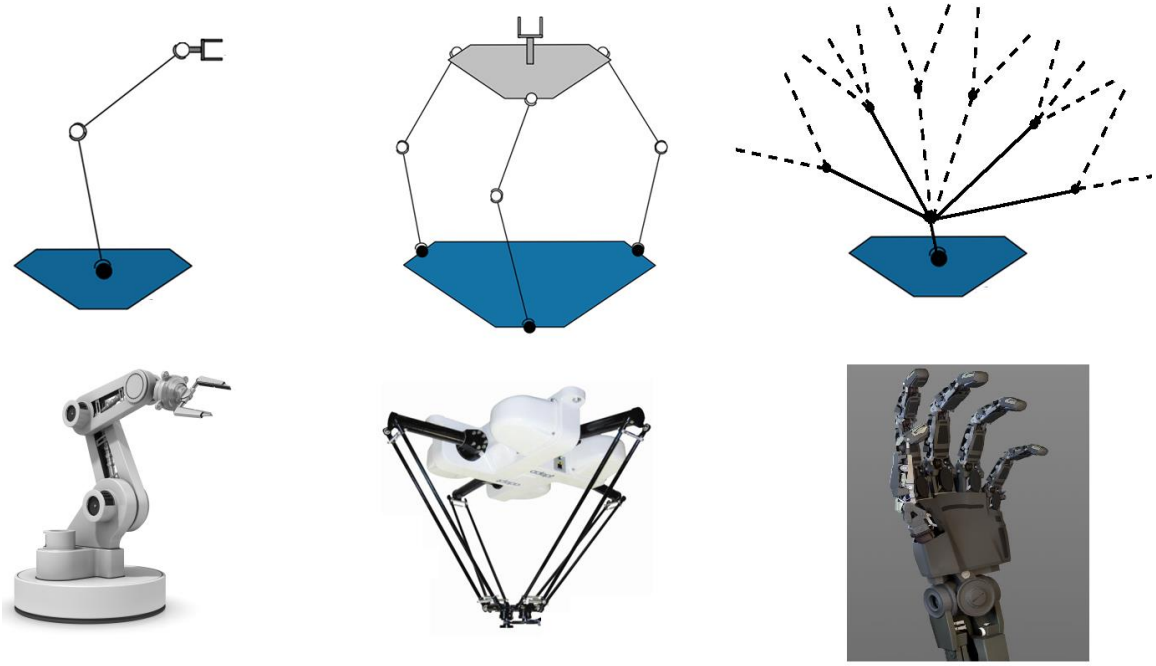
2 REVISÃO BIBLIOGRÁFICA

2.1 Conceituando a Cinemática de Robôs Manipuladores

2.1.1 Convenção Denavit-Hartenberg

A classificação de robôs quanto à sua arquitetura e estrutura é fundamental para identificar as diferenças entre eles que influenciarão nos cálculos de cinemática e controle. A arquitetura de um robô se refere à maneira como os respectivos elos e juntas se conectam, já a estrutura de um robô considera o posicionamento dos eixos das juntas para formação da cadeia cinemática do robô. De maneira genérica, um robô serial é aquele cuja cadeia cinemática é aberta parecida com a de um braço antropomórfico. Esse tipo de robô é formado por uma única cadeia cinemática onde apenas um elo está conectado à base e o final da cadeia cinemática é livre para se mover no espaço. Um robô paralelo é formado por no mínimo duas cadeias cinemáticas independentes, e estas várias cadeias cinemáticas se ligam a um outro elemento fixo que constituirá a base do efetuador (mecanismo de cadeia fechada tem o efetuador ligado a uma base fixa), esses robôs resultam na maioria das vezes em sistemas redundantes com mais atuadores que o número de graus de liberdade controlados pelo efetuador (BRANDSTÖTTER, 2016, p.6).

Dombre e Khalil(2007, p.3) explicam que uma modelagem sistemática de robôs requer um método apropriado para a descrição da morfologia deste. Vários métodos e notações foram propostos e o mais utilizado é o de Denavit-Hartenberg (DH), desenvolvido para estruturas abertas simples de robôs seriais, como a mostrada na Figura 1–(a). Este método DH apresenta ambiguidades quando é aplicado a robôs com cadeias cinemáticas fechadas ou robôs estruturados em árvore, Figura 1 – (b) e (c). Quando o modelo DH convencional não é aplicável, uma alternativa é a notação de Khalil e Kleinfinger (1986) que é um modelo que prevê modificações no modelo DH original e que permite a descrição unificada de robôs complexos e com estruturas mecânicas articuladas, além da tradicional descrição de robôs seriais.

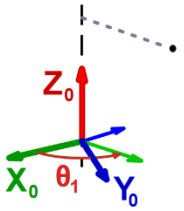
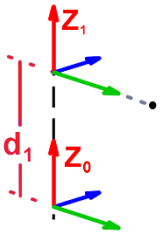
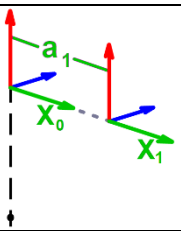
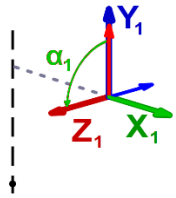
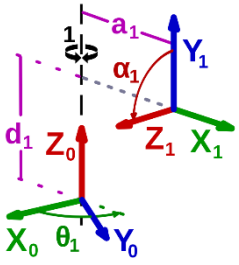


(a) Braço robótico Estrutura Cadeia Aberta | (b) Robô Paralelo Estrutura Cadeia Fechada | (c) Robô modelo Dexterous Estrutura em árvore

Figura 1: Desenhos esquemáticos e exemplificação real de robôs dada a estrutura
Fonte: YAVUZ, 2009; PARK,2015; CHEN, HAN E PENG, 2014.

Em análise à robôs seriais de estruturas abertas simples, Siciliano(2009) explica a convenção de Denavit-Hartenberg. Esta convenção precisou existir considerando que os sistemas de coordenadas de cada junta do manipulador não poderiam ser definidos arbitrariamente, por uma questão de consistência e eficiência computacional o modelo adotado permitiu a localização destes sistemas sem gerar ambiguidades. A Tabela 1 mostra a convenção Denavit-Hartenberg em sua forma tradicional:

Tabela 1: Convenção Denavit-Hartenberg

$\boldsymbol{R_{z,\theta}} \rightarrow$	rotação de θ_i em torno do eixo z_{i-1} (ângulo de junta)	
$\boldsymbol{T_{z,d}} \rightarrow$	translação de d_i ao longo do eixo z_{i-1} (distância entre origens)	
$\boldsymbol{T_{x,a}} \rightarrow$	translação de a_i ao longo do eixo x_{i-1} (comprimento do elo)	
$\boldsymbol{R_{x,\alpha}} \rightarrow$	rotação de α_i em torno do eixo x_{i-1} (torção do elo)	
	Rotação e Translação de Denavit Hartenberg $\boldsymbol{R_{z,\theta} * T_{z,d} * T_{x,a} * R_{x,\alpha}}$	

Fonte: (commons.wikimedia.org/wiki/File:Denavit-Hartenberg-Transformation)

Spong, et. al (2004, p.64) ao explicar o modelo Denavit-Hartenberg (DH) , afirma que não é possível representar qualquer matriz de transformação homogênea usando somente quatro parâmetros $(\theta, d,$ a, $\alpha)$, desta forma para garantir que somente as transformações homogêneas passíveis de serem expressas no produto das matrizes de transformações básicas sejam elegíveis $(\boldsymbol{R_{z,\theta} * T_{z,d} * T_{x,a} * R_{x,\alpha}})$ de modo que estas permitirão a existência da transformação homogênea final livre de singularidades, duas outras proposições devem ser atendidas no modelo DH juntamente com os quatro parâmetros de modo que permita especificar

qualquer transformação homogênea resultante no manipulador robótico : **(1)** o eixo x da junta posterior deve ser perpendicular ao eixo z da junta anterior ($\boldsymbol{x}_i \perp \boldsymbol{z}_{i-1}$), de modo a garantir que o produto escalar destes dois eixos seja nulo e portanto os ângulos θ e α existam; **(2)** o eixo x da junta posterior deve interceptar o eixo z da junta anterior ($\boldsymbol{x}_i \cap \boldsymbol{z}_{i-1}$), garantindo que o deslocamento de um elo para o outro resulte em uma equação que possa ser expressa pela combinação linear dos dois eixos. A Figura 2 a seguir ilustra as duas proposições anteriormente citadas:

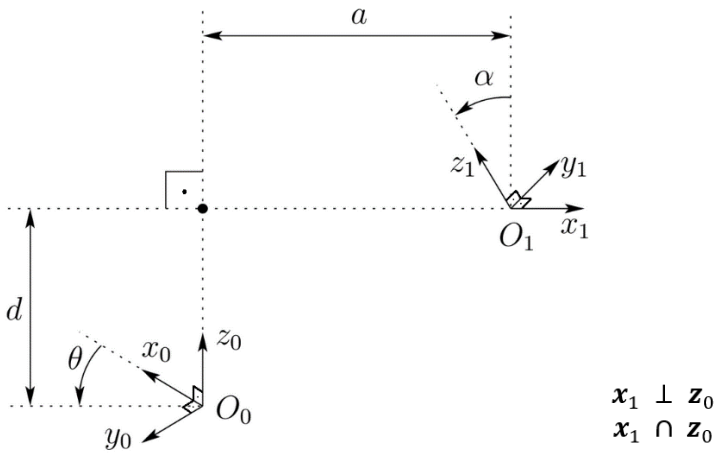


Figura 2: Coordenadas de duas juntas consecutivas obedecendo as duas proposições de Denavit Hartenberg
Fonte: SPONG, et. al,2004, p.65

2.1.2 Convenção nsa

O Sistema nsa definido por convenção para o efetuador está ilustrado na Figura 3, onde o eixo z representa a direção de ataque ou direção para a qual o efetuador está apontando ("approach"), o eixo y é a direção de escorregamento para abertura e fechamento da garra/ferramenta ("sliding") e o eixo x é direção normal que é ortogonal ao plano formado pelas direções z e y ("approach" e "sliding") saindo pela regra da mão direita. Esta convenção é também conhecida como movimentos de rolagem, arfagem e guinada (roll, pitch e yaw – Z_ω, $Y_\emptyset$ e X_γ) ou ainda pode ser chamada X-Y-Z ângulos fixos. A Figura 3 ilustra as coordenadas do efetuador obedecendo a convenção nsa (SCHILLING, 2003, p.53; CRAIG, 2005, p.42).

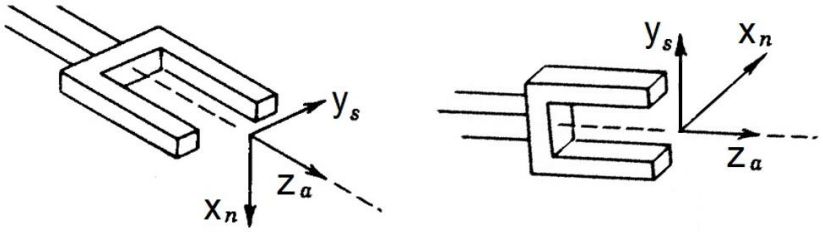


Figura 3: Sistema de Coordenadas do efetuador convenção nsa
Fonte: Autoria Própria

2.1.3 Cinemática Direta

Um robô cuja configuração é conhecida, ou seja, o comprimento de todos os elos e ângulos articulares do robô foram medidos e determinados. O cálculo da posição e da orientação

do efetuador do robô é chamado de análise cinemática direta. Em outras palavras, se todas as variáveis articulares do robô são dadas, então, com o uso das equações de cinemática direta é possível calcular onde o terminal final do robô está em qualquer instante de tempo (NIKU,2015, p.49).

Para um manipulador antropomórfico de n graus de liberdade têm-se a relação dada pela Figura 4 esquemática à seguir:

Figura 4: Desenho esquemático cinemática direta
Fonte: Autoria Própria

x_0 , y_0 e z_0 são as coordenadas cartesianas do efetuador no sistema de coordenadas da base. Estas que podem ser representadas pelo vetor de posição d, onde n, s, a são vetores ortonormais que descrevem a orientação do efetuador em relação a base (ROMANO, 2002). Estes vetores formam a matriz de rotação R. Para este tipo de manipulador, há n juntas e n + 1 elos conectados, incluindo o elo 0 que é a base do robô. Considerando que cada junta i conecta os elos (i − 1) e (i), pode-se representar a posição (ROMANO, 2002) e orientação do efetuador em relação ao sistema da base, usando a sequência de matrizes de transformação homogênea (SPONG et al., 2005)

Como visto na cinemática direta é possível determinar as coordenadas x, y e z e a orientação do efetuador de um manipulador se forem conhecidas as variáveis de junta θi. Essa tarefa é a chamada modelagem cinemática direta e pode ser realizada usando a convenção de Denavit-Hartenberg de uma maneira bem prática.

2.1.4 Cinemática Inversa

Para colocar o efetuador do robô em um local e orientação desejados, é preciso saber quais devem ser os ângulos das articulações do robô tal que, nesses valores, o efetuador estará na posição e orientação desejadas. Isso é chamado de análise cinemática inversa, o que significa que em vez de substituir as variáveis conhecidas do robô nas equações de cinemática direta dele, é preciso encontrar os inversos destas equações que permita saber os valores articulares necessários para colocar o robô no local e orientação desejados. Na realidade, as equações de cinemática inversa são mais importantes, já que o controlador do robô calculará os valores articulares usando essas equações e manejará o robô para a posição e a orientação desejadas. Desenvolve-se primeiro as equações de cinemática direta de robôs, e em seguida, usando essas equações, calcula-se as equações da cinemática inversa (NIKU, 2015, p.49).

Para realizar o comando de um manipulador é necessário o modelo inverso, nele é dado a posição e orientação desejadas para o efetuador, deseja-se saber os valores de θi em cada junta. Este é o problema da cinemática inversa de posição representado na Figura 5 (SPONG et al., 2005, p. 85)

Figura 5: Desenho esquemático cinemática inversa
Fonte: Autoria Própria

A cinemática inversa é muito parecida com a equação cinemática direta, porém com as entradas e saídas invertidas. O problema da cinemática inversa de posição pode ser bastante difícil de se resolver pois resulta em doze equações não lineares e n incógnitas (SPONG et al., 2005).

Quando a cinemática inversa existe, a forma explicita encontrada normalmente resulta em um conjunto de várias soluções (raramente o resultado da cinemática inversa possui uma única solução). Existem diversos métodos para solução da cinemática inversa, o método de Paul é um método que depende de cada robô individualmente e é muito usado em robôs industriais; o método de Pieper que permite a solução para robôs com seis graus de liberdade dos quais três juntas (de revolução ou prismáticas) possuem eixos que se interceptam; o método Raghavan e Rothque dá uma solução geral para robôs com seis graus de liberdade e polinômios para até o 16º grau. Quando não é possível calcular uma forma explicita da cinemática inversa, a alternativa é o cálculo de uma solução particular usando procedimentos numéricos (DOMBRE e KHALIL, 2007, p.7).

2.2 Cálculo da Cinemática

2.2.1 Matrizes de Transformação Homogênea

Em 1850 o matemático inglês J. Sylvester introduziu o termo matriz para uma ordenação retangular de números os quais ele poderia representar um mapeamento linear entre duas dimensões finitas de vetores espaciais (UEBERHUBER, 1995, p.193).

Niku (2015, p.37) afirma que, por definição, uma transformação é a realização de um movimento no espaço. Uma transformação é uma mudança no estado de um referencial (localização e orientação) e, portanto, um vetor, um objeto, ou um referencial móvel se movendo no espaço em relação a um sistema de referência fixo poderá ser representado como um referencial. A transformação pode se apresentar como uma das seguintes formas: uma translação pura, uma rotação pura em torno de um eixo ou ainda uma combinação de translações e/ ou rotações. A movimentação de um corpo rígido no espaço pode ser modelada como uma transformação linear representada por matrizes de transformação. Diversas transformações lineares podem ser combinadas em uma única matriz, denominada matriz afim, consistindo na preservação da distância entre pontos distintos e na colinearidade entre pontos, isto é, três pontos que se encontram em uma linha continuam a ser colineares após a transformação. Busca-se então uma representação para a translação e rotação de um objeto já que esses são os tipos de movimentação possíveis em um robô (Equação 1).

$$\boldsymbol{H} = \begin{bmatrix} R_{xx} & R_{xy} & R_{xz} & x_E \\ R_{yx} & R_{yy} & R_{yz} & y_E \\ R_{xx} & R_{yz} & R_{zz} & z_E \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{Eq.1}$$

Na representação padrão cada uma das matrizes podem ser obtidas por meio de quatro transformações homogêneas básicas, de acordo com a estrutura rígida do robô manipulador resultando nas Equações 2 e 3 (SPONG et al., 2005):

$$\boldsymbol{H}_{DH_i}^{i-1} = \boldsymbol{R}_{z,\theta} * \boldsymbol{T}_{z,d} * \boldsymbol{T}_{x,\mathrm{a}} * \boldsymbol{R}_{x,\alpha} \qquad \text{Eq.2}$$

$$\boldsymbol{H}_{DH_i}^{i-1} = \begin{bmatrix} C_{\theta_i} & -S_{\theta_i} & 0 & 0 \\ S_{\theta_i} & C_{\theta_i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & \alpha_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C_{\alpha_i} & -S_{\alpha_i} & 0 \\ 0 & S_{\alpha_i} & C_{\alpha_i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{Eq.3}$$

$$C_{\theta_i} = \cos(\theta_i) \qquad S_{\theta_i} = \operatorname{sen}(\theta_i)$$

$$C_{\alpha_i} = \cos(\alpha_i) \qquad S_{\alpha_i} = \operatorname{sen}(\alpha_i)$$

A matriz homogênea dada pela convenção Denavit-Hartenberg para representação de uma estrutura rígida pode ser produzida após resolução das multiplicações matriciais, o resultado exibe doze números (sem contar a última linha da matriz de transformação homogênea) que representam juntos rotação e translação das juntas conforme representado pela Equação 4 (SPONG et al., 2005 p. 69).

$$\boldsymbol{H}_{DH_i}^{i-1} = \begin{bmatrix} \cos\theta_i & -\operatorname{sen}\theta_i \cos\alpha_i & \operatorname{sen}\theta_i \operatorname{sen}\alpha_i & \mathbf{a}_i \cos\theta_i \\ \operatorname{sen}\theta_i & \cos\theta_i \cos\alpha_i & -\cos\theta_i \operatorname{sen}\alpha_i & \mathbf{a}_i \operatorname{sen}\theta_i \\ 0 & \operatorname{sen}\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{Eq.4}$$

A translação é dada pela última coluna da matriz final na Equação 1 resultando na Equação 5:

$$p = \begin{bmatrix} \boldsymbol{x_E} \\ \boldsymbol{y_E} \\ \boldsymbol{z_E} \end{bmatrix} \qquad \text{Eq.5}$$

A matriz global de rotação nsa, também chamada de matriz global roll, pitch e yaw é dada pelas Equações 6 e 7 e rotações em torno dos eixos Equações 8 , 9 e 10:

$$R_{ZYX} = R_{Z_\omega} * R_{Y_\emptyset} * R_{X_\gamma} \qquad \text{Eq.6}$$

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} C_\omega & -S_\omega & 0 \\ S_\omega & C_\omega & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_\emptyset & 0 & S_\emptyset \\ 0 & 1 & 0 \\ -S_\emptyset & 0 & C_\emptyset \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & C_\gamma & -S_\gamma \\ 0 & S_\gamma & C_\gamma \end{bmatrix}$$

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} C_\omega C_\emptyset & -S_\omega & C_\omega S_\emptyset \\ S_\omega C_\emptyset & C_\omega & S_\omega S_\emptyset \\ -S_\emptyset & 0 & C_\emptyset \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & C_\gamma & -S_\gamma \\ 0 & S_\gamma & C_\gamma \end{bmatrix}$$

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} (C_\omega\, C_\emptyset) & (C_\omega\, S_\emptyset\, S_\gamma - S_\omega C_\gamma) & (C_\omega\, S_\emptyset\, C_\gamma + S_\omega S_\gamma) \\ (S_\omega\, C_\emptyset) & (S_\omega\, S_\emptyset\, S_\gamma + C_\omega C_\gamma) & (S_\omega\, S_\emptyset\, C_\gamma - C_\omega S_\gamma) \\ -S_\emptyset & (\,C_\emptyset\, S_\gamma) & (\,C_\emptyset\, C_\gamma) \end{bmatrix} \qquad \text{Eq.7}$$

$$\emptyset_{\text{rot.Y}} = \operatorname{atan2}\left(\frac{-r_{31}}{\sqrt{{r_{11}}^2 + {r_{21}}^2}}\right) \qquad \text{Eq.8}$$

$$\omega_{\text{rot.Z}} = \operatorname{atan2}\left(\frac{{r_{21}}/{C_\emptyset}}{{r_{11}}/{C_\emptyset}}\right) = \operatorname{atan2}\left(\frac{r_{21}}{r_{11}}\right) \qquad \text{Eq.9}$$

$$\gamma_{\text{rot.X}} = \text{atan2}\left(\frac{r_{32}/C_{\emptyset}}{r_{33}/C_{\emptyset}}\right) = \text{atan2}\left(\frac{r_{32}}{r_{33}}\right) \quad \text{Eq.10}$$

A inversa de uma Matriz de Transformação Homogênea (MTH) é dada pela Equação 11.

$$\boldsymbol{H^{-1}} = \begin{bmatrix} n_x & n_y & n_z & -n^T p \\ s_x & s_y & s_z & -s^T p \\ a_x & a_y & a_z & -a^T p \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} n_x & n_y & n_z & (-n_x p_x - n_y p_y - n_z p_z) \\ s_x & s_y & s_z & (-s_x p_x - s_y p_y - s_z p_z) \\ a_x & a_y & a_z & (-a_x p_x - a_y p_y - a_z p_z) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Eq.11}$$

2.2.2 Quatérnios Duais Unitários

Uma transformação 3D completamente rígida é composta de uma componente translacional e outra rotacional, que é tradicionalmente calculada através de uma matriz 4x4 homogênea. No entanto, a maior complexidade na realização da interpolação entre as transformações quando usadas as matrizes homogêneas recebe destaque na literatura. Alternativamente, surgem os quatérnios duais como a melhor solução para rotações, neles as transformações podem ser gerenciadas com a vantagem de encapsular a translação e a rotação em um único estado que pode ser interpolado sem esforço (SHOEMAKE, 1985)**.**

A comunidade de roboticistas tem dado cada vez mais atenção especial aos QDUs tanto para modelagem cinemática para robôs com n graus de liberdade quanto para estratégias de controle cinemático, isso porque boa parte dos autores destacam a economia de memória e a eficiência computacional dos QDUs como superiores em relação às MTHs (WANG e YU,2011; GOUASMI, OUALI e BRAHIM, 2012).

Os QDUs já foram empregados de forma efetiva na computação gráfica, em visões computacionais, na navegação e etc. As operações com QDs destacam-se também pela maior robustez à erros numéricos (OZGURA e MEZOUAR, 2016). A falta de robustez nos cálculos é ruim por resultar em problemas de precisão numérica (ou estabilidade numérica) que surgem devido à aritmética dos computadores, pois os números adotados nos cálculos são normalmente números de ponto flutuante de precisão fixa ou até mesmo inteiros, após realizar vários cálculos usando o computador nota-se que ao invés dos números reais exatos esperados tem-se em teoria precisões arbitrárias, levando a respostas incorretas (MEI, TIPPER e XU, 2014).

Os quatérnios duais (QD) são uma ferramenta matemática com grande importância na mecânica clássica, uma vez que são também capazes de representar problemas complexos de forma compacta e unificada. Um quatérnio dual é capaz de combinar, em uma única variável, componentes lineares e rotacionais que podem ser interpoladas, concatenadas e transformadas obedecendo um conjunto de regras algébricas (KENWRIGHT,2012).

A comunidade de roboticistas tem dado cada vez mais atenção especial aos QDUs tanto para modelagem cinemática quanto para propostas de controle, isso porque a economia de memória e a eficiência computacional dos QDUs apresentam superioridade em relação às MTHs. (WANG e YU,2011; GOUASMI, OUALI e BRAHIM,2012).

Dentre as vantagens dos quatérnios duais **(**GE et.al,1998; SCHILLING,2011;

KENWRIGHT,2012; OZGURA e MEZOUAR, 2016):

- ✓ Combinam rotação e translação em uma única variável de estado;
- ✓ São uma representação compacta (8 números escalares);
- ✓ Eles são facilmente convertidos em outras formas (por exemplo, forma matricial);
- ✓ Podem ser facilmente interpolados sem ambiguidade;
- ✓ Computacionalmente mais eficientes (quando comparado a matrizes de transformação homogênea);
- ✓ Podem ser incorporados em um sistema já implementado sem fazer muita interferência;
- ✓ Ausência de singularidades ao serem representados no espaço Euclidiano;
- ✓ Melhor robustez devido a erros numéricos;
- ✓ Melhor performance na modelagem cinemática de braços robóticos com n graus de liberdade e, também, no controle proporcional.

Feng e Wan (2013) afirmam que os quatérnios simples tem sido uma ferramenta popular na computação gráfica 3D há mais de 20 anos, são quatro termos contendo números reais (q_r , q_x , q_y , q_z), dos quais os três termos (q_x , q_y , q_z) são componentes de um vetor. Os quatérnios podem ser representados pela Equação 12:

$$Q = q_r + q_x\,\vec{\boldsymbol{\imath}} + q_y\,\vec{\boldsymbol{\jmath}} + q_z\,\vec{\boldsymbol{k}} \qquad \text{Eq.12}$$

ou

$$Q = q_r + \vec{\boldsymbol{q}}$$

Onde $\vec{\boldsymbol{\imath}}$, $\vec{\boldsymbol{\jmath}}$ e $\vec{\boldsymbol{k}}$ são vetores unitários associados com os eixos do sistema de coordenadas cartesianas do vetor $\vec{\boldsymbol{q}}$ e q_r representa a parte real do quatérnio.

Os quatérnios clássicos são restritos para a representação somente de rotações, no entanto, em aplicações gráficas normalmente trabalha-se com rotação composta com a translação, isto é, transformações rígidas (FENG; WAN, 2013)

Um quatérnio dual pode ser utilizado para definir um corpo rígido rotacionando um ângulo φ em torno de um eixo $\vec{\boldsymbol{u}}$ que passa pela origem conforme Equações 13, 14, 15, 16 e 17.

$$Q = \cos\left(\frac{\varphi}{2}\right) + \vec{\boldsymbol{u}}\,\sin\left(\frac{\varphi}{2}\right) \qquad \text{Eq.13}$$

$$q_1 = \cos\left(\frac{\varphi}{2}\right) \qquad \text{Eq.14}$$

$$q_2 = \boldsymbol{u_x}\,\sin\left(\frac{\varphi}{2}\right) \qquad \text{Eq.15}$$

$$q_3 = \boldsymbol{u_y}\,\sin\left(\frac{\varphi}{2}\right) \qquad \text{Eq.16}$$

$$q_4 = \boldsymbol{u_z}\,\sin\left(\frac{\varphi}{2}\right) \qquad \text{Eq.17}$$

Os quatérnios duais são entidades matemáticas cujas componentes são números duplos, e eles podem ser expressos pelas Equações 18 e 19.

$$\underline{Q} = Q + \boldsymbol{\varepsilon}\,Q_0 \qquad \text{Eq.18}$$

Onde, tanto Q quanto Q_0 são quatérnios simples e $\boldsymbol{\varepsilon}$ é a unidade dual. Um quatérnio

dual pode formular um problema de forma mais concisa, resolvê-lo mais rapidamente e em menos etapas, apresentar o resultado mais claramente para os outros, ser posto em prática com menos linhas de código e depurado sem esforço.

$$\underline{Q} = \underbrace{\left(Q_{P_0} + Q_{P_i} + Q_{P_j} + Q_{P_k}\right)}_{\text{Parte Primária (Rotação)}} + \varepsilon \underbrace{\left(Q_{D_0} + Q_{D_i} + Q_{D_j} + Q_{D_k}\right)}_{\text{Parte Dual (Translação)}} \quad \text{Eq.19}$$

Outra representação dos quatérnios duais é dada pela Equação 20.

$$\underline{\boldsymbol{Q}} = \begin{bmatrix} Q_{P_0} \\ Q_{P_i} \\ Q_{P_j} \\ Q_{P_k} \\ Q_{D_0} \\ Q_{D_i} \\ Q_{D_j} \\ Q_{D_k} \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \\ q_7 \\ q_8 \end{bmatrix} \quad \text{Eq.20}$$

Um quatérnio dual tem uma representação unificada de rotação e translação(Equação 21):

$$\begin{cases} Q_P = r \\ Q_D = \frac{1}{2}\, t * r \end{cases} \quad \text{Eq.21}$$

Onde r é um quatérnio unitário que representa a rotação e t é um quatérnio descrevendo a translação representada pelo vetor t na Figura 6.

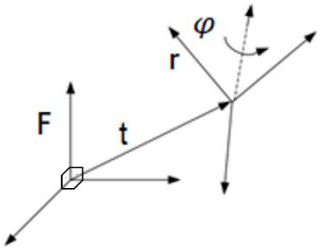


Figura 6: Transformação rígida com quatérnio dual

Os quatérnios duais representam transformações rígidas da mesma maneira como quatérnios clássicos representam rotações. Os quatérnios duais associados a algoritmos puderam ser aplicados em combinações de movimento, análises de movimento, enquadramento espacial chave, visão por computacional e hardware gráfico (FENG; WAN, 2013).

Adorno (2011) mostra a cinemática em manipuladores robóticos utilizando quatérnios duais para proceder uma sequência de multiplicações que podem ser empregadas em conjunto com os parâmetros de Denavit-Hartenberg de modo a encontrar o modelo cinemático direto. Fazendo uso dos quatérnios duais, a notação de Denavit-Hartenberg padrão fica definida como as Equações 22 e 23, onde o símbolo '■' representa a multiplicação de dois quatérnios duais definida no apêndice A.5:

$$\underline{\boldsymbol{Q}}_{DH}{}_{i}^{i-1} = \underline{\boldsymbol{R}}_{z,\theta} \blacksquare \underline{\boldsymbol{T}}_{z,d} \blacksquare \underline{\boldsymbol{T}}_{x,\mathrm{a}} \blacksquare \underline{\boldsymbol{R}}_{x,\alpha} \qquad \text{Eq.22}$$

$$\underline{\boldsymbol{Q}}_{DH}{}_{i}^{i-1} = \begin{bmatrix} \cos(\theta_i/2) \\ 0 \\ 0 \\ \operatorname{sen}(\theta_i/2) \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \blacksquare \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ d_i/2 \end{bmatrix} \blacksquare \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ \mathbf{a}_i/2 \\ 0 \\ 0 \end{bmatrix} \blacksquare \begin{bmatrix} \cos(\alpha_i/2) \\ \operatorname{sen}(\alpha_i/2) \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad \text{Eq.23}$$

A Equação 24 é análoga à definição em matrizes de transformação homogêneas com rotações puras de θ e α em torno de z e x, respectivamente, e pz e px representando translações puras de d e a ao longo de z e x, respectivamente.

$$\underline{\boldsymbol{Q}}_{DH}{}_{i}^{i-1} = \begin{bmatrix} \cos(\theta_i/2)\cos(\alpha_i/2) \\ \cos(\theta_i/2)\operatorname{sen}(\alpha_i/2) \\ \operatorname{sen}(\theta_i/2)\operatorname{sen}(\alpha_i/2) \\ \operatorname{sen}(\theta_i/2)\cos(\alpha_i/2) \\ -(\mathbf{a}_i/2)\cos(\theta_i/2)\operatorname{sen}(\alpha_i/2) - (d_i/2)\operatorname{sen}(\theta_i/2)\cos(\alpha_i/2) \\ +(\mathbf{a}_i/2)\cos(\theta_i/2)\cos(\alpha_i/2) - (d_i/2)\operatorname{sen}(\theta_i/2)\operatorname{sen}(\alpha_i/2) \\ +(\mathbf{a}_i/2)\operatorname{sen}(\theta_i/2)\cos(\alpha_i/2) + (d_i/2)\cos(\theta_i/2)\operatorname{sen}(\alpha_i/2) \\ -(\mathbf{a}_i/2)\operatorname{sen}(\theta_i/2)\operatorname{sen}(\alpha_i/2) + (d_i/2)\cos(\theta_i/2)\cos(\alpha_i/2) \end{bmatrix} \qquad \text{Eq.24}$$

Os quatérnios duais tiram vantagem em relação às matrizes homogêneas quando se trata da memória ocupada por estes, uma vez que as matrizes homogêneas exigem doze números para representar seis graus de liberdade enquanto que os quatérnios duais exigem apenas oito. (RADAVELLI, et.al, 2012)

A representação para a translação de um quatérnio dual pode ser dada pela Equação25:

$$\boldsymbol{transl}\left(\underline{\boldsymbol{Q}}\right) = 2 * \underline{\boldsymbol{Q}}_{\boldsymbol{D}} \blacksquare \underline{Q}^{*} \qquad \text{Eq.25}$$

$$\boldsymbol{transl}\left(\underline{\boldsymbol{Q}}\right) = 2 * \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \boldsymbol{Q}_{D_0} \\ \boldsymbol{Q}_{D_i} \\ \boldsymbol{Q}_{D_j} \\ \boldsymbol{Q}_{D_k} \end{bmatrix} \blacksquare \begin{bmatrix} \boldsymbol{Q}_{P_0} \\ -\boldsymbol{Q}_{P_i} \\ -\boldsymbol{Q}_{P_j} \\ -\boldsymbol{Q}_{P_k} \\ \boldsymbol{Q}_{D_0} \\ -\boldsymbol{Q}_{D_i} \\ -\boldsymbol{Q}_{D_j} \\ -\boldsymbol{Q}_{D_k} \end{bmatrix} =$$

Eq.26

$$\boldsymbol{transl}\left(\underline{\boldsymbol{Q}}\right) = 2 * \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \boldsymbol{q_5} \\ \boldsymbol{q_6} \\ \boldsymbol{q_7} \\ \boldsymbol{q_8} \end{bmatrix} \blacksquare \begin{bmatrix} \boldsymbol{q_1} \\ \boldsymbol{-q_2} \\ \boldsymbol{-q_3} \\ \boldsymbol{-q_4} \\ \boldsymbol{q_5} \\ \boldsymbol{-q_6} \\ \boldsymbol{-q_7} \\ \boldsymbol{-q_8} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \boldsymbol{x_E} \\ \boldsymbol{y_E} \\ \boldsymbol{z_E} \end{bmatrix}$$

Desenvolvendo-se a Equação 26, chega-se à translação (p) de um quatérnio dual é dada pela Equação 27:

$$p = \begin{bmatrix} -\mathbf{2} * \boldsymbol{Q_{D_0}} * \boldsymbol{Q_{P_i}} + \mathbf{2} * \boldsymbol{Q_{D_i}} * \boldsymbol{Q_{P_0}} - \mathbf{2} * \boldsymbol{Q_{D_j}} * \boldsymbol{Q_{P_k}} + \mathbf{2} * \boldsymbol{Q_{D_k}} * \boldsymbol{Q_{P_j}} \\ -\mathbf{2} * \boldsymbol{Q_{D_0}} * \boldsymbol{Q_{P_j}} + \mathbf{2} * \boldsymbol{Q_{D_i}} * \boldsymbol{Q_{P_k}} + \mathbf{2} * \boldsymbol{Q_{D_j}} * \boldsymbol{Q_{P_0}} - \mathbf{2} * \boldsymbol{Q_{D_k}} * \boldsymbol{Q_{P_i}} \\ -\mathbf{2} * \boldsymbol{Q_{D_0}} * \boldsymbol{Q_{P_k}} - \mathbf{2} * \boldsymbol{Q_{D_i}} * \boldsymbol{Q_{P_j}} + \mathbf{2} * \boldsymbol{Q_{D_j}} * \boldsymbol{Q_{P_i}} + \mathbf{2} * \boldsymbol{Q_{D_k}} * \boldsymbol{Q_{P_0}} \end{bmatrix}$$

$$p = \begin{bmatrix} -\mathbf{2} * \boldsymbol{q_5} * \boldsymbol{q_2} + \mathbf{2} * \boldsymbol{q_6} * \boldsymbol{q_1} - \mathbf{2} * \boldsymbol{q_7} * \boldsymbol{q_4} + \mathbf{2} * \boldsymbol{q_8} * \boldsymbol{q_3} \\ -\mathbf{2} * \boldsymbol{q_5} * \boldsymbol{q_3} + \mathbf{2} * \boldsymbol{q_6} * \boldsymbol{q_4} + \mathbf{2} * \boldsymbol{q_7} * \boldsymbol{q_1} - \mathbf{2} * \boldsymbol{q_8} * \boldsymbol{q_2} \\ -\mathbf{2} * \boldsymbol{q_5} * \boldsymbol{q_4} - \mathbf{2} * \boldsymbol{q_6} * \boldsymbol{q_3} + \mathbf{2} * \boldsymbol{q_7} * \boldsymbol{q_2} + \mathbf{2} * \boldsymbol{q_8} * \boldsymbol{q_1} \end{bmatrix}$$

$$p = \begin{bmatrix} \boldsymbol{x_E} \\ \boldsymbol{y_E} \\ \boldsymbol{z_E} \end{bmatrix} = 2 * \begin{bmatrix} \boldsymbol{-q_2} & \boldsymbol{+q_1} & \boldsymbol{-q_4} & \boldsymbol{+q_3} \\ \boldsymbol{-q_3} & \boldsymbol{+q_4} & \boldsymbol{+q_1} & \boldsymbol{-q_2} \\ \boldsymbol{-q_4} & \boldsymbol{-q_3} & \boldsymbol{+q_2} & \boldsymbol{+q_1} \end{bmatrix} \begin{bmatrix} \boldsymbol{q_5} \\ \boldsymbol{q_6} \\ \boldsymbol{q_7} \\ \boldsymbol{q_8} \end{bmatrix} \qquad \text{Eq.27}$$

Em notação quaterniônica, o sistema nsa ou sistema roll, pitch e yaw possui equivalência em relação às MTHs. Os parâmetros de Euler conhecidos como quatérnio unitário cuja matriz de rotação equivalente será dada pela Equação 28, de onde podem ser extraídas as rotações em torno dos eixos Y, Z, X (Equações 29, 30 e 31, respectivamente):

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} (1 - 2{q_3}^2 - 2{q_4}^2) & 2(q_2q_3 - q_1q_4) & 2(q_2q_4 + q_1q_3) \\ 2(q_2q_3 + q_1q_4) & (1 - 2{q_2}^2 - 2{q_4}^2) & 2(q_3q_4 - q_1q_2) \\ 2(q_2q_4 - q_1q_3) & 2(q_3q_4 + q_1q_2) & (1 - 2{q_2}^2 - 2{q_3}^2) \end{bmatrix} \qquad \text{Eq.28}$$

$$\emptyset_{\text{rot.Y}} = \text{atan2}\left(\frac{-r_{31}}{\sqrt{{r_{11}}^2 + {r_{21}}^2}}\right) \qquad \text{Eq.29}$$

$$\omega_{\text{rot.Z}} = \text{atan2}\left(\frac{r_{21}}{r_{11}}\right) \qquad \text{Eq.30}$$

$$\gamma_{\text{rot.X}} = \text{atan2}\left(\frac{r_{32}}{r_{33}}\right) \qquad \text{Eq.31}$$

Os termos q_1, q_2 , q_3 , q_4 são termos independentes que satisfazem a equação de orientação de um ponto localizado em uma hiper esfera de raio unitário de quatro dimensões e por isso satisfazem a Equação 32 (CRAIG, 2005, p.43).

$${q_1}^2 + {q_2}^2 + {q_3}^2 + {q_4}^2 = 1 \qquad \text{Eq.32}$$

A inversa de um Quatérnio Dual (QD) é dada pela Equação 33:

$$\underline{\boldsymbol{Q}}^{-1} = \frac{\underline{\boldsymbol{Q}}^{*}}{\left\|\underline{\boldsymbol{Q}}\right\|^{2}} \qquad \text{Eq.33}$$

$$\underline{\boldsymbol{Q}}^{-1} = \begin{bmatrix} \left(\frac{q_1}{s}\right) \\ \left(\frac{-q_2}{s}\right) \\ \left(\frac{-q_3}{s}\right) \\ \left(\frac{-q_4}{s}\right) \\ \left(\frac{q_5}{s} + \frac{-q_1 * 2\,\mathbf{v}}{(s)^2}\right) \\ \left(\frac{-q_6}{s} + \frac{q_2 * 2\,\mathbf{v}}{(s)^2}\right) \\ \left(\frac{-q_7}{s} + \frac{q_3 * 2\,\mathbf{v}}{(s)^2}\right) \\ \left(\frac{-q_8}{s} + \frac{q_4 * 2\,\mathbf{v}}{(s)^2}\right) \end{bmatrix}$$

$$s = (q_1{}^2 + q_2{}^2 + q_3{}^2 + q_4{}^2)$$

$$\mathbf{v} = (q_1 q_5 + q_2 q_6 + q_3 q_7 + q_4 q_8)$$

Quando se trata de Quatérnio Dual Unitário (QDU), os valores de s e v são definidos como $s = 1$ e $\mathrm{v} = 0$. Essas duas condições ou contrações apresentadas para as variáveis s e v é que permitem que o Quatérnio Dual possa representar qualquer corpo rígido no espaço 3D usando os 6 GDL equivalentes a qualquer objeto descrito em três dimensões, ou seja, as condições para s e v tornam a reprodução com 8 GDL dos Quatérnios Duais uma representação com 6 GDL. Assim, a equação anterior pode ser reescrita para a inversa de QDUs como a Equação 34 (GOUASMI, OUALI e BRAHIM,2012, p.15).

$$\underline{\boldsymbol{Q}}^{-1} = \begin{bmatrix} q_1 \\ -q_2 \\ -q_3 \\ -q_4 \\ q_5 \\ -q_6 \\ -q_7 \\ -q_8 \end{bmatrix} \qquad \text{Eq.34}$$

2.3 Análise de desempenho computacional

Os computadores armazenam os números reais como aproximações numéricas denominadas números ponto-flutuantes. Em algumas literaturas, o termo ‘flops’ é utilizado como uma medida de velocidade do processador e significa “operações ponto-flutuantes por segundo”. Neste trabalho, o termo ‘flop’ é concebido como uma unidade de contagem.

Dados dois algoritmos diferentes para resolver o mesmo problema, deseja-se

escolher qual desses algoritmos é o melhor, para isso deve-se pensar em termos de eficiência (ou custo computacional), ou seja, aquele algoritmo que consumir menos recursos para realizar uma mesma tarefa é o mais eficiente. Em geral, a análise do custo computacional é feita de duas formas: em termos de tempo e de espaço de memória ocupado. Quando se trata da eficiência espacial ou eficiência da memória, deseja-se a informação de quanta memória está sendo utilizada pelo algoritmo para armazenar os dados, sejam matrizes, vetores ou escalares e, também, informação em relação à hierarquia das memórias acessadas (registros, cache, memória principal, armazenamento em disco). Quando se trata de eficiência temporal, a intenção é saber quanto tempo um algoritmo leva para realizar determinada tarefa. É razoável de se pensar que o tempo vai ser proporcional ao número de operações de ponto flutuante (flop) feito pelo algoritmo, observa-se que o tempo total não depende apenas disso, mas também de outros fatores como memória, taxas de transferências de dados da memória para o CPU, etc. A contagem do número de operações ponto flutuante (flop) para realizar determinada tarefa é uma maneira de verificar a eficiência computacional. É importante salientar que a eficiência temporal e a eficiência espacial não são independentes, uma leva à outra já que a habilidade de um algoritmo acessar uma memória influencia diretamente no tempo de execução do algoritmo (UEBERHUBER, 1995, p.227).

Na área de análise de algoritmos, existem dois tipos de tratamentos bem distintos, o primeiro é a análise de um único algoritmo isoladamente e o segundo é a análise e posterior comparação de dois ou mais algoritmos. Quando a análise considera um algoritmo em particular verifica-se qual o custo de usar o algoritmo para resolver um problema específico, neste caso, algumas características importantes do algoritmo em questão são investigadas, geralmente uma análise do número de vezes que cada parte do algoritmo deve ser executada, seguida de um estudo da quantidade de memória necessária em cada instrução do algoritmo. Por outro lado, quando a análise é de uma classe de algoritmos e deseja-se o algoritmo de menor custo possível para resolver um problema, toda uma família de algoritmos é investigada com o objetivo de identificar um que seja o melhor possível. Desta forma, a medida do custo computacional será um único modelo matemático aplicado a diferentes algoritmos e então é possível confrontá-los e escolher o mais adequado para resolver o problema em questão. Isto significa colocar limites para a complexidade computacional dos algoritmos pertencentes à mesma classe, estabelece-se o conjunto de operações a serem executadas, assim como o custo associado com a execução de cada operação. É usual ignorar o custo de algumas das operações envolvidas e considerar apenas as operações mais significativas. Por exemplo, em algoritmos de comparações para obter o maior e menor elemento de um conjunto de n números ($n \geq 1$), o mais importante neste exemplo é estimar o número mínimo de comparações necessárias, para encontrar o maior elemento de um conjunto de n números, pode-se pensar que um algoritmo como este fará pelo menos $(n-1)$ comparações entre os elementos. É comum, em exemplos como este, considerar apenas as operações mais importantes (contagem do número de comparações) e não contabilizar o custo de outras operações presentes no algoritmo (ZIVIANI, 1999, p.3).

2.3.1 Multiplicações, Adições e Funções trigonométricas

O esforço em descrever formalmente o produto (composição) de dois mapeamentos lineares de dimensão dim dados por $L_1, L_2: \mathbb{R}^{dim} \rightarrow \mathbb{R}^{dim}$ é que levou à definição natural para multiplicação de matrizes onde as linhas e colunas dos algoritmos requeriam (dim^3) multiplicações e $(dim^2 * (dim-1))$ adições. Uma matriz 3x3, por exemplo, quando multiplicada por outra matriz de mesma dimensão conterá 3^3 multiplicações e $3^2 * (3-1)$ adições. Deste modo, a multiplicação de matrizes possui complexidade dita assintótica uma vez que as operações aritméticas são de ordem cúbica (dim^3) tanto para as adições quanto para as multiplicações (UEBERHUBER, 1995, p.193).

Feng e Wan (2013) declararam que os modelos usando quatérnio dual são mais precisos, computacionalmente mais eficientes, robustos e um método flexível de representar as transformações rígidas. De acordo com os autores, os QDUs permitem a criação de programas mais sofisticados e mais claros computacionalmente, além disso, são mais fáceis de trabalhar e controlar, quando implementados módulos pré-programados de quatérnios duais que estejam incluídas operações de multiplicação e normalização. O custo computacional de utilizar as matrizes de transformação homogênea ou quatérnios duais é apresentado pelos autores como a contagem das operações matemáticas.

Adorno (2011, p.41-43) afirma que o custo de cálculo de transformações rígidas usando matrizes de transformação homogênea é maior que o custo de utilizar quatérnios duais, dos resultados das análises deste autor decorrem-se as Equações 35 e 36. O custo de cálculo da multiplicação de m MTHs (Equação 35) leva em conta o custo individual do cálculo de cada MTH e também multiplicações entre elas.

$$custo_{TOTAL}\left(\boldsymbol{H}_{DH}{}_{1}^{0} * \boldsymbol{H}_{DH}{}_{2}^{1} \cdots \boldsymbol{H}_{DH}{}_{m}^{m-1}\right) = \{func.trig., mult., adições\}$$
$$m * custo\left(\boldsymbol{H}_{DH}{}_{i}^{i-1}\right) + (m-1) * custo(multiplic.entre\ duas\ MTH) =$$
$$m * \{4, 6, 0\} + (m-1) * \{0, 64, 48\} =$$
$$custo_{TOTAL}\left(\boldsymbol{H}_{DH}{}_{m}^{0}\right) = \{4m, 70m - 64, 48m - 48\} \quad \text{Eq.35}$$

Em comparação às MTHs, Adorno (2011, p.41-43) também revela que o custo de cálculo da multiplicação de m QDUs será dado pela Equação 36 que leva em conta o custo individual do cálculo de cada QDU e também a multiplicações entre eles.

$$custo_{TOTAL}\left(\underline{\boldsymbol{Q}}_{DH}{}_{1}^{0} \blacksquare \underline{\boldsymbol{Q}}_{DH}{}_{2}^{1} \cdots \underline{\boldsymbol{Q}}_{DH}{}_{m}^{m-1}\right) = \{func.trig., mult., adições\}$$
$$m * custo\left(\underline{\boldsymbol{Q}}_{DH}{}_{i}^{i-1}\right) + (m-1) * custo(multiplic.entre\ dois\ QD) =$$
$$n * \{4, 12, 4\} + (m-1) * \{0, 48, 40\} =$$
$$custo_{TOTAL}\left(\underline{\boldsymbol{Q}}_{DH}{}_{m}^{0}\right) = \{4m, 60m - 48, 44m - 40\} \quad \text{Eq.36}$$

As Tabelas 2, 3, 4 e 5 mostram a análise individual para se chegar aos resultados das duas equações anteriores (Eq. 35 e Eq.36) para custo de cálculo da multiplicação de m MTHs e m QDUs. Nas Tabelas 2 e 4 são evidenciadas as operações matemáticas para uma única MTH e um único QDU, respectivamente. Na Tabela 3 é mostrada as operações para a multiplicação de duas matrizes 4x4 e na Tabela 5 as operações devido a multiplicação de dois quatérnios duais. Nota-se que as operações matemáticas repetidas são contabilizadas uma única vez.

Tabela 2: Custo do cálculo de uma única MTH

$$\boldsymbol{H}_{DH}{}_{i}^{i-1} = \begin{bmatrix} \boxed{\cos\theta_i} & -\operatorname{sen}\theta_i \star \cos\alpha_i & \operatorname{sen}\theta_i \star \operatorname{sen}\alpha_i & a_i \star \cos\theta_i \\ \boxed{\operatorname{sen}\theta_i} & \cos\theta_i \star \cos\alpha_i & -\cos\theta_i \star \operatorname{sen}\alpha_i & a_i \star \operatorname{sen}\theta_i \\ 0 & \boxed{\operatorname{sen}\alpha_i} & \boxed{\cos\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Funções trigonométricas ▭	Multiplicações ★	Somas/Subtrações ✚ ━
4	6	0

Fonte: Autoria Própria

Tabela 3: Custo de cálculo da multiplicação de duas MTH

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} \end{bmatrix} \begin{bmatrix} b_{1,1} & b_{1,2} & b_{1,3} & b_{1,4} \\ b_{2,1} & b_{2,2} & b_{2,3} & b_{2,4} \\ b_{3,1} & b_{3,2} & b_{3,3} & b_{3,4} \\ b_{4,1} & b_{4,2} & b_{4,3} & b_{4,4} \end{bmatrix} = \begin{bmatrix} c_{1,1} & c_{1,2} & c_{1,3} & c_{1,4} \\ c_{2,1} & c_{2,2} & c_{2,3} & c_{2,4} \\ c_{3,1} & c_{3,2} & c_{3,3} & c_{3,4} \\ c_{4,1} & c_{4,2} & c_{4,3} & c_{4,4} \end{bmatrix}$$

$$c_{1,1} = a_{1,1} \star b_{1,1} + a_{1,2} \star b_{2,1} + a_{1,3} \star b_{3,1} + a_{1,4} \star b_{4,1}$$
$$c_{1,2} = a_{1,1} \star b_{1,2} + a_{1,2} \star b_{2,2} + a_{1,3} \star b_{3,2} + a_{1,4} \star b_{4,2}$$
$$c_{1,3} = a_{1,1} \star b_{1,3} + a_{1,2} \star b_{2,3} + a_{1,3} \star b_{3,3} + a_{1,4} \star b_{4,3}$$
$$c_{1,4} = a_{1,1} \star b_{1,4} + a_{1,2} \star b_{2,4} + a_{1,3} \star b_{3,4} + a_{1,4} \star b_{4,4}$$

linha 1

$$c_{2,1} = a_{2,1} \star b_{1,1} + a_{2,2} \star b_{2,1} + a_{2,3} \star b_{3,1} + a_{2,4} \star b_{4,1}$$
$$c_{2,2} = a_{2,1} \star b_{1,2} + a_{2,2} \star b_{2,2} + a_{2,3} \star b_{3,2} + a_{2,4} \star b_{4,2}$$
$$c_{2,3} = a_{2,1} \star b_{1,3} + a_{2,2} \star b_{2,3} + a_{2,3} \star b_{3,3} + a_{2,4} \star b_{4,3}$$
$$c_{2,4} = a_{2,1} \star b_{1,4} + a_{2,2} \star b_{2,4} + a_{2,3} \star b_{3,4} + a_{2,4} \star b_{4,4}$$

linha 2

$$c_{3,1} = a_{3,1} \star b_{1,1} + a_{3,2} \star b_{2,1} + a_{3,3} \star b_{3,1} + a_{3,4} \star b_{4,1}$$
$$c_{3,2} = a_{3,1} \star b_{1,2} + a_{3,2} \star b_{2,2} + a_{3,3} \star b_{3,2} + a_{3,4} \star b_{4,2}$$
$$c_{3,3} = a_{3,1} \star b_{1,3} + a_{3,2} \star b_{2,3} + a_{3,3} \star b_{3,3} + a_{3,4} \star b_{4,3}$$
$$c_{3,4} = a_{3,1} \star b_{1,4} + a_{3,2} \star b_{2,4} + a_{3,3} \star b_{3,4} + a_{3,4} \star b_{4,4}$$

linha 3

$$c_{4,1} = a_{4,1} \star b_{1,1} + a_{4,2} \star b_{2,1} + a_{4,3} \star b_{3,1} + a_{4,4} \star b_{4,1}$$
$$c_{4,2} = a_{4,1} \star b_{1,2} + a_{4,2} \star b_{2,2} + a_{4,3} \star b_{3,2} + a_{4,4} \star b_{4,2}$$
$$c_{4,3} = a_{4,1} \star b_{1,3} + a_{4,2} \star b_{2,3} + a_{4,3} \star b_{3,3} + a_{4,4} \star b_{4,3}$$
$$c_{4,4} = a_{4,1} \star b_{1,4} + a_{4,2} \star b_{2,4} + a_{4,3} \star b_{3,4} + a_{4,4} \star b_{4,4}$$

linha 4

Funções trigonométricas ▭	Multiplicações ★	Somas/Subtrações ✚ ━
0	64	48

Fonte: Autoria Própria

Tabela 4: Custo do cálculo de um único QDU

$$\underline{\boldsymbol{Q}}_{DH}{}_{i}^{i-1} = \begin{bmatrix} \boxed{\cos(\theta_i/2)} \star \boxed{\cos(\alpha_i/2)} \\ \cos(\theta_i/2) \star \operatorname{sen}(\alpha_i/2) \\ \boxed{\operatorname{sen}(\theta_i/2)} \star \boxed{\operatorname{sen}(\alpha_i/2)} \\ \operatorname{sen}(\theta_i/2) \star \cos(\alpha_i/2) \\ -(\mathbf{a}_i/2) \star \cos(\theta_i/2) \operatorname{sen}(\alpha_i/2) - (d_i/2) \star \operatorname{sen}(\theta_i/2) \cos(\alpha_i/2) \\ +(\mathbf{a}_i/2) \star \cos(\theta_i/2) \cos(\alpha_i/2) - (d_i/2) \star \operatorname{sen}(\theta_i/2) \operatorname{sen}(\alpha_i/2) \\ +(\mathbf{a}_i/2) \star \operatorname{sen}(\theta_i/2) \cos(\alpha_i/2) + (d_i/2) \star \cos(\theta_i/2) \operatorname{sen}(\alpha_i/2) \\ -(\mathbf{a}_i/2) \star \operatorname{sen}(\theta_i/2) \operatorname{sen}(\alpha_i/2) + (d_i/2) \star \cos(\theta_i/2) \cos(\alpha_i/2) \end{bmatrix}$$

Funções trigonométricas ▭	Multiplicações ★	Somas/Subtrações ✚ ━
4	12	4

Fonte: Autoria Própria

Tabela 5: Custo de cálculo da multiplicação de dois QDU

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \\ a_8 \end{bmatrix} \blacksquare \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \\ b_8 \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \\ c_8 \end{bmatrix}$$

$$c_1 = a_1 \star b_1 - a_2 \star b_2 - a_3 \star b_3 - a_4 \star b_4$$

$$c_2 = a_1 \star b_2 + a_2 \star b_1 + a_3 \star b_4 - a_4 \star b_3$$

$$c_3 = a_1 \star b_3 - a_2 \star b_4 + a_3 \star b_1 + a_4 \star b_2$$

$$c_4 = a_1 \star b_4 + a_2 \star b_3 - a_3 \star b_2 + a_4 \star b_1$$

$$c_5 = a_1 \star b_5 - a_2 \star b_6 - a_3 \star b_7 - a_4 \star b_8 + a_5 \star b_1 - a_6 \star b_2 - a_7 \star b_3 - a_8 \star b_4$$

$$c_6 = a_1 \star b_6 + a_2 \star b_5 + a_3 \star b_8 - a_4 \star b_7 + a_5 \star b_2 + a_6 \star b_1 + a_7 \star b_4 - a_8 \star b_3$$

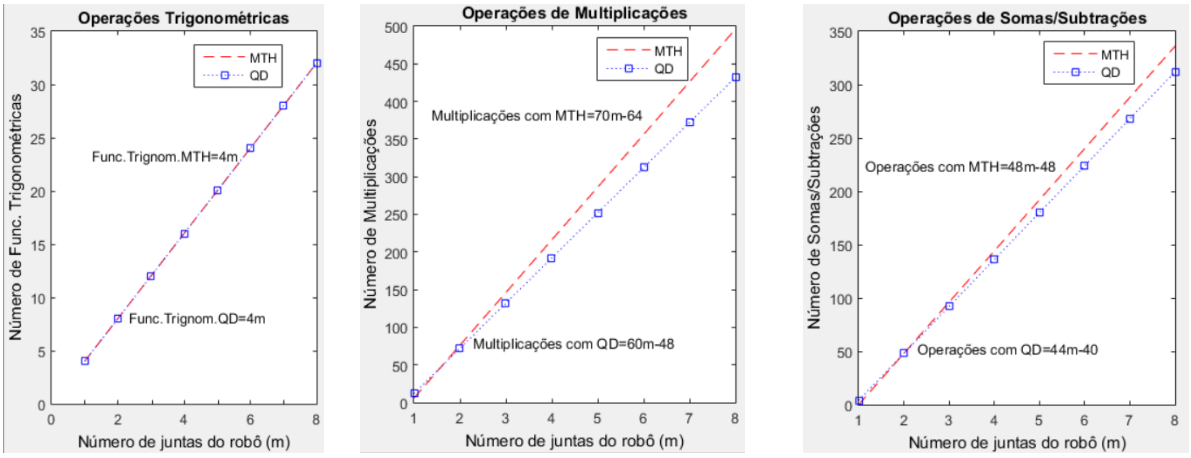
$$c_7 = a_1 \star b_7 - a_2 \star b_8 + a_3 \star b_5 + a_4 \star b_6 + a_5 \star b_3 - a_6 \star b_4 + a_7 \star b_1 + a_8 \star b_2$$

$$c_8 = a_1 \star b_8 + a_2 \star b_7 - a_3 \star b_6 + a_4 \star b_5 + a_5 \star b_4 + a_6 \star b_3 - a_7 \star b_2 + a_8 \star b_1$$

Funções trigonométricas ▭	Multiplicações ★	Somas/Subtrações ✚ ━
0	48	40

Fonte: Autoria Própria

A comparação entre as equações Eq.35 e Eq.36 foi mostrada no Gráfico 1, nele é evidenciado que à medida que o número de graus de liberdade do robô cresce (m cresce) os Quatérnios Duais executarão menos operações (multiplicação e somas ou subtrações) do que as Matrizes de Transformação Homogêneas (Gráfico 1- (b) e (c)). O número de operações Trigonométricas executado cresce obedecendo a mesma constante de linearidade independente do uso de QDUs ou MTHs (Gráfico 1- (a)).



(a)Função Trigonom. (b) Multiplicações (c) Somas e/ou Subtrações

Gráfico 1: Comparação performance computacional para manipuladores com m juntas usando MTHs e QDs. Fonte: Autoria Própria

2.3.2 Número de Flop

Anton (2012, p.501) afirma que o custo computacional de resolver um sistema linear é, muitas vezes, determinado pelo tempo que um computador leva para executar os cálculos. Em geral, o tempo de computação depende de dois fatores: a velocidade do processador e o número de operações exigidas pelo algoritmo. Assim, a escolha do algoritmo correto tem implicações financeiras importantes num contexto industrial ou de pesquisa. Computacionalmente falando, uma operação aritmética (soma, subtração e multiplicação) entre dois números reais é contabilizada como flop, que é um acrônimo para "operação ponto-flutuante" (em inglês *Floating Point Operation*). O número total de flop necessários para resolver um problema é denominado o custo da solução, este fornece uma maneira conveniente de escolher entre vários algoritmos para resolver o problema. Se a velocidade do processador do computador e os aspectos financeiros de suas operações forem conhecidos, podemos converter, quando necessário, o custo em flop para unidades de tempo ou dinheiro se pensado em escala industrial. Por exemplo, muitos dos computadores pessoais atualmente são capazes de executar cerca de 10 gigaflop por segundo (1 gigaflop $= 10^9$ flop). Assim, neste caso um algoritmo que custa 1.000.000 flop seria executado em 0,0001 segundos.

Addison(1993, p.8) realizou uma comparação (do termo em inglês 'benchmark') para mensurar a distribuição de memória em algoritmos, a forma fundamental para avaliar a performace foi o tempo para completar uma tarefa específica. Em sua definição de tempo, o autor afirma ter usado em seus experimentos uma medição de tempo externa (do inglês 'external clock') para contabilizar o tempo decorrido em cada operação, mas outros meios de contabilizar o tempo, segundo o autor, poderiam ser cuidadosamente analisados se fosse possível, claro, visualizar o tempo dedicado para uma tarefa específica. A dificuldade para o autor, está no fato de que o tempo de CPU, por exemplo, quando usado para medir o tempo de operação poderia variar dependendo do sistema operacional levando a uma comparação injusta dependendo do processador de cada computador. A quantidade aritmética para avaliar a performace é dada em FLOP ou MFLOP (10^6 FLOP), essa é uma unidade de contagem de tarefas executadas. É importante não confundir com 'Flop/s' ou 'MFlop/s' que é usado para medir a taxa de operações por segundo. Para evitar confusão, o termo 'flop' não deve receber plural. De maneira simplificada, os custos de execução das operações mais comuns são dados pela Tabela 6.

Tabela 6: Comparação das operações mais comuns

Operação	Peso da Operação Ponto Flutuante
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)

2.3.3 Comandos de Interrupção em Algoritmos

Rafiquzzaman (2014, p.295) explica que normalmente, há dois tipos de interrupções: as interrupções externas e as interrupções internas. A diferença básica entre os dois tipos refere-se à origem destas, as interrupções externas são iniciadas através dos pinos de interrupção (INT) de um microcontrolador(MCU) utilizando dispositivos externos ligados à ele, nestas os sinais lógicos dos pinos são monitorarados a fim de verificar mudanças no nível lógico do circuito externo ligado ao MCU (por exemplo, um botão RESET quando pressionado ou um sensor de incêndio quando detecta o início de alguma chama), já as interrupções internas são aquelas que são ativadas por periféricos contidos no próprio MCU ou ativadas quando certas situações previamente estabelecidas nas linhas de programação forem atingidas. Interrupções internas são desencadeadas intrinsecamente por condições como, por exemplo, a conclusão da conversão A/D (analógica para digital), intercepção de um temporizador, interferência de um comparador, acionamento de portas como a serial e/ou paralela e ainda devido a erros constatados no hardware.

Uma interrupção, de acordo com Candy (2010, p.222), é um evento assíncrono, imprevisível e que pode ocorrer múltiplas vezes antes, durante ou após um ciclo de instrução de um programa, requerendo portanto imediata atenção. Na entrada e saída de uma interrupção, um dispositivo externo ou uma condição interna pode forçar a CPU a interromper a execução do programa principal temporariamente para que possa executar um outro programa conhecido como rotina de interrupção do serviço (*ISR- "Interrupt Service Routine"*), o programador escreve o comando da rotina de serviço de interrupção em outro endereço. Essa rotina satisfaz as necessidades do dispositivo externo ou da condição interna. Uma vez que a interrupção é reconhecida, o microcontrolador salva internamente o endereço de retorno e o conteúdo de alguns outros registros e ramifica-se automaticamente para um endereço pré-definido pelo fabricante. O microcontrolador geralmente deixa incompleta a instrução atual e com o endereço de retorno e conteúdo de certos registros internos, depois de completar a rotina de interrupção, retorna à instrução posterior para executar o controle no ponto de parada do programa principal. A interrupção é muito semelhante às sub-rotinas ou funções encontradas linguagem C. A Figura 7 mostra de maneira esquemática como ocorre o mecanismo de interrupção durante a execução do programa principal.

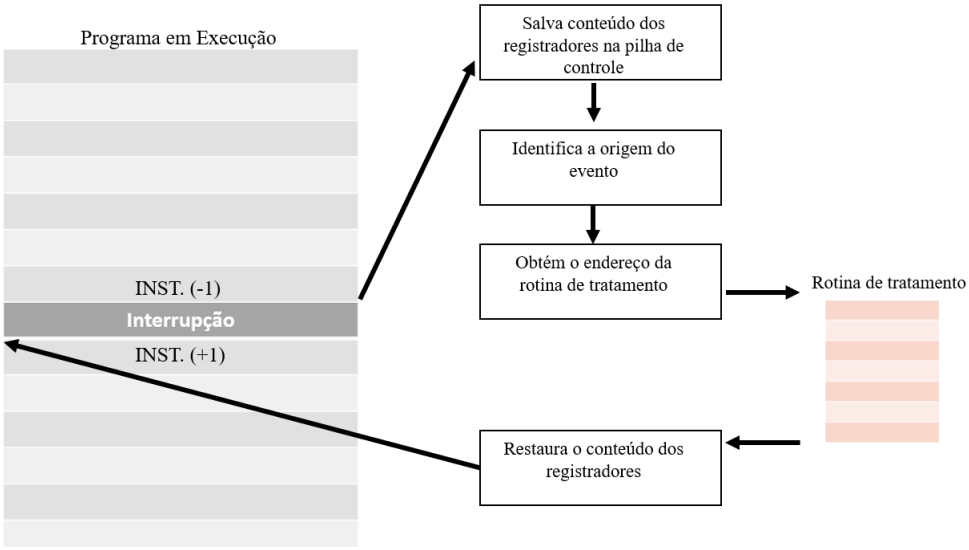


Figura 7: Mecanismo de interrupção
Fonte: Autoria Própria

Segundo Bates (2008) a maneira mais eficaz de integrar as operações cronometradas dentro de um programa é usando uma interrupção de tempo (do inglês *"timer interrupt").* Uma rotina de interrupção pode ser escrita e atribuída à interrupção do tempo. O tempo é configurado durante o início do programa principal e o contador prossegue simultaneamente com a execução do programa, até que ocorra um tempo limite e a interrupção seja gerada. O programa principal então é suspenso e a rotina de interrupção é executada. Quando terminada a interrupção, o programa principal é retomado no ponto original. Se a rotina de interrupção contiver uma instrução para alternar um bit de saída, uma onda quadrada pode ser obtida com um período de duas vezes o atraso do temporizador. Quando as interrupções são usadas por programas em linguagem Assembly, é mais fácil prever o efeito delas, porque o programador tem controle direto sobre a sequência exata da rotina de interrupção. Por outro lado, em um programa em linguagem C, como a interrupção é gerada automaticamente pelo compilador, neste caso o tempo exato que resulta de uma interrupção é menos óbvio. Por esse motivo, o uso de um sistema operacional em tempo real (sigla RTOS do inglês - "*Real-Time Operating System"*) é por vezes preferido no ambiente C, especialmente quando os programas se tornam mais complexos. De fato, a linguagem C foi originalmente desenvolvida precisamente para este fim, para escrever em alto nível sistemas operacionais para computadores.

Crisp (2004) afirma que um timer ou contador é uma série de flip-flops que mudam de estado para cada pulso de entrada, assim, cada um desses circuitos dividiria a freqüência de entrada por um fator de dois (1 /2). Se este sinal é então alimentado para o próximo flip-flop, então a saída é 1 / 4 da frequência original. O próximo circuito teria uma saída de 1/8, depois 1/16 e assim por diante. Existem dois timers, T0 e T1. Estes podem ser programados para dividir 256, 8192 ou 65536 e irão gerar um sinal de interrupção até o final de modo que pode ser detectado pelo software. Um dos modos permite que o timer / contador de 8 bits (divisão por $2^8 = 256$) recarregue e comece a contar novamente a cada vez continuamente. O sinal de entrada que está sendo contado pode se originar de um circuito externo então ele conta o número de pulsos de entrada, ou pode usar um sinal interno que é na verdade 1/12 da freqüência de clock em uso. Como mencionado acima, isto gera um sinal de interrupção quando atinge seu valor máximo. Podemos pré-carregar o temporizador com um número para começar contando a partir dele. Isso permitirá que a interrupção seja gerada após qualquer número necessário de eventos ou intervalo de tempo.

2.3.4 Análise de Complexidade de Algoritmos – Modelo RAM

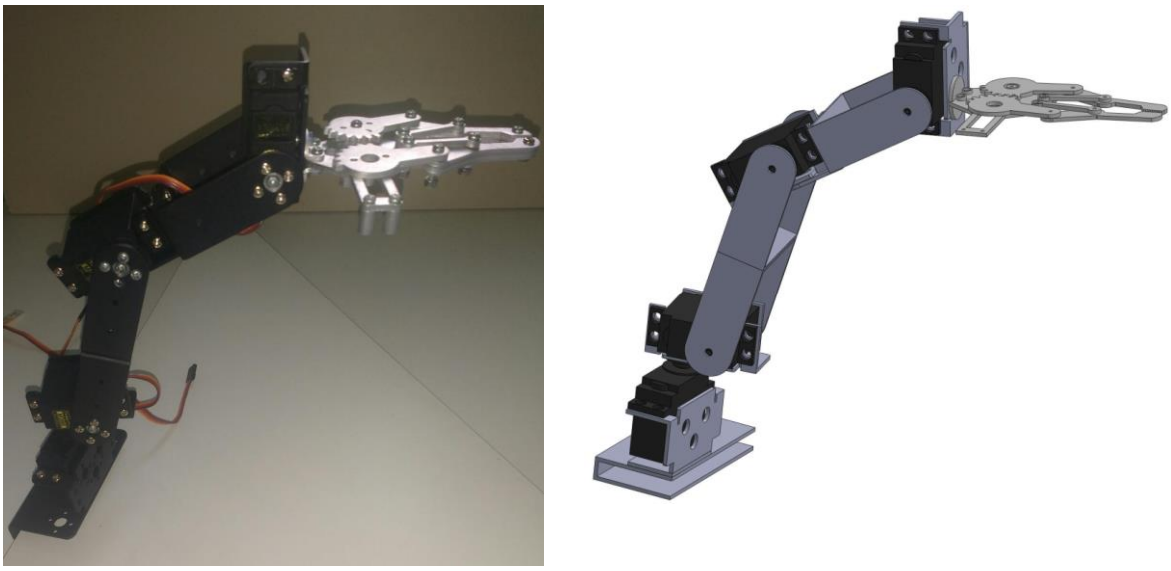
O modelo de Máquina de Acesso Aleatório (sigla RAM do inglês "Random Access Machine") caracteriza o tempo de execução como uma função do tamanho da entrada (n) no algoritmo. Muito cuidado para não confundir este modelo com outro termo da computação relacionado à memória RAM (sigla RAM também usada para se referir ao acesso aleatório à memória do inglês "Random Access Memory"). O modelo RAM em questão possui uma abordagem teórica que analisa uma descrição de alto nível do algoritmo, trata-se de um modelo de computação genérico que contém instruções encontradas em algoritmos reais cujos tempos são considerados constantes na maioria dos computadores, tais como instruções aritméticas (soma, subtração, resto, piso, etc), de movimentação de dados (carregar, armazenar, copiar), de controle (desvio condicional, chamada e retorno de funções). O modelo mede a eficiência de execução contando o número de passos de acordo com o tamanho da entrada no programa. O modelo parte do princípio de que independente da máquina usada (uso de um computador hipotético) é possível contar os passos ('step') de acesso aleatório ou RAM de acordo com a as instruções em cada linha do algoritmo, obedecendo as duas descrições à seguir (SKIENA, 2008, p. 31):

- Cada acesso à memória leva exatamente um pulso de "clock" ou passo, ou seja, cada operação simples no algoritmo (+, *, -, =, if, call) é contabilizada uma única vez.
- Loops (comandos: for, while) e sub-rotinas não são considerados como operações simples. Em vez disso, eles são contabilizados como uma composição de muitas operações. Não faria sentido classificá-los como uma operação de etapa única, já que executar um loop 1.000.000 vezes certamente demoraria muito mais do que executá-lo 10 vezes. O tempo que leva para percorrer um "loop" ou executar um subprograma depende do número (n) de iterações do loop ou da natureza específica da sub-rotina.

Uma alternativa para a escolha do melhor algoritmo para solução de um mesmo problema seria o modelo RAM aplicável quando, por exemplo, vários algoritmos distintos possuem a mesma eficiência temporal e se deseja escolher um deles, Gonnet e Baeza-Yates (1991, p.7) apresentam argumentos a favor deste modelo como uma alternativa para análise tanto para a eficiência de execução em algumas situações isoladas (quando não se pretende fazer comparações, somente análise simples de um único algoritmo) quanto para situações onde haja necessidade de uma escolha entre vários algoritmos diferentes usados para resolver um mesmo tipo de problema e todos com um custo temporal dentro de uma mesma ordem de grandeza. Para o caso da necessidade de solucionar um impasse entre algoritmos, os custos reais das operações são todos considerados, assim os custos não aparentes tais como alocação de memória, indexação, carga, etc, não deixam de ser contabilizados. Diversos outros autores discutiram o Modelo RAM em questão (Anexo B) e sua aplicação foi demonstrada com exemplos nos Apêndices D.

3 METODOLOGIA

Foi construído um protótipo com 4 Graus de Liberdade e um desenho foi feito em escala real com três dimensões no software SolidWork para melhor acompanhar o movimento e espaço de trabalho do manipulador. Além disso, parece mais adequado para testes um robô menor, mais leve, de fácil montagem, simplicidade para manejar e realizar medições dos movimentos, além da segurança do operador durante os comandos. No protótipo, cada vértice corresponde a uma articulação representadas por meio de um coeficiente. A Figura 8 mostra um comparativo entre o manipulador e o desenho em escala natural. A Figura 9 mostra o robô e suas respectivas dimensões representadas no software SolidWorks.



(a) Fotografia do Robô (b) Desenho em escala natural 1:1

Figura 8: Comparativo Robô 4GDL e Desenho em SolidWorks
Fonte: Autoria Própria

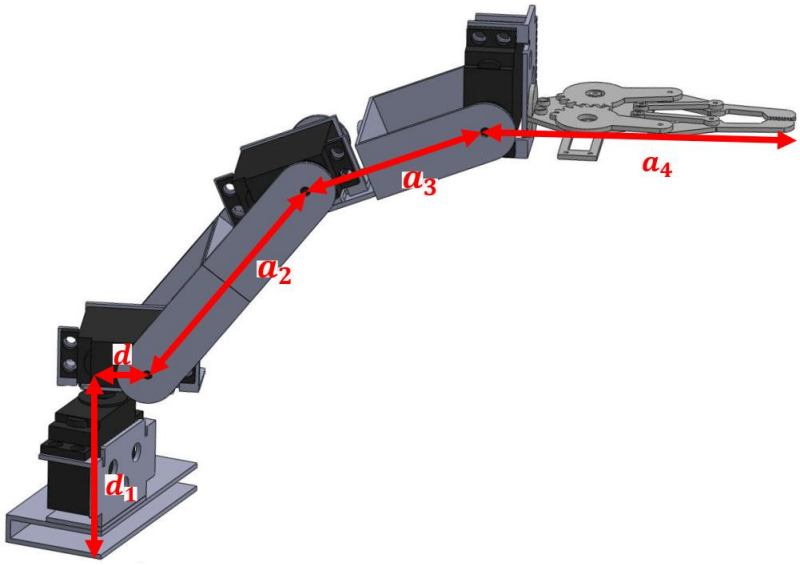
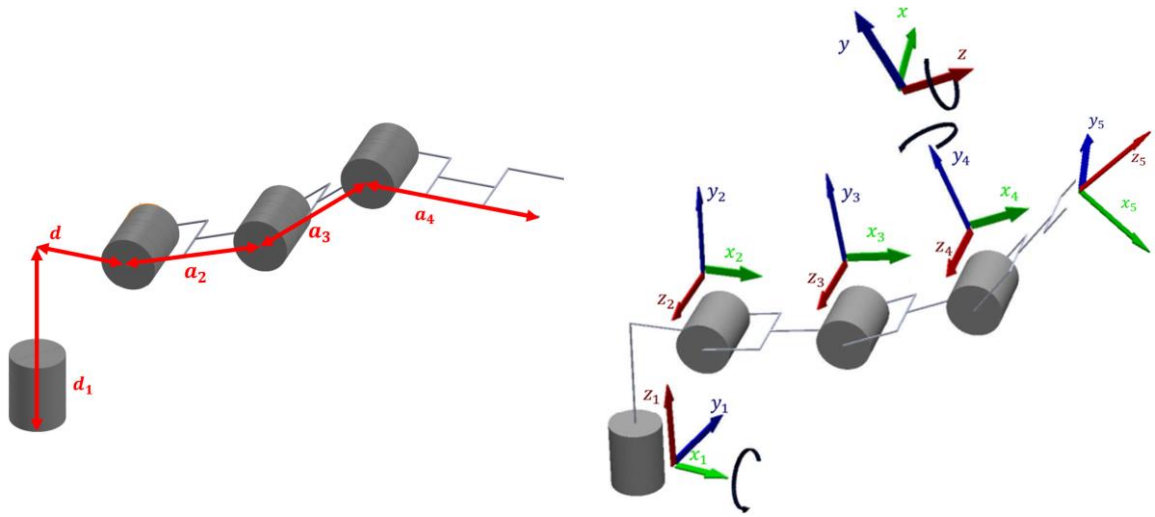


Figura 9: Desenho em SolidWorks do manipulador
Fonte: Autoria Própria

A representação esquemática (Figura 10) foi apresentada como a melhor forma de visualizar os parâmetros Denavit-Hartenberg contidos na Tabela 7.



(a) Valores dos elos (b) Coordenadas Denavit Hartenberg

Figura 10: Desenhos Esquemáticos do Manipulador
Fonte: Autoria Própria

Tabela 7: Valores Denavit-Hartenberg do manipulador

Theta	d	a	α
θ_1^*	$d_1 = 8{,}40$	$\mathbf{d} = 1{,}13$	90°
θ_2^*	0	$a_2 = 9{,}81$	0°
θ_3^*	0	$a_3 = 7{,}18$	0°
θ_4^*	0	$a_4 = 12{,}29$	0°

$\theta_1, \theta_2, \theta_3, \theta_4$ são variáveis das juntas rotativas

Os cálculos das cinemáticas direta e inversa foram realizados para o manipulador de três maneiras distintas: por geometria, por matrizes de transformação homogênea e por meio dos quatérnios duais. As equações cinemáticas obtidas geometricamente constam nos apêndices B1 e C para cinemática direta e inversa, respectivamente. Os cálculos utilizando Matrizes de Transformação Homogênea e Quatérnios Duais serão apresentados na sequência.

3.1 Cálculo da Cinemática Direta do manipulador

3.1.1 Cinemática Direta com Matriz de transformação Homogênea (MTH)

Utilizando a Equação 4 para matrizes de transformação homogênea e a Tabela 7 Denavit-Hartenberg do manipulador, tem-se resultados para as transformações homogêneas (Equações 37, 38, 39 e 40):

$$\boldsymbol{H}_{1}^{0} = \begin{bmatrix} \cos\theta_1 & -\operatorname{sen}\theta_1 * cos(90°) & \operatorname{sen}\theta_1 * sen(90°) & \mathbf{d}\cos\theta_1 \\ \operatorname{sen}\theta_1 & \cos\theta_1 * cos(90°) & -\cos\theta_1 * sen(90°) & \mathbf{d}\operatorname{sen}\theta_1 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{cases}\cos(90°) = 0 \\ \operatorname{sen}(90°) = 1\end{cases}$$

$$\boldsymbol{H}\,{}_1^0 = \begin{bmatrix}\cos\theta_1 & 0 & \operatorname{sen}\theta_1 & \mathbf{d}\cos\theta_1 \\ \operatorname{sen}\theta_1 & 0 & -\cos\theta_1 & \mathbf{d}\operatorname{sen}\theta_1 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1\end{bmatrix} \quad \text{Eq.37}$$

$$\boldsymbol{H}\,{}_2^1 = \begin{bmatrix}\cos\theta_2 & -\operatorname{sen}\theta_2 & 0 & a_2 * \cos\theta_2 \\ \operatorname{sen}\theta_2 & \cos\theta_2 & 0 & a_2 * \operatorname{sen}\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1\end{bmatrix} \quad \text{Eq.38}$$

$$\boldsymbol{H}\,{}_3^2 = \begin{bmatrix}\cos\theta_3 & -\operatorname{sen}\theta_3 & 0 & a_3 * \cos\theta_3 \\ \operatorname{sen}\theta_3 & \cos\theta_3 & 0 & a_3 * \operatorname{sen}\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1\end{bmatrix} \quad \text{Eq.39}$$

$$\boldsymbol{H}\,{}_4^3 = \begin{bmatrix}\cos\theta_4 & -\operatorname{sen}\theta_4 & 0 & a_4 * \cos\theta_4 \\ \operatorname{sen}\theta_4 & \cos\theta_4 & 0 & a_4 * \operatorname{sen}\theta_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1\end{bmatrix} \quad \text{Eq.40}$$

Efetuando a multiplicação das MTHs, tem-se as composições de transformações dadas pelas Equações 41, 42 e 43:

$$\boldsymbol{H}\,{}_2^0 = \boldsymbol{H}\,{}_1^0 * \boldsymbol{H}\,{}_2^1 =$$

$$= \begin{bmatrix}(C_1 * C_2) & (-C_1 * S_2) & (S_1) & (a_2 * C_1 * C_2 + d * C_1) \\ (S_1 * C_2) & (-S_1 * S_2) & (-C_1) & (a_2 * S_1 * C_2 + d * S_1) \\ S_2 & C_2 & 0 & (a_2 * S_2 + d_1) \\ 0 & 0 & 0 & 1\end{bmatrix} \quad \text{Eq.41}$$

$$\boldsymbol{H}\,{}_3^0 = \boldsymbol{H}\,{}_2^0 * \boldsymbol{H}\,{}_3^2 =$$

$$= \begin{bmatrix}(C_1 * C_{23}) & (-C_1 * S_{23}) & (S_1) & (a_3\,C_1\,C_{23} + a_2\,C_1 C_2 + dC_1) \\ (S_1 * C_{23}) & (-S_1 * S_{23}) & (-C_1) & (a_3\,S_1\,C_{23} + a_2\,S_1\,C_2 + dS_1) \\ S_{23} & C_{23} & 0 & (a_3\,S_{23} + a_2\,S_2 + d_1) \\ 0 & 0 & 0 & 1\end{bmatrix} \quad \text{Eq.42}$$

$$\boldsymbol{H}\,{}_4^0 = \boldsymbol{H}\,{}_3^0 * \boldsymbol{H}\,{}_4^3 =$$

$$= \begin{bmatrix}(C_1C_{234}) & (-C_1\,S_{234}) & (S_1C_{234}) & (a_4\,C_1C_{234} + a_3\,C_1\,C_{23} + a_2\,C_1\,C_2 + d\,C_1) \\ (S_1C_{234}) & (-S_1\,S_{234}) & (-C_1C_{234}) & (a_4\,S_1C_{234} + a_3\,S_1\,C_{23} + a_2\,S_1\,C_2 + d\,S_1) \\ S_{234} & C_{234} & 0 & (a_4\,S_{234} + a_3\,S_{23} + a_2\,S_2 + d_1) \\ 0 & 0 & 0 & 1\end{bmatrix} \quad \text{Eq.43}$$

A última coluna da matriz Equação 43 determina a posição do efetuador dado pela Equação 44.

$$p = \begin{bmatrix} x_E \\ y_E \\ z_E \end{bmatrix} = \begin{bmatrix} a_4\, C_1 C_{234} + a_3\, C_1\, C_{23} + a_2\, C_1\, C_2 + d\, C_1 \\ a_4\, S_1 C_{234} + a_3\, S_1\, C_{23} + a_2\, S_1\, C_2 + d\, S_1 \\ a_4\, S_{234} + a_3\, S_{23} + a_2\, S_2 + d_1 \end{bmatrix} \quad \text{Eq.44}$$

Utilizando a convenção nsa (roll, pitch e yaw) para encontrar os ângulos (Equação 45) encontra-se as rotações em torno dos eixos (Equações 46, 47 e 48):

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} (C_\omega\, C_\emptyset) & (C_\omega\, S_\emptyset\, S_\gamma - S_\omega C_\gamma) & (C_\omega\, S_\emptyset\, C_\gamma + S_\omega S_\gamma) \\ (S_\omega\, C_\emptyset) & (S_\omega\, S_\emptyset\, S_\gamma + C_\omega C_\gamma) & (S_\omega\, S_\emptyset\, C_\gamma - C_\omega S_\gamma) \\ -S_\emptyset & (\, C_\emptyset\, S_\gamma) & (\, C_\emptyset\, C_\gamma) \end{bmatrix}$$

$$\begin{cases} \tan(\emptyset_{\text{rot.Y}}) = \dfrac{-r_{31}}{\sqrt{{r_{11}}^2 + {r_{21}}^2}} = \dfrac{S_\emptyset}{\sqrt{(C_\omega\, C_\emptyset)^2 + (S_\omega\, C_\emptyset)^2}} \\ \tan(\omega_{\text{rot.Z}}) = \dfrac{r_{21}}{r_{11}} = \dfrac{(S_\omega\, C_\emptyset)}{(C_\omega\, C_\emptyset)} \\ \tan(\gamma_{\text{rot.X}}) = \dfrac{r_{32}}{r_{33}} = \dfrac{(\, C_\emptyset\, S_\gamma)}{(\, C_\emptyset\, C_\gamma)} \end{cases} \quad \text{Eq.45}$$

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} (C_1 C_{234}) & (-C_1\, S_{234}) & (S_1 C_{234}) \\ (S_1 C_{234}) & (-S_1\, S_{234}) & (-C_1 C_{234}) \\ S_{234} & C_{234} & 0 \end{bmatrix}$$

$$\tan(\emptyset_{\text{rot.Y}}) = \frac{-r_{31}}{\sqrt{{r_{11}}^2 + {r_{21}}^2}} = \frac{-S_{234}}{\sqrt{(C_1 C_{234})^2 + (S_1 C_{234})^2}} = \frac{-S_{234}}{C_{234}} = -\tan(\theta_2 + \theta_3 + \theta_4) =$$

$$\tan(\emptyset_{\text{rot.Y}}) = \tan(-\theta_2 - \theta_3 - \theta_4)$$

$$\emptyset_{\text{rot.Y}} = -(\theta_2 + \theta_3 + \theta_4) \quad \text{Eq.46}$$

$$C_\emptyset = \cos(-\theta_2 - \theta_3 - \theta_4) = \cos(\theta_2 + \theta_3 + \theta_4) = C_{234} \quad \text{Eq.47}$$

$$\tan(\omega_{\text{rot.Z}}) = \frac{r_{21}/C_\emptyset}{r_{11}/C_\emptyset} = \frac{(S_1 C_{234})/C_{234}}{(C_1 C_{234})/C_{234}} = \frac{S_1}{C_1} = \tan(\theta_1)$$

$$\omega_{\text{rot.Z}} = \theta_1 \quad \text{Eq.48}$$

$$\tan(\gamma_{\text{rot.X}}) = \frac{r_{32}/C_\emptyset}{r_{33}/C_\emptyset} = \frac{C_{234}/C_{234}}{0/C_{234}} = \infty$$

$$\tan(\gamma_{\text{rot.X}}) = \tan(\infty) = 90°$$

$$\gamma_{\text{rot.X}} = 90° \quad \text{Eq.49}$$

$$\begin{cases} C_{234} = \cos(\theta_2 + \theta_3 + \theta_4) \\ S_{234} = \text{sen}(\theta_2 + \theta_3 + \theta_4) \\ C_{23} = \cos(\theta_2 + \theta_3) = \cos(\theta_2) * \cos(\theta_3) - sen(\theta_2) * sen(\theta_3) \\ S_{23} = \text{sen}(\theta_2 + \theta_3) = \text{sen}(\theta_2) * \cos(\theta_3) + sen(\theta_3) * \cos(\theta_2) \\ C_{\emptyset} = \cos(\emptyset) \\ S_{\emptyset} = \text{sen}(\emptyset) \end{cases}$$

Os resultados encontrados pela programação utilizando o software Maple no cálculo de transformações homogêneas podem ser verificados na Tabela 8, já os comandos utilizados no mesmo software podem ser vistos no Apêndice B2.

Tabela 8: Resultados da cinemática direta usando MTH

	Valores de entrada	Valores de Saída
1	$\theta_1 = 100°\ (1.745329252\ rad)$ $\theta_2 = 20°\ (0.3490658504\ rad)$ $\theta_3 = 30°(0.5235987757\ rad)$ $\theta_4 = -50°\ (-0.8726646261\ rad)$	$x_E = -4.732537912$ $y_E = 26.83955623$ $z_E = 17.25541671$ $\emptyset_{\text{rot.Y}} = 0°$ $\omega_{\text{rot.Z}} = 100°$
2	$\theta_1 = 90°\ (1.570796327\ rad)$ $\theta_2 = 30°\ (0.5235987757\ rad)$ $\theta_3 = 40°(0.6981317011\ rad)$ $\theta_4 = -70°\ (-1.221730477\ rad)$	$x_E = 0$ $y_E = 24.37141383$ $z_E = 20.05199302$ $\emptyset_{\text{rot.Y}} = 0°$ $\omega_{\text{rot.Z}} = 90°$
3	$\theta_1 = 180°\ (3.141592654\ rad)$ $\theta_2 = 45°\ (\ 0.7853981633\ rad)$ $\theta_3 = 60°(1.047197551\ rad)$ $\theta_4 = -105°\ (-1.832595714\ rad)$	$x_E = -18.49839678$ $y_E = 0$ $z_E = 22.27206496$ $\emptyset_{\text{rot.Y}} = 0°$ $\omega_{\text{rot.Z}} = 180°$
4	$\theta_1 = 10°\ (0.1745329252\ rad)$ $\theta_2 = 50°\ (0.8726646261\ rad)$ $\theta_3 = 70°(1.221730477\ rad)$ $\theta_4 = -120°\ (-2.094395103\ rad)$	$x_E = 15.89060820$ $y_E = 2.801942963$ $z_E = 22.13295839$ $\emptyset_{\text{rot.Y}} = 90°$ $\omega_{\text{rot.Z}} = 10°$
5	$\theta_1 = 10°\ (0.1745329252\ rad)$ $\theta_2 = 50°\ (0.8726646261\ rad)$ $\theta_3 = 70°(1.221730477\ rad)$ $\theta_4 = -30°\ (-0.5235987761 rad)$	$x_E = 3.787320913$ $y_E = 0.6678068565$ $z_E = 34.42295839$ $\emptyset_{\text{rot.Y}} = 90°$ $\omega_{\text{rot.Z}} = 10°$

3.1.2 Cinemática Direta com Quatérnios Duais Unitários (QDU)

Utilizando a Equação 24 para Quatérnios Duais, tem-se os resultados para as

transformações na Cinemática Direta (Equações 50,51, 52 e 53).

$$\underline{\boldsymbol{Q}}_{1}^{0} = \begin{bmatrix} \cos(\theta_1/2)\cos(45°) \\ \cos(\theta_1/2)\,\mathrm{sen}(45°) \\ \mathrm{sen}(\theta_1/2)\,\mathrm{sen}(45°) \\ \mathrm{sen}(\theta_1/2)\cos(45°) \\ -(\mathbf{d}/2)\cos(\theta_1/2)\,\mathrm{sen}(45°) - (d_1/2)\,\mathrm{sen}(\theta_1/2)\cos(45°) \\ +(\mathbf{d}/2)\cos(\theta_1/2)\cos(45°) - (d_1/2)\,\mathrm{sen}(\theta_1/2)\,\mathrm{sen}(45°) \\ +(\mathbf{d}/2)\mathrm{sen}(\theta_1/2)\cos(45°) + (d_1/2)\cos(\theta_1/2)\,\mathrm{sen}(45°) \\ -(\mathbf{d}/2)\mathrm{sen}(\theta_1/2)\,\mathrm{sen}(45°) + (d_1/2)\cos(\theta_1/2)\cos(45°) \end{bmatrix}$$

$$\begin{cases} 45° = 0.7853981635rad \\ \boldsymbol{Cq} = \cos(45°) = \cos(0.7853981635) = 0.707106781 = \sqrt{2}/2 \\ \boldsymbol{Sq} = \mathrm{sen}(45°) = \mathrm{sen}(0.7853981635) = 0.707106781 = \sqrt{2}/2 \end{cases}$$

$$\underline{\boldsymbol{Q}}_{1}^{0} = \begin{bmatrix} \cos(\theta_1/2)\,Cq \\ \cos(\theta_1/2)\,Sq \\ \mathrm{sen}(\theta_1/2)\,Sq \\ \mathrm{sen}(\theta_1/2)\,Cq \\ -(\mathbf{d}/2)\cos(\theta_1/2)\,Sq - (d_1/2)\,\mathrm{sen}(\theta_1/2)\,Cq \\ +(\mathbf{d}/2)\cos(\theta_1/2)\,Cq - (d_1/2)\,\mathrm{sen}(\theta_1/2)\,Sq \\ +(\mathbf{d}/2)\mathrm{sen}(\theta_1/2)\,Cq + (d_1/2)\cos(\theta_1/2)\,Sq \\ -(\mathbf{d}/2)\mathrm{sen}(\theta_1/2)\,Sq + (d_1/2)\cos(\theta_1/2)\,Cq \end{bmatrix} \quad \text{Eq.50}$$

$$\underline{\boldsymbol{Q}}_{2}^{1} = \begin{bmatrix} \cos(\theta_2/2) \\ 0 \\ 0 \\ \mathrm{sen}(\theta_2/2) \\ 0 \\ +(\mathbf{a}_{\mathbf{2}}/2)\cos(\theta_2/2) \\ +(\mathbf{a}_{\mathbf{2}}/2)\mathrm{sen}(\theta_2/2) \\ 0 \end{bmatrix} \quad \text{Eq.51}$$

$$\underline{\boldsymbol{Q}}_{3}^{2} = \begin{bmatrix} \cos(\theta_3/2) \\ 0 \\ 0 \\ \mathrm{sen}(\theta_3/2) \\ 0 \\ +(\mathbf{a}_{\mathbf{3}}/2)\cos(\theta_3/2) \\ +(\mathbf{a}_{\mathbf{3}}/2)\mathrm{sen}(\theta_3/2) \\ 0 \end{bmatrix} \quad \text{Eq.52}$$

$$\underline{\boldsymbol{Q}}\,{}^{3}_{4} = \begin{bmatrix} \cos(\theta_4/2) \\ 0 \\ 0 \\ \operatorname{sen}(\theta_4/2) \\ 0 \\ +(\mathbf{a}_4/2)\cos(\theta_4/2) \\ +(\mathbf{a}_4/2)\operatorname{sen}(\theta_4/2) \\ 0 \end{bmatrix} \qquad \text{Eq.53}$$

Efetuando a multiplicação dos QDUs, tem-se a composição de transformações (Equações 54, 55 e 56):

$$\underline{\boldsymbol{Q}}\,{}^{0}_{2} = \underline{\boldsymbol{Q}}\,{}^{0}_{1} \blacksquare \underline{\boldsymbol{Q}}\,{}^{1}_{2} = \qquad \text{Eq.54}$$

$$= \begin{bmatrix} \boldsymbol{C}_{(12)} * \cos(45°) \\ \cos(\boldsymbol{A}) * \operatorname{sen}(45°) \\ \operatorname{sen}(\boldsymbol{A}) * \operatorname{sen}(45°) \\ \boldsymbol{S}_{(12)} * \cos(45°) \\ -\operatorname{sen}(45°)\left[\frac{\mathbf{a_2}}{\mathbf{2}} * \cos(\boldsymbol{A}) + \left(\frac{d}{2}\right)\boldsymbol{C}_{(12)}\right] - \left(\frac{d_1}{2}\right)\cos(45°)\,\boldsymbol{S}_{(12)} \\ \cos(45°)\left[\frac{\mathbf{a_2}}{\mathbf{2}}\boldsymbol{C}_{(12)} + \left(\frac{d}{2}\right)\cos(\boldsymbol{A})\right] - \left(\frac{d_1}{2}\right)\operatorname{sen}(45°)\operatorname{sen}(\boldsymbol{A}) \\ \cos(45°)\left[\frac{\mathbf{a_2}}{\mathbf{2}}\boldsymbol{S}_{(12)} + \left(\frac{d}{2}\right)\operatorname{sen}(\boldsymbol{A})\right] + \left(\frac{d_1}{2}\right)\operatorname{sen}(45°)\cos(\boldsymbol{A}) \\ -\operatorname{sen}(45°)\left[\frac{\mathbf{a_2}}{\mathbf{2}}\operatorname{sen}(\boldsymbol{A}) + \left(\frac{d}{2}\right)\boldsymbol{S}_{(12)}\right] + \left(\frac{d_1}{2}\right)\cos(45°)\,\boldsymbol{C}_{(12)} \end{bmatrix}$$

$$\underline{\boldsymbol{Q}}\,{}^{0}_{3} = \underline{\boldsymbol{Q}}\,{}^{0}_{2} \blacksquare \underline{\boldsymbol{Q}}\,{}^{2}_{3} = \qquad \text{Eq.55}$$

$$= \begin{bmatrix} \boldsymbol{C}_{(123)} * \cos(45°) \\ \cos(\mathbf{B}) * \operatorname{sen}(45°) \\ \operatorname{sen}(\mathbf{B}) * \operatorname{sen}(45°) \\ \boldsymbol{S}_{(123)} * \cos(45°) \\ -S_q\left[\frac{\mathbf{a_3}}{\mathbf{2}}\cos(\mathbf{B}) + \frac{\mathbf{a_2}}{\mathbf{2}}\cos(\mathbf{C}) + \left(\frac{d}{2}\right)\boldsymbol{C}_{(123)}\right] - \left(\frac{d_1}{2}\right)\cos(45°)\,\boldsymbol{S}_{(123)} \\ C_q\left[\frac{\mathbf{a_3}}{\mathbf{2}}\boldsymbol{C}_{(123)} + \frac{\mathbf{a_2}}{\mathbf{2}}\cos(\mathbf{D}) + \left(\frac{d}{2}\right)\cos(\mathbf{B})\right] - \left(\frac{d_1}{2}\right)\operatorname{sen}(45°)\operatorname{sen}(\mathbf{B}) \\ C_q\left[\frac{\mathbf{a_3}}{\mathbf{2}}\boldsymbol{S}_{(123)} + \frac{\mathbf{a_2}}{\mathbf{2}}\operatorname{sen}(\mathbf{D}) + \left(\frac{d}{2}\right)\operatorname{sen}(\mathbf{B})\right] + \left(\frac{d_1}{2}\right)\operatorname{sen}(45°)\cos(\mathbf{B}) \\ -S_q\left[\frac{\mathbf{a_3}}{\mathbf{2}}\operatorname{sen}(\mathbf{B}) + \frac{\mathbf{a_2}}{\mathbf{2}}\operatorname{sen}(\mathbf{C}) + \left(\frac{d}{2}\right)\boldsymbol{S}_{(123)}\right] + \left(\frac{d_1}{2}\right)\cos(45°)\,\boldsymbol{C}_{(123)} \end{bmatrix}$$

$\mathbf{A} = \frac{\theta_1 - \theta_2}{2}$	$\mathbf{B} = \frac{\theta_1 - \theta_2 - \theta_3}{2}$
$\mathbf{C} = \frac{\theta_1 - \theta_2 + \theta_3}{2}$	$\mathbf{D} = \frac{\theta_1 + \theta_2 - \theta_3}{2}$

$$\begin{cases} C_{(12)} = \cos\left(\dfrac{\theta_1+\theta_2}{2}\right) = \cos(\theta_1/2)\cos(\theta_2/2) - sen(\theta_1/2)sen(\theta_2/2) \\ S_{(12)} = \text{sen}\left(\dfrac{\theta_1+\theta_2}{2}\right) = \text{sen}(\theta_1/2)\cos(\theta_2/2) + sen(\theta_2/2)\cos(\theta_1/2) \\ C_{(123)} = \cos\left(\dfrac{\theta_1+\theta_2+\theta_3}{2}\right) \\ S_{(123)} = \text{sen}\left(\dfrac{\theta_1+\theta_2+\theta_3}{2}\right) \\ C_{45°} = \cos(45°) \\ S_{45°} = \text{sen}(45°) \end{cases}$$

$$\underline{\boldsymbol{Q}}^{0}_{4} = \underline{\boldsymbol{Q}}^{0}_{3} \blacksquare \underline{\boldsymbol{Q}}^{3}_{4} = \qquad \text{Eq.56}$$

$$= \begin{bmatrix} \boldsymbol{C_E} * \cos(45°) \\ \boldsymbol{C_F} * \text{sen}(45°) \\ -\boldsymbol{S_F} * \text{sen}(45°) \\ \boldsymbol{S_E} * \cos(45°) \\ \text{sen}(45°) * \left[-\dfrac{\mathbf{a_4}}{\mathbf{2}}\boldsymbol{C_F} - \dfrac{\mathbf{a_3}}{\mathbf{2}}\boldsymbol{C_G} - \dfrac{\mathbf{a_2}}{\mathbf{2}}\boldsymbol{C_H} - \left(\dfrac{d}{2}\right)\boldsymbol{C_E}\right] - \left(\dfrac{d_1}{2}\right)\cos(45°)\,\boldsymbol{S_E} \\ \cos(45°) * \left[+\dfrac{\mathbf{a_4}}{\mathbf{2}}\boldsymbol{C_E} + \dfrac{\mathbf{a_3}}{\mathbf{2}}\boldsymbol{C_K} + \dfrac{\mathbf{a_2}}{\mathbf{2}}\boldsymbol{C_J} + \left(\dfrac{d}{2}\right)\boldsymbol{C_F}\right] + \left(\dfrac{d_1}{2}\right)\text{sen}(45°)\,\boldsymbol{S_F} \\ \cos(45°) * \left[+\dfrac{\mathbf{a_4}}{\mathbf{2}}\boldsymbol{S_E} - \dfrac{\mathbf{a_3}}{\mathbf{2}}\boldsymbol{S_K} - \dfrac{\mathbf{a_2}}{\mathbf{2}}\boldsymbol{S_J} - \left(\dfrac{d}{2}\right)\boldsymbol{S_F}\right] + \left(\dfrac{d_1}{2}\right)\text{sen}(45°)\,\boldsymbol{C_F} \\ \text{sen}(45°) * \left[+\dfrac{\mathbf{a_4}}{\mathbf{2}}\boldsymbol{S_F} - \dfrac{\mathbf{a_3}}{\mathbf{2}}\boldsymbol{S_G} - \dfrac{\mathbf{a_2}}{\mathbf{2}}\boldsymbol{S_H} - \left(\dfrac{d}{2}\right)\boldsymbol{S_E}\right] + \left(\dfrac{d_1}{2}\right)\cos(45°)\,\boldsymbol{C_E} \end{bmatrix}$$

$$\begin{cases} \boldsymbol{C_E} = \cos(E) & \boldsymbol{S_E} = \text{sen}(E) \\ \boldsymbol{C_F} = \cos(F) & \boldsymbol{S_F} = \text{sen}(F) \\ \boldsymbol{C_G} = \cos(G) & \boldsymbol{S_G} = \text{sen}(G) \\ \boldsymbol{C_H} = \cos(H) & \boldsymbol{S_H} = \text{sen}(H) \\ \boldsymbol{C_J} = \cos(J) & \boldsymbol{S_J} = \text{sen}(J) \\ \boldsymbol{C_K} = \cos(K) & \boldsymbol{S_K} = \text{sen}(K) \end{cases}$$

$\boldsymbol{E} = \dfrac{\theta_1+\theta_2+\theta_3+\theta_4}{2}$	$\boldsymbol{F} = \dfrac{-\theta_1+\theta_2+\theta_3+\theta_4}{2}$	$\boldsymbol{G} = \dfrac{+\theta_1-\theta_2-\theta_3+\theta_4}{2}$
$\boldsymbol{H} = \dfrac{+\theta_1-\theta_2+\theta_3+\theta_4}{2}$	$\boldsymbol{J} = \dfrac{-\theta_1-\theta_2+\theta_3+\theta_4}{2}$	$\boldsymbol{K} = \dfrac{-\theta_1-\theta_2-\theta_3+\theta_4}{2}$

Usando a Equação 25 ou Equação 26 para extrair do quatérnio dual final ($\underline{\boldsymbol{Q}}^{0}_{4}$) a sua translação, tem-se a Equação 57 de translação final (Equações 58,59 e 60).

$$\boldsymbol{transl}\left(\underline{\boldsymbol{Q}}\,{}^{0}_{4}\right) = \qquad \text{Eq.57}$$

$$= \mathbf{2} * \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ p_5 \\ p_6 \\ p_7 \\ p_8 \end{bmatrix} \blacksquare \begin{bmatrix} \boldsymbol{C_E} * \cos(45°) \\ -\boldsymbol{C_F} * \operatorname{sen}(45°) \\ +\boldsymbol{S_F} * \operatorname{sen}(45°) \\ -\boldsymbol{S_E} * \cos(45°) \\ Sq * \left[-\frac{\mathbf{a_4}}{\mathbf{2}}\boldsymbol{C_F} - \frac{\mathbf{a_3}}{\mathbf{2}}\boldsymbol{C_G} - \frac{\mathbf{a_2}}{\mathbf{2}}\boldsymbol{C_H} - \left(\frac{d}{2}\right)\boldsymbol{C_E}\right] - \left(\frac{d_1}{2}\right) Cq\, \boldsymbol{S_E} \\ -Cq * \left[+\frac{\mathbf{a_4}}{\mathbf{2}}\boldsymbol{C_E} + \frac{\mathbf{a_3}}{\mathbf{2}}\boldsymbol{C_K} + \frac{\mathbf{a_2}}{\mathbf{2}}\boldsymbol{C_J} + \left(\frac{d}{2}\right)\boldsymbol{C_F}\right] - \left(\frac{d_1}{2}\right) Sq\, \boldsymbol{S_F} \\ -Cq * \left[+\frac{\mathbf{a_4}}{\mathbf{2}}\boldsymbol{S_E} - \frac{\mathbf{a_3}}{\mathbf{2}}\boldsymbol{S_K} - \frac{\mathbf{a_2}}{\mathbf{2}}\boldsymbol{S_J} - \left(\frac{d}{2}\right)\boldsymbol{S_F}\right] - \left(\frac{d_1}{2}\right) Sq\, \boldsymbol{C_F} \\ -Sq * \left[+\frac{\mathbf{a_4}}{\mathbf{2}}\boldsymbol{S_F} - \frac{\mathbf{a_3}}{\mathbf{2}}\boldsymbol{S_G} - \frac{\mathbf{a_2}}{\mathbf{2}}\boldsymbol{S_H} - \left(\frac{d}{2}\right)\boldsymbol{S_E}\right] - \left(\frac{d_1}{2}\right) Cq\, \boldsymbol{C_E} \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ x_E \\ y_E \\ z_E \end{bmatrix}$$

$$\begin{cases} \boldsymbol{Sq} = \operatorname{sen}(45°) \\ \boldsymbol{Cq} = \cos(45°) \\ p_5 = Sq\left[-\frac{\mathbf{a_4}}{\mathbf{2}}\boldsymbol{C_F} - \frac{\mathbf{a_3}}{\mathbf{2}}\boldsymbol{C_G} - \frac{\mathbf{a_2}}{\mathbf{2}}\boldsymbol{C_H} - \left(\frac{d}{2}\right)\boldsymbol{C_E}\right] - \left(\frac{d_1}{2}\right) Cq\, \boldsymbol{S_E} \\ p_6 = Cq\left[+\frac{\mathbf{a_4}}{\mathbf{2}}\boldsymbol{C_E} + \frac{\mathbf{a_3}}{\mathbf{2}}\boldsymbol{C_K} + \frac{\mathbf{a_2}}{\mathbf{2}}\boldsymbol{C_J} + \left(\frac{d}{2}\right)\boldsymbol{C_F}\right] + \left(\frac{d_1}{2}\right) Sq\, \boldsymbol{S_F} \\ p_7 = Cq\left[+\frac{\mathbf{a_4}}{\mathbf{2}}\boldsymbol{S_E} - \frac{\mathbf{a_3}}{\mathbf{2}}\boldsymbol{S_K} - \frac{\mathbf{a_2}}{\mathbf{2}}\boldsymbol{S_J} - \left(\frac{d}{2}\right)\boldsymbol{S_F}\right] + \left(\frac{d_1}{2}\right) Sq\, \boldsymbol{C_F} \\ p_8 = Sq\left[+\frac{\mathbf{a_4}}{\mathbf{2}}\boldsymbol{S_F} - \frac{\mathbf{a_3}}{\mathbf{2}}\boldsymbol{S_G} - \frac{\mathbf{a_2}}{\mathbf{2}}\boldsymbol{S_H} - \left(\frac{d}{2}\right)\boldsymbol{S_E}\right] + \left(\frac{d_1}{2}\right) Cq\, \boldsymbol{C_E} \end{cases}$$

$$\begin{aligned} \boldsymbol{x_E} = {} & \cos(45°)^2\left[\mathbf{a_4}\left(\boldsymbol{C_E}^2 - \boldsymbol{S_E}^2\right) + \mathbf{a_3}(\boldsymbol{C_E C_K} + \boldsymbol{S_E S_K}) + \mathbf{a_2}\left(\boldsymbol{C_E C_J} + \boldsymbol{S_E S_J}\right)\right] \\ & + \operatorname{sen}(45°)^2\left[\mathbf{a_4}\left(\boldsymbol{C_F}^2 - \boldsymbol{S_F}^2\right) + \mathbf{a_3}(\boldsymbol{C_G C_F} + \boldsymbol{S_G S_F}) + \mathbf{a_2}(\boldsymbol{C_F C_H} \right. \\ & \left. + \boldsymbol{S_F S_H})\right] \\ & + 2\operatorname{sen}(45°)\left[\left(\frac{d_1}{2}\right)\cos(45°)\,(-\boldsymbol{C_E S_F} + \boldsymbol{C_F S_E}) \right. \\ & \left. + \left(\frac{d}{2}\right)\operatorname{sen}(45°)\,(\boldsymbol{C_E C_F} + \boldsymbol{S_E S_F})\right] \\ & + 2\cos(45°)\left[\left(\frac{d}{2}\right)\cos(45°)\,(\boldsymbol{C_E C_F} + \boldsymbol{S_E S_F}) \right. \\ & \left. + \left(\frac{d_1}{2}\right)\operatorname{sen}(45°)\,(\boldsymbol{C_E S_F} - \boldsymbol{C_F S_E})\right] \end{aligned}$$

$$\begin{aligned}\boldsymbol{x_E} = \cos(45°)^2\,[\mathbf{a_4}(\boldsymbol{C_{234}C_1} - \boldsymbol{S_{234}S_1}) + \mathbf{a_3}(\boldsymbol{C_1C_{23}} - \boldsymbol{S_1S_{23}}) + \mathbf{a_2}(\boldsymbol{C_{12}})]\\ + \operatorname{sen}(45°)^2\,[\mathbf{a_4}(\boldsymbol{C_{234}C_1} + \boldsymbol{S_{234}S_1}) + \mathbf{a_3}(\boldsymbol{C_1C_{23}} + \boldsymbol{S_1S_{23}}) + \mathbf{a_2}(\boldsymbol{C_1C_2}\\ + \boldsymbol{S_1S_2})]\\ + 2\operatorname{sen}(45°)\left[\left(\frac{d_1}{2}\right)\cos(45°)\,(\boldsymbol{S_1}) + \left(\frac{d}{2}\right)\operatorname{sen}(45°)\,(\boldsymbol{C_1})\right]\\ + 2\cos(45°)\left[\left(\frac{d}{2}\right)\cos(45°)\,(\boldsymbol{C_1}) + \left(\frac{d_1}{2}\right)\operatorname{sen}(45°)\,(-\boldsymbol{S_1})\right]\end{aligned}$$

$$\begin{aligned}\boldsymbol{x_E} = {}& \mathrm{a}_4\boldsymbol{C_{234}C_1}[\operatorname{sen}(45°)^2 + \cos(45°)^2]\\ &+ \mathrm{a}_4\boldsymbol{S_{234}S_1}[\operatorname{sen}(45°)^2 - \cos(45°)^2]\\ &+ \mathrm{a}_3\boldsymbol{C_1C_{23}}[\operatorname{sen}(45°)^2 + \cos(45°)^2]\\ &+ \mathrm{a}_3\boldsymbol{S_1S_{23}}[\operatorname{sen}(45°)^2 - \cos(45°)^2]\\ &+ \mathrm{a}_2\boldsymbol{C_1C_2}[\operatorname{sen}(45°)^2 + \cos(45°)^2] + \mathrm{a}_2\boldsymbol{S_1S_2}\,[\operatorname{sen}(45°)^2 - \cos(45°)^2]\\ &+ d\boldsymbol{C_1}[\operatorname{sen}(45°)^2 + \cos(45°)^2]\end{aligned}$$

$$\boldsymbol{x_E} = \mathrm{a}_4\boldsymbol{C_{234}C_1} + \mathrm{a}_3\boldsymbol{C_1C_{23}} + \mathrm{a}_2\boldsymbol{C_1C_2} + d\boldsymbol{C_1} \qquad \text{Eq.58}$$

$$\begin{aligned}\boldsymbol{y_E} = \cos(45°)^2\,[\mathbf{a_4}(2\boldsymbol{C_ES_E}) + \mathbf{a_3}(-\boldsymbol{C_ES_K} + \boldsymbol{C_KS_E}) - \mathbf{a_2}(\boldsymbol{C_ES_J} - \boldsymbol{C_JS_E}) + d(-\boldsymbol{C_ES_F}\\ + \boldsymbol{C_FS_E})]\\ - \operatorname{sen}(45°)^2\,[\mathbf{a_4}(2\boldsymbol{C_FS_F}) + \mathbf{a_3}(\boldsymbol{C_GS_F} - \boldsymbol{S_GC_F}) + \mathbf{a_2}(\boldsymbol{C_HS_F} - \boldsymbol{S_HC_F})\\ + d(\boldsymbol{C_ES_F} - \boldsymbol{C_FS_E})]\end{aligned}$$

$$\begin{aligned}\boldsymbol{y_E} = \cos(45°)^2\,[\mathbf{a_4}(\boldsymbol{S_{234}C_1} + \boldsymbol{C_\emptyset S_1}) + \mathbf{a_3}(\boldsymbol{S_1C_{23}} + \boldsymbol{C_1S_{23}}) - \mathbf{a_2}(-\boldsymbol{S_{12}}) + d(\boldsymbol{S_1})]\\ - \operatorname{sen}(45°)^2\,[\mathbf{a_4}(\boldsymbol{S_{234}C_1} - \boldsymbol{C_\emptyset S_1}) + \mathbf{a_3}(\boldsymbol{S_{23}C_1} - \boldsymbol{S_1C_{23}}) + \mathbf{a_2}(\boldsymbol{S_2C_1} - \boldsymbol{S_1C_2})\\ + d(-\boldsymbol{S_1})]\end{aligned}$$

$$\boldsymbol{y_E} = \mathrm{a}_4\boldsymbol{C_{234}S_1} + \mathrm{a}_3\boldsymbol{S_1C_{23}} + \mathrm{a}_2\boldsymbol{S_1C_2} + d\boldsymbol{S_1} \qquad \text{Eq.59}$$

$$\begin{aligned}\boldsymbol{z_E} = {}& + \operatorname{sen}(45°)\cos(45°)\left[2\mathbf{a_4}(\boldsymbol{C_ES_F} + \boldsymbol{C_FS_E})\right.\\ &- \mathbf{a_3}\big((\boldsymbol{C_ES_G} - \boldsymbol{C_GS_E}) + (-\boldsymbol{C_KS_F} + \boldsymbol{C_FS_K})\big) - \mathbf{a_2}((\boldsymbol{C_ES_H} - \boldsymbol{C_HS_E})\\ &\left.+ \big(-\boldsymbol{C_JS_F} + \boldsymbol{C_FS_J}\big)\big)\right]\\ &+ d_1\left[\cos(45°)^2\left(\boldsymbol{C_E}^2 + \boldsymbol{S_E}^2\right) + \operatorname{sen}(45°)^2\left(\boldsymbol{C_F}^2 + \boldsymbol{S_F}^2\right)\right]\end{aligned}$$

$$\begin{aligned}\boldsymbol{z_E} = {}& + \operatorname{sen}(45°)\cos(45°)\left[2\mathbf{a_4}(\boldsymbol{S_{234}}) - \mathbf{a_3}\big((-\boldsymbol{S_{23}}) + (-\boldsymbol{S_{23}})\big) - \mathbf{a_2}((-\boldsymbol{S_2})\right.\\ &\left.+ (-\boldsymbol{S_2}))\right] + d_1[\cos(45°)^2\,(\mathbf{1}) + \operatorname{sen}(45°)^2\,(\mathbf{1})]\end{aligned}$$

$$\boldsymbol{z_E} = \mathrm{a}_4\boldsymbol{S_{234}} + \mathrm{a}_3\boldsymbol{S_{23}} + \mathrm{a}_2\boldsymbol{S_2} + \boldsymbol{d_1} \qquad \text{Eq.60}$$

A matriz de rotações para comparação com o sistema nsa é dada pela Equação 61, os ângulos de rotação em torno dos eixos são obtidos (Equações 62, 63 e 64):

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} (1 - 2{q_3}^2 - 2{q_4}^2) & 2(q_2q_3 - q_4q_1) & 2(q_2q_4 + q_3q_1) \\ 2(q_2q_3 + q_4q_1) & (1 - 2{q_2}^2 - 2{q_4}^2) & 2(q_3q_4 - q_2q_1) \\ 2(q_2q_4 - q_3q_1) & 2(q_3q_4 + q_2q_1) & (1 - 2{q_2}^2 - 2{q_3}^2) \end{bmatrix}$$

$$\begin{cases} \emptyset_{\text{rot.Y}} = \text{atan2}\left(\dfrac{-2q_2q_4 + 2q_1q_3}{\sqrt{(1-2{q_3}^2-2{q_4}^2)^2 + (2q_2q_3+2q_1q_4)^2}}\right) \\ \omega_{\text{rot.Z}} = \text{atan2}\left(\dfrac{2q_2q_3 + 2q_1q_4}{1-2{q_3}^2-2{q_4}^2}\right) \\ \gamma_{\text{rot.X}} = \text{atan2}\left(\dfrac{2q_1q_2 + 2q_3q_4}{1-2{q_2}^2-2{q_3}^2}\right) \end{cases} \quad \text{Eq.61}$$

$$\emptyset_{\text{rot.Y}} = \text{atan2}\left(\frac{-r_{31}}{\sqrt{{r_{11}}^2+{r_{21}}^2}}\right) = \text{atan2}\left(\frac{-2q_2q_4+2q_1q_3}{\sqrt{(1-2{q_3}^2-2{q_4}^2)^2+(2q_2q_3+2q_1q_4)^2}}\right)$$

$$\tan(\emptyset_{\text{rot.Y}}) = \frac{-C_E S_F - C_F S_E}{\sqrt{(-{S_E}^2 - {S_F}^2 + 1)^2 + (C_E S_E - C_F S_F)^2}}$$

$$\tan(\emptyset_{\text{rot.Y}}) = \frac{-1(C_E S_F + C_F S_E)}{\sqrt{(C_1 * C_{234})^2 + (S_1 * C_{234})^2}}$$

$$\tan(\emptyset_{\text{rot.Y}}) = \frac{-1(S_{234})}{\sqrt{{C_{234}}^2({S_1}^2 + {C_1}^2)}}$$

$$\tan(\emptyset_{\text{rot.Y}}) = \frac{-1(S_{234})}{C_{234}}$$

$$\tan(\emptyset_{\text{rot.Y}}) = \tan2(-\theta_2 - \theta_3 - \theta_4)$$

$$\emptyset_{\text{rot.Y}} = -(\theta_2 + \theta_3 + \theta_4) \quad \text{Eq.62}$$

$$\omega_{\text{rot.Z}} = \text{atan2}\left(\frac{r_{21}}{r_{11}}\right) = \text{atan2}\left(\frac{2q_2q_3+2q_1q_4}{1-2{q_3}^2-2{q_4}^2}\right)$$

$$\tan(\omega_{\text{rot.Z}}) = \frac{-2C_{(1)} * S_{(1)} * \left(C_{(234)}\right)^2 + 2C_{(1)}S_{(1)} * (S_{(234)})^2}{2{C_{(1)}}^2 * \left(S_{(234)}\right)^2 + 2{S_{(1)}}^2 * \left(C_{(234)}\right)^2 - 1}$$

$$\tan(\omega_{\text{rot.Z}}) = \frac{-S_1\left(C_{(234)}\right)^2 + S_1(S_{(234)})^2}{2{C_{(1)}}^2\left(1 - C_{(234)}\right)^2 + 2\left(1 - C_{(1)}\right)^2\left(C_{(234)}\right)^2 - 1}$$

$$\tan(\omega_{\text{rot.Z}}) = \frac{-S_1\left({C_{(234)}}^2 - {S_{(234)}}^2\right)}{-4{C_{(1)}}^2\left(C_{(234)}\right)^2 + 2\left(C_{(1)}\right)^2 + 2\left(C_{(234)}\right)^2 - 1}$$

$$\tan(\omega_{\text{rot.Z}}) = \frac{-S_1(C_{234})}{\left(2{C_{(1)}}^2 - 1\right)(-1)\left(2{C_{(234)}}^2 - 1\right)}$$

$$\tan(\omega_{\text{rot.Z}}) = \frac{-S_1(C_{234})}{(C_1)(-1)(C_{234})} = \frac{-S_1}{(C_1)(-1)}$$

$$\omega_{\text{rot.Z}} = \theta_1 \qquad \text{Eq.63}$$

$$\gamma_{\text{rot.X}} = \text{atan2}\left(\frac{r_{32}}{r_{33}}\right) = \text{atan2}\left(\frac{2q_1q_2+2q_3q_4}{1-2{q_2}^2-2{q_3}^2}\right)$$

$$\tan(\gamma_{\text{rot.X}}) = \frac{r_{32}}{r_{33}} = \frac{C_E C_F - S_F S_E}{-{C_F}^2 - {S_F}^2 + 1}$$

$$\frac{C_E C_F - S_F S_E}{-{C_F}^2 - {S_F}^2 + 1} = \frac{C_{234}}{\left(-{C_1}^2 - {S_1}^2\right){C_{234}}^2 + 1 + \left(-{C_1}^2 - {S_1}^2\right){S_{234}}^2}$$

$$\frac{C_{234}}{0} = \infty$$

$$\tan(\gamma_{\text{rot.X}}) = \tan(\infty) = 90°$$

$$\gamma_{\text{rot.X}} = 90° \qquad \text{Eq.64}$$

Utilizando a Tabela 7 Denavit Hartenberg do manipulador, tem-se os resultados encontrados pelo software Maple no cálculo de transformações fazendo uso dos quatérnios duais (Tabela 9), os comandos utilizados no software podem ser vistos por completo no Apêndice B3.

Tabela 9: Resultados da Cinemática Direta usando QD

	Valores de entrada	Valores de Saída
1	$\theta_1 = 100°\ (1.745329252\ rad)$ $\theta_2 = 20°\ (0.3490658504\ rad)$ $\theta_3 = 30°(0.5235987757\ rad)$ $\theta_4 = -50°\ (-0.8726646261\ rad)$	$x_E = -4.7325379115490$ $y_E = 26.8395562329541$ $z_E = 17.2554167076794$ $\emptyset_{\text{rot.Y}} = 0°$ $\omega_{\text{rot.Z}} = 100°$
2	$\theta_1 = 90°\ (1.570796327\ rad)$ $\theta_2 = 30°\ (0.5235987757\ rad)$ $\theta_3 = 40°(0.6981317011\ rad)$ $\theta_4 = -70°\ (-1.221730477\ rad)$	$x_E = 0$ $y_E = 24.3714138397228$ $z_E = 20.0519930152759$ $\emptyset_{\text{rot.Y}} = 0°$ $\omega_{\text{rot.Z}} = 90°$
3	$\theta_1 = 180°\ (3.141592654\ rad)$ $\theta_2 = 45°\ (\ 0.7853981633\ rad)$ $\theta_3 = 60°(1.047197551\ rad)$ $\theta_4 = -105°\ (-1.832595714\ rad)$	$x_E = -18.498396780499$ $y_E = 0$ $z_E = 22.2720649576841$ $\emptyset_{\text{rot.Y}} = 0°$ $\omega_{\text{rot.Z}} = 180°$

4	$\theta_1 = 10°\ (0.1745329252\ rad)$ $\theta_2 = 50°\ (0.8726646261\ rad)$ $\theta_3 = 70°(1.221730477\ rad)$ $\theta_4 = -120°\ (-2.094395103\ rad)$	$x_E = 15.8906082011919$ $y_E = 2.80194296545369$ $z_E = 22.1329583666828$ $\emptyset_{\text{rot.Y}} = 90°$ $\omega_{\text{rot.Z}} = 10°$
5	$\theta_1 = 10°\ (0.1745329252\ rad)$ $\theta_2 = 50°\ (0.8726646261\ rad)$ $\theta_3 = 70°(1.221730477\ rad)$ $\theta_4 = -30°\ (-0.5235987761 rad)$	$x_E = 3.78732091962131$ $y_E = 0.667806861157773$ $z_E = 34.4229583767364$ $\emptyset_{\text{rot.Y}} = 90°$ $\omega_{\text{rot.Z}} = 10°$

3.2 Cinemática Inversa do manipulador

3.2.1 Cinemática Inversa com Matrizes de Transformação Homogênea (MTH)

A Cinemática inversa usando Matrizes de Transformação Homogênea foi feita passo a passo conforme Equações 65, 66 e 67:

$$(\boldsymbol{H}\,{}^{0}_{1})^{-1} = \begin{bmatrix} \cos\theta_1 & \operatorname{sen}\theta_1 & 0 & -\mathbf{d} \\ 0 & 0 & 1 & -d_1 \\ \operatorname{sen}\theta_1 & -\cos\theta_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Eq.65}$$

$$\boldsymbol{H}\,{}^{0}_{4} = \begin{bmatrix} n_x & s_x & a_x & x_E \\ n_y & s_y & a_y & y_E \\ n_z & s_z & a_z & z_E \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Eq.66}$$

$$\boldsymbol{H}\,{}^{1}_{4} = \begin{bmatrix} 1 & 0 & 0 & (a_3 * C_{23} + a_2 * C_2) \\ 0 & 1 & 0 & (a_3 * S_{23} + a_2 * S_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Eq.67}$$

Por comparação com a transformação $1 \rightarrow 4$, da composição de transformações, tem-se as Equação 68 que resulta na Equação 69 que descreve o ângulo para a primeira junta do manipulador:

$$(\boldsymbol{H}\,{}^{0}_{1})^{-1} * \boldsymbol{H}\,{}^{0}_{4} = \boldsymbol{H}\,{}^{1}_{4} \quad \text{Eq.68}$$

$$\begin{bmatrix} \cos\theta_1 & \operatorname{sen}\theta_1 & 0 & -\mathbf{d} \\ 0 & 0 & 1 & -d_1 \\ \operatorname{sen}\theta_1 & -\cos\theta_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} n_x & s_x & a_x & x_E \\ n_y & s_y & a_y & y_E \\ n_z & s_z & a_z & z_E \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$= \begin{bmatrix} 1 & 0 & 0 & (a_3 * C_{23} + a_2 * C_2) \\ 0 & 1 & 0 & (a_3 * S_{23} + a_2 * S_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\operatorname{sen}\theta_1 * x_E - \cos\theta_1 * y_E + 0 * z_E = 0$$

$$\operatorname{sen}\theta_1 * x_E = \cos\theta_1 * y_E$$

$$\frac{\operatorname{sen}\theta_1}{\cos\theta_1} = \frac{y_E}{x_E}$$

$$\theta_1 = arctan2\left(\frac{y_E}{x_E}\right) \qquad \text{Eq.69}$$

Por comparação com a transformação 3 → 4 , da composição de transformações, tem-se as Equação 70 e 71 que resulta em sistemas que quando resolvidos chegam à Equação 72 que descreve o ângulo para a terceira junta do manipulador:

$$(\boldsymbol{H}\,{}^{0}_{3})^{-1} = \begin{bmatrix} (C_1\,C_{23}) & (S_1\,C_{23}) & (S_{23}) & (-a_2\,C_3 - d\,C_{23} - d_1\,S_{23} - a_3) \\ (-C_1\,S_{23}) & (-S_1\,S_{23}) & (C_{23}) & (+a_2\,S_3 + d\,S_{23} - d_1\,C_{23}) \\ (S_1) & (-C_1) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{Eq.70}$$

$$(\boldsymbol{H}\,{}^{0}_{3})^{-1} \;*\; \boldsymbol{H}\,{}^{0}_{4} \;=\; \boldsymbol{H}\,{}^{3}_{4} \qquad \text{Eq.71}$$

$$\begin{bmatrix} (C_1 * C_{23}) & (S_1 * C_{23}) & (S_{23}) & (-a_2 * C_3 - d * C_{23} - d_1 * S_{23} - a_3) \\ (-C_1 * S_{23}) & (-S_1 * S_{23}) & (C_{23}) & (+a_2 * S_3 + d * S_{23} - d_1 * C_{23}) \\ (S_1) & (-C_1) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} n_x & s_x & a_x & x_E \\ n_y & s_y & a_y & y_E \\ n_z & s_z & a_z & z_E \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} \cos\theta_4 & -\operatorname{sen}\theta_4 & 0 & a_4 * \cos\theta_4 \\ \operatorname{sen}\theta_4 & \cos\theta_4 & 0 & a_4 * \operatorname{sen}\theta_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{cases} \textbf{(I)}\ (C_1\,C_{23})\,\boldsymbol{x_E} + (S_1\,C_{23})\,\boldsymbol{y_E} + (S_{23})\,\boldsymbol{z_E} - a_2\,C_3 - d\,C_{23} - d_1\,S_{23} - a_3 = \\ \qquad = a_4\,(C_4) \\ \textbf{(II)}\ (-C_1\,S_{23})\,\boldsymbol{x_E} + (-S_1\,S_{23})\,\boldsymbol{y_E} + (C_{23})\,\boldsymbol{z_E} + a_2\,S_3 + d\,S_{23} - d_1\,C_{23} = \\ \qquad = a_4\,(S_4) \end{cases}$$

$$\begin{cases} \textbf{(I)}\ (C_1\,C_{23})\,\boldsymbol{x_E} + (S_1\,C_{23})\,\boldsymbol{y_E} + (S_{23})\,\boldsymbol{z_E} - a_2\,C_3 - d\,C_{23} - d_1\,S_{23} - a_3 = \\ \qquad = a_4\,(C_\emptyset C_{23} - S_\emptyset S_{23}) \\ \textbf{(II)}\ (-C_1\,S_{23})\,\boldsymbol{x_E} + (-S_1\,S_{23})\,\boldsymbol{y_E} + (C_{23})\,\boldsymbol{z_E} + a_2\,S_3 + d\,S_{23} - d_1\,C_{23} = \\ \qquad = a_4\,(+C_\emptyset S_{23} - S_\emptyset C_{23}) \end{cases}$$

$$\begin{cases} \textbf{(I)}\quad (C_1\boldsymbol{x_E} + S_1\boldsymbol{y_E} - d - a_4C_\emptyset) * C_{23} + (\boldsymbol{z_E} - \mathrm{d}_1 + a_4S_\emptyset) * S_{23} = a_3 + a_2C_3 \\ \textbf{(II)}\quad (\boldsymbol{z_E} - \mathrm{d}_1 + a_4S_\emptyset)\,C_{23} - (C_1\,\boldsymbol{x_E} + S_1\,\boldsymbol{y_E} - d - a_4C_\emptyset)\,S_{23} = -\,a_2\,S_3 \end{cases}$$

$$\boldsymbol{P} = (C_1 * \boldsymbol{x_E} + S_1 * \boldsymbol{y_E} - d - a_4C_\emptyset)$$
$$\boldsymbol{Q} = (\boldsymbol{z_E} - \mathrm{d}_1 + a_4S_\emptyset)$$

$$\begin{cases} \textbf{(I)}\quad \boldsymbol{P} * C_{23} + \mathbf{Q} * S_{23} = a_3 + a_2C_3 \\ \textbf{(II)}\quad \boldsymbol{Q} * C_{23} - \mathbf{P} * S_{23} = -\,a_2S_3 \end{cases}$$

Elevando (I) e (II) ao quadrado:

$$\begin{cases} \textbf{(I)} \quad \boldsymbol{P^2}\, C_{23}{}^2 + 2\mathbf{PQ} * S_{23}\, C_{23} + \boldsymbol{Q^2}\, S_{23}{}^2 = a_3{}^2 + 2\, a_2\, a_3\, C_3 + a_2{}^2\, C_3{}^2 \\ \textbf{(II)} \quad \boldsymbol{P^2}\, S_{23}{}^2 - 2\mathbf{PQ} * S_{23}\, C_{23} + \boldsymbol{Q^2}\, C_{23}{}^2 = a_2{}^2\, S_3{}^2 \end{cases}$$

Somando (I) e (II) :

$$\boldsymbol{P^2} + \boldsymbol{Q^2} = a_2{}^2 + a_3{}^2 + 2 * a_2 * a_3 * C_3$$

$$C_3 = \frac{\boldsymbol{P^2} + \boldsymbol{Q^2} - a_2{}^2 - a_3{}^2}{2 * a_2 * a_3}$$

$$C_3 = \frac{(C_1 * \boldsymbol{x_E} + S_1 * \boldsymbol{y_E} - d - a_4 C_\emptyset)^2 + (\boldsymbol{z_E} - \mathrm{d}_1 + a_4 S_\emptyset)^2 - a_2{}^2 - a_3{}^2}{2 * a_2 * a_3} = \boldsymbol{\beta}$$

$$C_3 = \frac{\left(\sqrt{\boldsymbol{x_E}^2 + \boldsymbol{y_E}^2} - d - a_4 C_\emptyset\right)^2 + (\boldsymbol{z_E} - \mathrm{d}_1 + a_4 S_\emptyset)^2 - a_2{}^2 - a_3{}^2}{2 * a_2 * a_3} = \boldsymbol{\beta}$$

$$S_3 = \sqrt{\mathbf{1} - \boldsymbol{\beta}^2}$$

$$\theta_3 = arctan2\left(\frac{\pm\sqrt{\mathbf{1} - \boldsymbol{\beta}^2}}{\boldsymbol{\beta}}\right) \qquad \text{Eq.72}$$

Por comparação com a transformação $0 \rightarrow 4$, da composição de transformações, tem-se as Equação 73 que resolvendo o sistema de equações resulta na Equação 74 que descreve o ângulo para a segunda junta do manipulador:

$$\boldsymbol{H}_4^0 = \begin{bmatrix} n_x & s_x & a_x & x_E \\ n_y & s_y & a_y & y_E \\ n_z & s_z & a_z & z_E \\ 0 & 0 & 0 & 1 \end{bmatrix} = \qquad \text{Eq.73}$$

$$= \begin{bmatrix} (C_1 C_{234}) & (-C_1\, S_{234}) & (S_1 C_{234}) & (a_4\, C_1 C_{234} + a_3\, C_1\, C_{23} + a_2\, C_1\, C_2 + d\, C_1) \\ (S_1 C_{234}) & (-S_1\, S_{234}) & (-C_1 C_{234}) & (a_4\, S_1 C_{234} + a_3\, S_1\, C_{23} + a_2\, S_1\, C_2 + d\, S_1) \\ S_{234} & C_{234} & 0 & (a_4\, S_{234} + a_3\, S_{23} + a_2\, S_2 + d_1) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{cases} \textbf{(I)} \quad (a_4 * C_{234} + a_3 * C_{23} + a_2 * C_2 + d) * C_1 = \boldsymbol{x_E} \\ \textbf{(II)} \quad (a_4 * C_{234} + a_3 * C_{23} + a_2 * C_2 + d) * S_1 = \boldsymbol{y_E} \\ \textbf{(III)} \quad a_4 * S_{234} + a_3 * S_{23} + a_2 * S_2 + d_1 = \boldsymbol{z_E} \end{cases}$$

Elevando (I) e (II) ao quadrado e somando os resultados:

$$\textbf{(IV)} \quad (a_4 * C_{234} + a_3 * C_{23} + a_2 * C_2 + d) = \sqrt{\boldsymbol{x_E}^2 + \boldsymbol{y_E}^2}$$

$$\begin{cases} \textbf{(III)} \quad a_3 * S_{23} + a_2 * S_2 = \boldsymbol{z_E} - d_1 - a_4 * S_{234} \\ \textbf{(IV)} \quad a_3 * C_{23} + a_2 * C_2 = \sqrt{\boldsymbol{x_E}^2 + \boldsymbol{y_E}^2} - d - a_4 * C_{234} \end{cases}$$

Dividindo (III) e (IV) ao quadrado:

$$\frac{a_3 * S_{23} + a_2 * S_2}{a_3 * C_{23} + \ a_2 \ * C_2} = \frac{\boldsymbol{z_E} - d_1 - a_4 * S_{234}}{\sqrt{\boldsymbol{x_E}^2 + \boldsymbol{y_E}^2} - d - \ a_4 * C_{234}}$$

$$\frac{a_3 * (S_2 * C_3 + S_3 * C_2) + a_2 * S_2}{a_3 * (C_2 * C_3 - S_2 * S_3) + \ a_2 \ * C_2} = \frac{\boldsymbol{z_E} - d_1 - a_4 * S_{234}}{\sqrt{\boldsymbol{x_E}^2 + \boldsymbol{y_E}^2} - d - \ a_4 * C_{234}}$$

$$\frac{(a_3 * S_3) * C_2 + (a_2 + a_3 * C_3) * S_2}{(a_2 + a_3 * C_3) * C_2 - (a_3 * S_3) * S_2} = \frac{\boldsymbol{z_E} - d_1 - a_4 * S_{234}}{\sqrt{\boldsymbol{x_E}^2 + \boldsymbol{y_E}^2} - d - \ a_4 * C_{234}}$$

Atribuindo:

$$\boldsymbol{R} = \frac{\boldsymbol{z_E} - d_1 - a_4 * S_{234}}{\sqrt{\boldsymbol{x_E}^2 + \boldsymbol{y_E}^2} - d - \ a_4 * C_{234}}$$

$$(a_3 S_3) * C_2 + (a_2 + a_3 C_3) * S_2 = \boldsymbol{R} * [\ (a_2 + a_3 C_3) * C_2 - (a_3 S_3) * S_2\ ]$$

$$(a_3 S_3) * C_2 + (a_2 + a_3 C_3) * S_2 = \boldsymbol{R} * (a_2 + a_3 C_3) * C_2 - \boldsymbol{R} * (a_3 S_3) * S_2$$

Dividindo a equação por $\boldsymbol{R} * (a_2 + a_3 * C_3) * C_2$:

$$\frac{(a_3 S_3) C_2}{\boldsymbol{R}(a_2 + a_3 C_3) C_2} + \frac{(a_2 + a_3 C_3) S_2}{\boldsymbol{R}(a_2 + a_3 C_3) C_2} = \frac{\boldsymbol{R}(a_2 + a_3 C_3) C_2}{\boldsymbol{R}(a_2 + a_3 C_3) C_2} - \frac{\boldsymbol{R}(a_3 S_3) S_2}{\boldsymbol{R}(a_2 + a_3 C_3) C_2}$$

$$\frac{(a_3 * S_3)}{\boldsymbol{R} * (a_2 + a_3 * C_3)} + \frac{S_2}{\boldsymbol{R} * C_2} = \mathbf{1} - \frac{(a_3 * S_3) * S_2}{(a_2 + a_3 * C_3) * C_2}$$

Atribuindo:

$$\boldsymbol{M} = \frac{(a_3 * S_3)}{(a_2 + a_3 * C_3)}$$

$$\frac{1}{\boldsymbol{R}} * \frac{(a_3 * S_3)}{(a_2 + a_3 * C_3)} + \frac{1}{\boldsymbol{R}} * \frac{S_2}{C_2} = \mathbf{1} - \frac{(a_3 * S_3)}{(a_2 + a_3 * C_3)} * \frac{S_2}{C_2}$$

$$\frac{1}{\boldsymbol{R}} * \boldsymbol{M} + \frac{1}{\boldsymbol{R}} * \frac{S_2}{C_2} = \mathbf{1} - \boldsymbol{M} * \frac{S_2}{C_2}$$

$$\frac{1}{\boldsymbol{R}} * \frac{S_2}{C_2} + \boldsymbol{M} * \frac{S_2}{C_2} = \mathbf{1} - \frac{\boldsymbol{M}}{\boldsymbol{R}}$$

$$\left(\frac{1}{\boldsymbol{R}} + \boldsymbol{M}\right) * \frac{S_2}{C_2} = \mathbf{1} - \frac{\boldsymbol{M}}{\boldsymbol{R}}$$

$$\left(\frac{1 + \boldsymbol{RM}}{\boldsymbol{R}}\right) * \frac{S_2}{C_2} = \frac{\boldsymbol{R} - \boldsymbol{M}}{\boldsymbol{R}}$$

$$\frac{S_2}{C_2} = \left(\frac{\boldsymbol{R} - \boldsymbol{M}}{\boldsymbol{R}}\right) * \left(\frac{\boldsymbol{R}}{1 + \boldsymbol{RM}}\right)$$

$$\tan\theta_2 = \frac{\boldsymbol{R} - \boldsymbol{M}}{1 + \boldsymbol{RB}}$$

$$\theta_2 = arctan2\left(\frac{\boldsymbol{R} - \boldsymbol{M}}{1 + \boldsymbol{RM}}\right)$$

Utilizando a fórmula:

$$arctan\left(\frac{\boldsymbol{a} - \boldsymbol{b}}{1 + \boldsymbol{ab}}\right) = \arctan(\boldsymbol{a}) - \arctan(\boldsymbol{b})$$

$$\theta_2 = arctan2(\boldsymbol{R}) - arctan2(\boldsymbol{M})$$

$$\theta_2 = \text{atan2}\left(\frac{z_E - d_1 - \text{a}_4 * S_{234}}{\sqrt{{x_E}^2 + {y_E}^2} - \text{a}_4 * C_{234} - d}\right) - \text{atan2}\left(\frac{a_3 * \sin\theta_3}{a_2 + a_3 * \cos\theta_3}\right)$$

Pelas identidades trigonométricas:

$$cos(-\theta_2 - \theta_3 - \theta_4) = \cos(\theta_2 + \theta_3 + \theta_4) \rightarrow C_{234} = \cos(\emptyset)$$
$$sen(-\theta_2 - \theta_3 - \theta_4) = -\text{sen}(\theta_2 + \theta_3 + \theta_4) \rightarrow S_{234} = -\text{sen}(\emptyset)$$

$$\theta_2 = \text{atan2}\left(\frac{z_E - d_1 + \text{a}_4 * \text{sen}(\emptyset)}{\sqrt{{x_E}^2 + {y_E}^2} - \text{a}_4 * \cos(\emptyset) - d}\right) - \text{atan2}\left(\frac{a_3 * \sin\theta_3}{a_2 + a_3 * \cos\theta_3}\right) \quad \text{Eq.74}$$

3.2.2 Cinemática Inversa com Quatérnios Duais (QD)

A Cinemática Inversa usando Quatérnios Duais:

$$\left(\underline{\boldsymbol{Q}}\,{}^{0}_{1}\right)^{-1} = \begin{bmatrix} \cos(\theta_1/2)\cos(45°) \\ -\cos(\theta_1/2)\,\text{sen}(45°) \\ -\text{sen}(\theta_1/2)\,\text{sen}(45°) \\ -\text{sen}(\theta_1/2)\cos(45°) \\ -(\mathbf{d}/2)\cos(\theta_1/2)\,\text{sen}(45°) - (d_1/2)\,\text{sen}(\theta_1/2)\cos(45°) \\ -(\mathbf{d}/2)\cos(\theta_1/2)\cos(45°) + (d_1/2)\,\text{sen}(\theta_1/2)\,\text{sen}(45°) \\ -(\mathbf{d}/2)\text{sen}(\theta_1/2)\cos(45°) - (d_1/2)\cos(\theta_1/2)\,\text{sen}(45°) \\ +(\mathbf{d}/2)\text{sen}(\theta_1/2)\,\text{sen}(45°) - (d_1/2)\cos(\theta_1/2)\cos(45°) \end{bmatrix} \quad \text{Eq.75}$$

$$\underline{\boldsymbol{Q}}\,{}^{0}_{4} = \begin{bmatrix} \boldsymbol{q_1} \\ \boldsymbol{q_2} \\ \boldsymbol{q_3} \\ \boldsymbol{q_4} \\ \boldsymbol{q_5} \\ \boldsymbol{q_6} \\ \boldsymbol{q_7} \\ \boldsymbol{q_8} \end{bmatrix} \quad \text{Eq.76}$$

$$\underline{\boldsymbol{Q}}\,{}^{1}_{4} = \begin{bmatrix} \mathbf{1} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ +\frac{\mathbf{a_4}}{\mathbf{2}} + \frac{\mathbf{a_3}}{\mathbf{2}}\boldsymbol{C_{23}} + \frac{\mathbf{a_2}}{\mathbf{2}}\boldsymbol{C_2} \\ \frac{\mathbf{a_3}}{\mathbf{2}}\boldsymbol{S_{23}} + \frac{\mathbf{a_2}}{\mathbf{2}}\boldsymbol{S_2} \\ \mathbf{0} \end{bmatrix} \qquad \text{Eq.77}$$

$$\left(\underline{\boldsymbol{Q}}\,{}^{0}_{1}\right)^{-1} \blacksquare\ \underline{\boldsymbol{Q}}\,{}^{0}_{4} = \underline{\boldsymbol{Q}}\,{}^{1}_{4} \qquad \text{Eq.78}$$

$$\begin{bmatrix} \cos(\theta_1/2)\cos(45°) \\ -\cos(\theta_1/2)\,\mathrm{sen}(45°) \\ -\mathrm{sen}(\theta_1/2)\,\mathrm{sen}(45°) \\ -\mathrm{sen}(\theta_1/2)\cos(45°) \\ -\left(\frac{d}{2}\right)\cos(\theta_1/2)\,\mathrm{sen}(45°) - \left(\frac{d_1}{2}\right)\mathrm{sen}(\theta_1/2)\cos(45°) \\ -\left(\frac{d}{2}\right)\cos(\theta_1/2)\cos(45°) + \left(\frac{d_1}{2}\right)\mathrm{sen}(\theta_1/2)\,\mathrm{sen}(45°) \\ -\left(\frac{d}{2}\right)\mathrm{sen}(\theta_1/2)\cos(45°) - \left(\frac{d_1}{2}\right)\cos(\theta_1/2)\,\mathrm{sen}(45°) \\ +\left(\frac{d}{2}\right)\mathrm{sen}(\theta_1/2)\,\mathrm{sen}(45°) - \left(\frac{d_1}{2}\right)\cos(\theta_1/2)\cos(45°) \end{bmatrix} \blacksquare \begin{bmatrix} \boldsymbol{q_1} \\ \boldsymbol{q_2} \\ \boldsymbol{q_3} \\ \boldsymbol{q_4} \\ \boldsymbol{q_5} \\ \boldsymbol{q_6} \\ \boldsymbol{q_7} \\ \boldsymbol{q_8} \end{bmatrix} = \begin{bmatrix} \mathbf{1} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ +\frac{\mathbf{a_4}}{\mathbf{2}} + \frac{\mathbf{a_3}}{\mathbf{2}}\boldsymbol{C_{23}} + \frac{\mathbf{a_2}}{\mathbf{2}}\boldsymbol{C_2} \\ \frac{\mathbf{a_3}}{\mathbf{2}}\boldsymbol{S_{23}} + \frac{\mathbf{a_2}}{\mathbf{2}}\boldsymbol{S_2} \\ \mathbf{0} \end{bmatrix}$$

$$\boldsymbol{transl}\left(\left(\underline{\boldsymbol{Q}}\,{}^{0}_{1}\right)^{-1} \blacksquare\ \underline{\boldsymbol{Q}}\,{}^{0}_{4}\right) = \boldsymbol{transl}\left(\underline{\boldsymbol{Q}}\,{}^{1}_{4}\right) \qquad \text{Eq.79}$$

$$\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ 2*(\boldsymbol{q_1 q_5} + \boldsymbol{q_2 q_6} + \boldsymbol{q_3 q_7} + \boldsymbol{q_4 q_8}) \\ -\left(\boldsymbol{q_1}^2 + \boldsymbol{q_2}^2 + \boldsymbol{q_3}^2 + \boldsymbol{q_4}^2\right) d + \boldsymbol{C_1} * \boldsymbol{x_E} + \boldsymbol{S_1} * \boldsymbol{y_E} \\ -\left(\boldsymbol{q_1}^2 + \boldsymbol{q_2}^2 + \boldsymbol{q_3}^2 + \boldsymbol{q_4}^2\right) d_1 + \boldsymbol{z_E} \\ \boldsymbol{S_1} * \boldsymbol{x_E} - \boldsymbol{C_1} * \boldsymbol{y_E} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ +\mathbf{a_4} + \mathbf{a_3}\boldsymbol{C_{23}} + \mathbf{a_2}\boldsymbol{C_2} \\ \mathbf{a_3}\boldsymbol{S_{23}} + \mathbf{a_2}\boldsymbol{S_2} \\ \mathbf{0} \end{bmatrix}$$

$$\mathrm{sen}\,\theta_1 * \boldsymbol{x_E} - \cos\theta_1 * \boldsymbol{y_E} = 0$$

$$\mathrm{sen}\,\theta_1 * \boldsymbol{x_E} = \cos\theta_1 * \boldsymbol{y_E}$$

$$\theta_1 = arctan2\left(\frac{\boldsymbol{y_E}}{\boldsymbol{x_E}}\right) \qquad \text{Eq.80}$$

$$\left(\underline{\boldsymbol{Q}}\ {}^{0}_{3}\right)^{-1} =$$

$$= \begin{bmatrix} \boldsymbol{C_{(123)}} * \cos(45°) \\ -\cos(\mathbf{B}) * \operatorname{sen}(45°) \\ -\operatorname{sen}(\mathbf{B}) * \operatorname{sen}(45°) \\ -\boldsymbol{S_{(123)}} * \cos(45°) \\ -\operatorname{sen}(45°)\left[\frac{\mathbf{a_3}}{2}\cos(\mathbf{B}) + \frac{\mathbf{a_2}}{2}\cos(\mathbf{C}) + \left(\frac{d}{2}\right)\boldsymbol{C_{(123)}}\right] - \left(\frac{d_1}{2}\right)\cos(45°)\,\boldsymbol{S_{(123)}} \\ -\cos(45°)\left[\frac{\mathbf{a_3}}{2}\boldsymbol{C_{(123)}} + \frac{\mathbf{a_2}}{2}\cos(\mathbf{D}) + \left(\frac{d}{2}\right)\cos(\mathbf{B})\right] + \left(\frac{d_1}{2}\right)\operatorname{sen}(45°)\operatorname{sen}(\mathbf{B}) \\ -\cos(45°)\left[\frac{\mathbf{a_3}}{2}\boldsymbol{S_{(123)}} + \frac{\mathbf{a_2}}{2}\operatorname{sen}(\mathbf{D}) + \left(\frac{d}{2}\right)\operatorname{sen}(\mathbf{B})\right] - \left(\frac{d_1}{2}\right)\operatorname{sen}(45°)\cos(\mathbf{B}) \\ +\operatorname{sen}(45°)\left[\frac{\mathbf{a_3}}{2}\operatorname{sen}(\mathbf{B}) + \frac{\mathbf{a_2}}{2}\operatorname{sen}(\mathbf{C}) + \left(\frac{d}{2}\right)\boldsymbol{S_{(123)}}\right] - \left(\frac{d_1}{2}\right)\cos(45°)\,\boldsymbol{C_{(123)}} \end{bmatrix} \quad \text{Eq.81}$$

$$\underline{\boldsymbol{Q}}\ {}^{3}_{4} = \begin{bmatrix} C_{(\emptyset)}C_{(23)} + S_{(\emptyset)}S_{(23)} \\ 0 \\ 0 \\ S_{(\emptyset)}C_{(23)} - C_{(\emptyset)}S_{(23)} \\ 0 \\ +(\mathbf{a}_4/2)\left(C_{(\emptyset)}C_{(23)} + S_{(\emptyset)}S_{(23)}\right) \\ +(\mathbf{a}_4/2)\left(S_{(\emptyset)}C_{(23)} - C_{(\emptyset)}S_{(23)}\right) \\ 0 \end{bmatrix} \quad \text{Eq.82}$$

$$\left(\underline{\boldsymbol{Q}}\ {}^{0}_{3}\right)^{-1} \blacksquare\ \underline{\boldsymbol{Q}}\ {}^{0}_{4} = \underline{\boldsymbol{Q}}\ {}^{3}_{4} \quad \text{Eq.83}$$

$$\begin{bmatrix} \boldsymbol{C_{(123)}} * \cos(45°) \\ -\cos(\mathbf{B}) * \operatorname{sen}(45°) \\ -\operatorname{sen}(\mathbf{B}) * \operatorname{sen}(45°) \\ -\boldsymbol{S_{(123)}} * \cos(45°) \\ -\operatorname{sen}(45°)\left[\frac{\mathbf{a_3}}{2}\cos(\mathbf{B}) + \frac{\mathbf{a_2}}{2}\cos(\mathbf{C}) + \left(\frac{d}{2}\right)\boldsymbol{C_{(123)}}\right] - \left(\frac{d_1}{2}\right)\cos(45°)\,\boldsymbol{S_{(123)}} \\ -\cos(45°)\left[\frac{\mathbf{a_3}}{2}\boldsymbol{C_{(123)}} + \frac{\mathbf{a_2}}{2}\cos(\mathbf{D}) + \left(\frac{d}{2}\right)\cos(\mathbf{B})\right] + \left(\frac{d_1}{2}\right)\operatorname{sen}(45°)\operatorname{sen}(\mathbf{B}) \\ -\cos(45°)\left[\frac{\mathbf{a_3}}{2}\boldsymbol{S_{(123)}} + \frac{\mathbf{a_2}}{2}\operatorname{sen}(\mathbf{D}) + \left(\frac{d}{2}\right)\operatorname{sen}(\mathbf{B})\right] - \left(\frac{d_1}{2}\right)\operatorname{sen}(45°)\cos(\mathbf{B}) \\ +\operatorname{sen}(45°)\left[\frac{\mathbf{a_3}}{2}\operatorname{sen}(\mathbf{B}) + \frac{\mathbf{a_2}}{2}\operatorname{sen}(\mathbf{C}) + \left(\frac{d}{2}\right)\boldsymbol{S_{(123)}}\right] - \left(\frac{d_1}{2}\right)\cos(45°)\,\boldsymbol{C_{(123)}} \end{bmatrix} \blacksquare \begin{bmatrix} \boldsymbol{q_1} \\ \boldsymbol{q_2} \\ \boldsymbol{q_3} \\ \boldsymbol{q_4} \\ \boldsymbol{q_5} \\ \boldsymbol{q_6} \\ \boldsymbol{q_7} \\ \boldsymbol{q_8} \end{bmatrix} =$$

$$= \begin{bmatrix} C_{(\emptyset)}C_{(23)} + S_{(\emptyset)}S_{(23)} \\ 0 \\ 0 \\ S_{(\emptyset)}C_{(23)} - C_{(\emptyset)}S_{(23)} \\ 0 \\ +(\mathbf{a}_4/2)\left(C_{(\emptyset)}C_{(23)} + S_{(\emptyset)}S_{(23)}\right) \\ +(\mathbf{a}_4/2)\left(S_{(\emptyset)}C_{(23)} - C_{(\emptyset)}S_{(23)}\right) \\ 0 \end{bmatrix}$$

$$\boldsymbol{transl}\left(\left(\underline{\boldsymbol{Q}}\ {}^{0}_{3}\right)^{-1} \blacksquare\ \underline{\boldsymbol{Q}}\ {}^{0}_{4}\right) = \boldsymbol{transl}\left(\underline{\boldsymbol{Q}}\ {}^{3}_{4}\right) \quad \text{Eq.84}$$

$$\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ 2*(\boldsymbol{q_1 q_5}+\boldsymbol{q_2 q_6}+\boldsymbol{q_3 q_7}+\boldsymbol{q_4 q_8}) \\ (C_1\,C_{23})\,\boldsymbol{x_E}+(S_1\,C_{23})\,\boldsymbol{y_E}+(S_{23})\,\boldsymbol{z_E}+(-a_2\,C_3-d\,C_{23}-d_1\;S_{23}-a_3)*(\boldsymbol{q_1}^2+\boldsymbol{q_2}^2+\boldsymbol{q_3}^2+\boldsymbol{q_4}^2) \\ (-C_1S_{23})\boldsymbol{x_E}-S_1S_{23}\boldsymbol{y_E}+(C_{23})\boldsymbol{z_E}+(+a_2S_3+dS_{23}-d_1C_{23})*(\boldsymbol{q_1}^2+\boldsymbol{q_2}^2+\boldsymbol{q_3}^2+\boldsymbol{q_4}^2) \\ \boldsymbol{S_1}*\boldsymbol{x_E}-\boldsymbol{C_1}*\boldsymbol{y_E} \end{bmatrix} =$$

$$= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ +\mathbf{a}_4\,(C_\emptyset C_{23}-S_\emptyset S_{23}) \\ +\mathbf{a}_4(-S_\emptyset C_{23}+C_\emptyset S_{23}) \\ 0 \end{bmatrix}$$

Similar às matrizes de transformação Homogêneas, tem-se:

$$\begin{cases} \textbf{(I)}\ (C_1\,C_{23})\,\boldsymbol{x_E}+(S_1C_{23})\boldsymbol{y_E}+(S_{23})\boldsymbol{z_E}-a_2\,C_3-d\,C_{23}-d_1S_{23}-a_3 = +\mathrm{a}_4\,(C_4) \\ \textbf{(II)}\ (-C_1\,S_{23})\,\boldsymbol{x_E}+(-S_1\,S_{23})\boldsymbol{y_E}+(C_{23})\boldsymbol{z_E}+a_2\,S_3+d*S_{23}-d_1C_{23} = \mathrm{a}_4(S_4) \end{cases}$$

$$C_3 = \frac{(C_1\boldsymbol{x_E}+S_1\,\boldsymbol{y_E}-d-a_4C_\emptyset)^2+(\boldsymbol{z_E}-\mathrm{d}_1+a_4\,S_\emptyset)^2-{a_2}^2-{a_3}^2}{2*a_2*a_3} = \boldsymbol{\beta}$$

$$S_3 = \sqrt{\mathbf{1}-\boldsymbol{\beta}^2}$$

$$\theta_3 = arctan2\left(\frac{\pm\sqrt{\mathbf{1}-\boldsymbol{\beta}^2}}{\boldsymbol{\beta}}\right) \qquad \text{Eq.85}$$

$$\boldsymbol{transl}\left(\underline{\boldsymbol{Q}}\,{}^0_4\right) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ (\,\mathrm{a}_4\,\mathrm{C}_1\,C_{234}+\;\mathrm{a}_3\mathrm{C}_1\mathrm{C}_{23}+\;\mathrm{a}_2\,\mathrm{C}_1\mathrm{C}_2+\mathrm{d}\,\mathrm{C}_1) \\ (\,\mathrm{a}_4\,\mathrm{S}_1\,C_{234}+\;\mathrm{a}_3\mathrm{S}_1\mathrm{C}_{23}+\mathrm{a}_2\mathrm{S}_1\mathrm{C}_2+\mathrm{d}\,\mathrm{S}_1) \\ (\;\mathrm{a}_4\,S_{234}+\;\mathrm{a}_3\,\mathrm{S}_{23}+\mathrm{a}_2*\mathrm{S}_2+\mathrm{d}_1) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ x_E \\ y_E \\ z_E \end{bmatrix} \qquad \text{Eq.86}$$

Similar aos cálculos com matrizes de transformação Homogêneas, tem-se:

$$\begin{cases} \textbf{(I)}\ (\,a_4*C_{234}+\;a_3*C_{23}+\;a_2\;*C_2+d)*C_1 = \boldsymbol{x_E} \\ \textbf{(II)}\ (\,a_4*C_{234}+\;a_3*C_{23}+a_2*C_2+d)*S_1 = \boldsymbol{y_E} \\ \textbf{(III)}\ a_4*S_{234}+\;a_3*S_{23}+a_2*S_2+d_1 = \boldsymbol{z_E} \end{cases}$$

$$\theta_2 = \mathrm{atan2}\left(\frac{z_E-d_1+\;\mathrm{a}_4*\sin(\emptyset)}{\sqrt{{x_E}^2+{y_E}^2}-\;\mathrm{a}_4*\cos(\emptyset)-d}\right)-\mathrm{atan2}\left(\frac{a_3*\sin\theta_3}{a_2+a_3*\cos\theta_3}\right) \qquad \text{Eq.87}$$

A cinemática inversa chega às mesmas equações para os quatro ângulos das juntas, conforme resumido na Tabela 10.

Tabela 10: Equações cinemática inversa

Coordenadas Efetuador	Ângulos das juntas
$\boldsymbol{x_E}$ $\boldsymbol{y_E}$ $\boldsymbol{z_E}$ $\emptyset$	$\boldsymbol{\theta_1} = \arctan2\left(\frac{y_E}{x_E}\right)$ $\boldsymbol{\beta} = \frac{(z_E - d_1 + \mathrm{a}_4\, S_\emptyset)^2 + \left(\sqrt{{x_E}^2 + {y_E}^2} - \mathrm{a}_4 C_\emptyset - d\right)^2 - {a_2}^2 - {a_3}^2}{2 * a_2 * a_3}$ $\boldsymbol{\theta_3} = \arctan2\left(\frac{\pm\sqrt{1 - \boldsymbol{\beta^2}}}{\boldsymbol{\beta}}\right)$ $\boldsymbol{\theta_2} = \mathrm{atan2}\left(\frac{z_E - d_1 + \mathrm{a}_4\ \sin(\emptyset)}{\sqrt{{x_E}^2 + {y_E}^2} - \mathrm{a}_4 \cos(\emptyset) - d}\right) - \mathrm{atan2}\left(\frac{a_3\ \sin\theta_3}{a_2 + a_3\ \cos\theta_3}\right)$ $\boldsymbol{\theta_4} = -\emptyset - \theta_2 - \theta_3$

Utilizando a Tabela 7 Denavit-Hartenberg do manipulador e os dados contido na Tabela 10 anterior, temos os seguintes resultados apresentados na Tabela 11 calculados pelo software Maple para a cinemática inversa do Manipulador.

Tabela 11: Resultados da cinemática inversa usando equações dos ângulos das juntas

Valores de entrada		Valores de Saída
1	$x_E = -4.7325379115490$ $y_E = 26.8395562329541$ $z_E = 17.2554167076794$ $\emptyset = 0°$	$\theta_1 = 100°\ (1.745329252\ rad)$ $\theta_2 = 20°\ (0.3490658504\ rad)$ $\theta_3 = 30°(0.5235987757\ rad)$ $\theta_4 = -\emptyset - \theta_2 - \theta_3 = -50°\ (-0.8726646261\ rad)$
2	$x_E = 0$ $y_E = 24.3714138397228$ $z_E = 20.0519930152759$ $\emptyset = 0°$	$\theta_1 = 90°\ (1.570796327\ rad)$ $\theta_2 = 30°\ (0.5235987757\ rad)$ $\theta_3 = 40°(0.6981317011\ rad)$ $\theta_4 = -\emptyset - \theta_2 - \theta_3 = -70°\ (-1.221730477\ rad)$
3	$x_E = -18.498396780499$ $y_E = 0$ $z_E = 22.2720649576841$ $\emptyset = 0°$	$\theta_1 = 180°\ (3.141592654\ rad)$ $\theta_2 = 45°\ (\ 0.7853981633\ rad)$ $\theta_3 = 60°(1.047197551\ rad)$ $\theta_4 = -\emptyset - \theta_2 - \theta_3 = -105°\ (-1.832595714\ rad)$

4	$x_E = 15.8906082011919$ $y_E = 2.80194296545369$ $z_E = 22.1329583666828$ $\emptyset = 0°$	$\theta_1 = 10° \ (0.1745329252\ rad)$ $\theta_2 = 50° \ (0.8726646261\ rad)$ $\theta_3 = 70° (1.221730477\ rad)$ $\theta_4 = -\emptyset - \theta_2 - \theta_3 = -120° \ (-2.094395103\ rad)$
5	$x_E = 3.78732091962131$ $y_E = 0.66780686115777$ $z_E = 34.4229583767364$ $\emptyset = \mathbf{90}°$	$\theta_1 = 10° \ (0.1745329252\ rad)$ $\theta_2 = 50° \ (0.8726646261\ rad)$ $\theta_3 = 70° (1.221730477\ rad)$ $\theta_4 = -\emptyset - \theta_2 - \theta_3 = -30° \ (-0.5235987761 rad)$

3.4 Adaptações, Testes e Ensaios de Eficiência na Execução de Algoritmos

3.4.1 Contagem de Operações

A contagem de operações presentes na cinemática Direta e Inversa para os dois tipos de implementação MTH e QDU aconteceu através da avaliação de cada equação individualmente conforme Tabela 12. Observa-se que nos QDUs não foram contabilizadas as operações de divisão, uma vez que estas serão a entrada dos valores de cinemática juntamente com a entrada dos parâmetros da tabela Denavit-Hartenberg.

Tabela 12: Comparativo Operações para Cinemática Direta e Inversa MTH e QDU

		Total de Operações		
		Funções Trigonom.	Soma e Subtrações	Multiplic.
1	Sem simplificações MTH m=1 (Eq.35)	4	0	6
2	Resultado obtido para $\boldsymbol{H_1^0}$ (Eq.37)	2*	0	2*
3	Resultado obtido para $\boldsymbol{H_2^1}$ (Eq.38)	2*	0	2*
4	Resultado obtido para $\boldsymbol{H_3^2}$ (Eq.39)	2*	0	2*
5	Resultado obtido para $\boldsymbol{H_4^3}$ (Eq.40)	2*	0	2*
6	Sem simplificações QDU m=1 (Eq.36)	4	4	12
7	Resultado obtido para $\underline{\boldsymbol{Q_1^0}}$ (Eq.50)	2*	4	12
8	Resultado obtido para $\underline{\boldsymbol{Q_2^1}}$ (Eq.51)	2*	0*	2*
9	Resultado obtido para $\underline{\boldsymbol{Q_3^2}}$ (Eq.52)	2*	0*	2*
10	Resultado obtido para $\underline{\boldsymbol{Q_4^3}}$ (Eq.53)	2*	0*	2*
11	Sem simplificações MTH m=2 (Eq.35)	8	48	76
12	Resultado obtido para $\boldsymbol{H_2^0}$ (Eq.41)	4*	3*	9*
13	Sem simplificações QDU m=2 (Eq.36)	8	48	72
14	Resultado obtido para $\underline{\boldsymbol{Q_2^0}}$ (Eq.54)	4*	8*	20*
15	Sem simplificações MTH m=3 (Eq.35)	12	96	146
16	Resultado obtido para $\boldsymbol{H_3^0}$ (Eq.42)	6*	6*	14*
17	Resultado obtido para $\boldsymbol{H_4^1}$ (Eq.67)	4*	2*	4*
18	Sem simplificações QDU m=3 (Eq.36)	12	92	132
19	Resultado obtido para $\underline{\boldsymbol{Q_3^0}}$ (Eq.55)	8*	12*	20*

20	Resultado obtido para $\underline{\boldsymbol{Q}}_{\boldsymbol{4}}^{\boldsymbol{1}}$ (Eq.77)	4*	3*	4*
21	Sem simplificações MTH m=4 (Eq.35)	16	144	216
22	Resultado obtido para $\boldsymbol{H}_{\boldsymbol{4}}^{\boldsymbol{0}}$ (Eq.43)	8*	9*	19*
23	Sem simplificações QDU m=4 (Eq.36)	16	136	192
24	Resultado obtido para $\underline{\boldsymbol{Q}}_{\boldsymbol{4}}^{\boldsymbol{0}}$ (Eq.56)	8*	12*	28*
25	Operações para obter translação x, y e z MTH (Eq.5)	0	0	0
26	Operações para obter translação x, y e z QDU (Eq. 27)	0	9	24
27	Operações para Inversa de uma MTH (Eq.11)	0	6	9
28	Operações para Inversa de um QDU (Eq.34)	0	0	0

Os resultados com asterisco (*) são os resultados cujas simplificações foram possíveis

A Tabela 12 compara as simplificações no número de operações executadas tais como substituições trigonométricas de ângulos conhecidos ($\text{seno}(90°) = 1$ e $\text{cosseno}(90°) = 0$) aliados às equações de identidades trigonométricas fundamentais (ANEXO A) e demais simplificações aritméticas possíveis. As multiplicações por (-1) não foram contabilizadas nesta tabela. Os gráficos exibem os dados da tabela anterior de modo a facilitar a comparação entre MTHs e QDUs. O Gráfico 2 exibe a comparação entre cada uma das transformações da cinemática direta do robô com 4GDL ($m = 4$), já o Gráfico 3 mostra as transformações da cinemática direta que envolvem multiplicação de matrizes e multiplicação quatérnios duais.

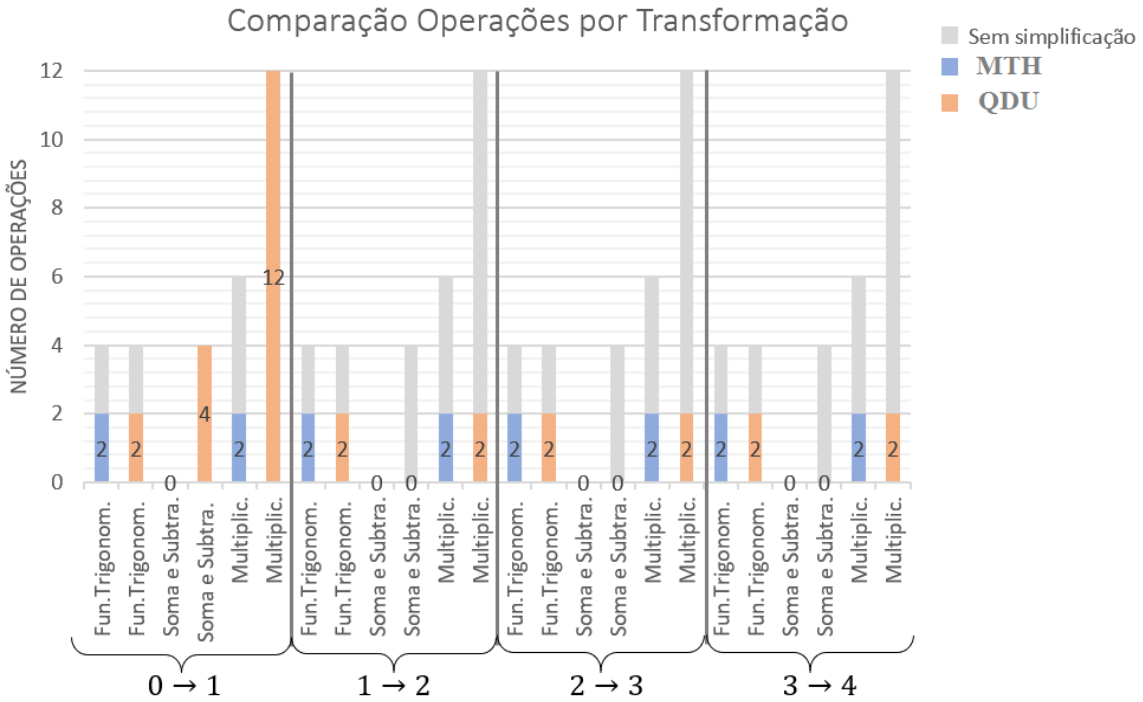


Gráfico 2: Comparação entre cada uma das transformações da cinemática direta do robô com 4GDL ($m = 4$) usando MTH e QDU e confronto com os resultados sem as simplificações.

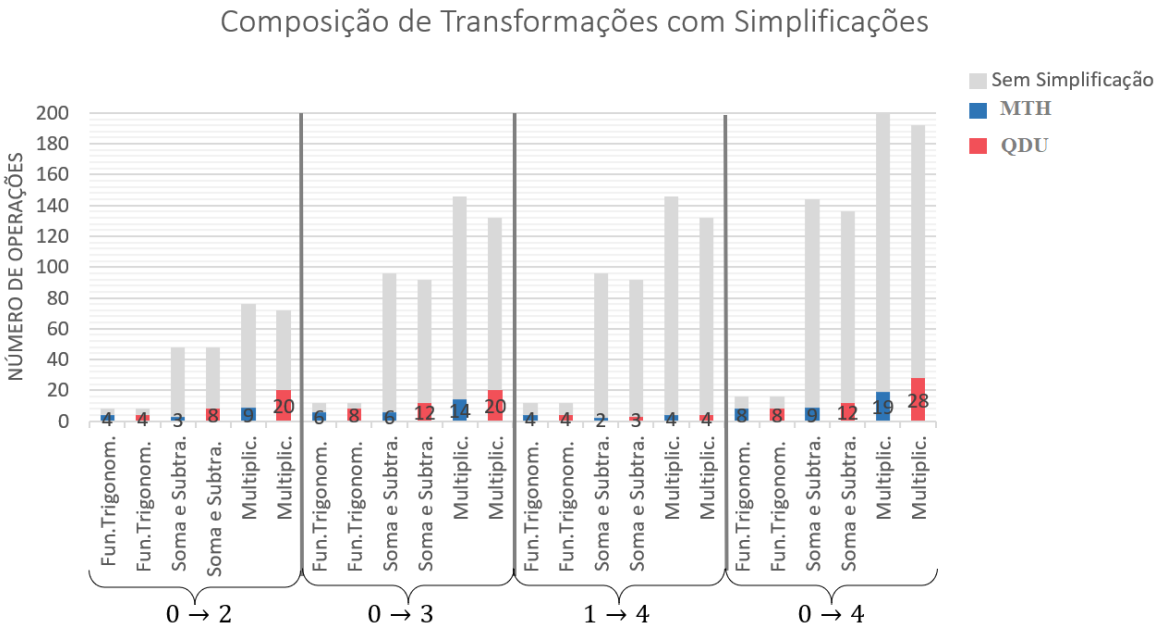


Gráfico 3: Comparação entre composição de duas ou mais transformações do robô com 4GDL ($m = 4$) usando MTH e QDU e confronto com os resultados sem as simplificações.

Os resultados da cinemática Direta e Inversa com simplificações foram dispostos nas Tabelas 13 e 14.

Tabela 13: Contagem de operações aritméticas Cinemática Direta MTH e QDU

Cinemática Direta com Simplificações		Operações		
		Funções Trigonom.	Soma e Subtrações	Multiplic.
MTH	$\boldsymbol{H_4^0}$	8	9	19
QDU	$\underline{\boldsymbol{Q}}_4^0 + \boldsymbol{transl}\left(\underline{\boldsymbol{Q}}\ {}^0_4\right)$	12	21	52

Tabela 14: Contagem de operações aritméticas Cinemática Inversa MTH e QDU

Cinemática Inversa com simplificações		Operações		
		Funções Trigonom.	Soma e Subtrações	Multiplic.
MTH	$(\mathbf{H}\ {}^0_1)^{-1}$, $\mathbf{H}\ {}^1_4$,$(\mathbf{H}\ {}^0_3)^{-1}$	4	14	22
QDU	$\left(\underline{\boldsymbol{Q}}\ {}^0_1\right)^{-1}$, $\underline{\boldsymbol{Q}}\ {}^1_4$, $\left(\underline{\boldsymbol{Q}}\ {}^0_3\right)^{-1}$ $\boldsymbol{transl}\left(\left(\underline{\boldsymbol{Q}}\ {}^0_1\right)^{-1} \blacksquare\ \underline{\boldsymbol{Q}}\ {}^0_4\right)$ $\boldsymbol{transl}\left(\underline{\boldsymbol{Q}}\ {}^1_4\right)$ $\boldsymbol{transl}\left(\left(\underline{\boldsymbol{Q}}\ {}^0_3\right)^{-1} \blacksquare\ \underline{\boldsymbol{Q}}\ {}^0_4\right)$ $\boldsymbol{transl}\left(\underline{\boldsymbol{Q}}\ {}^3_4\right)$	4	39	100

3.4.2 Número de Flop no Microcontrolador

Foi realizada uma análise do tempo gasto pelo microcontrolador em cada uma das operações matemáticas, o código à seguir mostra a implementação no microcontrolador para a variável do tipo byte realizando multiplicações. A Tabela 15 faz um comparativo entre o tempo gasto em cada uma das operações matemática utilizando o microcontrolador Atmel Atmega 328/P.

Tabela 15: Algoritmo para contabilizar o tempo gasto em cada operação no MCU

```
#define TIPO_DA_VARIAVEL float
#define OPERACAO i*j

volatile TIPO_DA_VARIAVEL x, i, j;
unsigned long time;

void setup ()
{
    Serial.begin(115200);
    Serial.println ();
    time = micros();
    for (i = 1; i < 101; i++)
      {
        for (j = 1; j < 101; j++)
          {
            x = OPERACAO;
          }
      }
    time = micros() - time;
    Serial.print("Tempo para completar: ");
    Serial.println(time);
} // fim setup

void loop() {}
```

$\begin{cases} byte, int \\ long, float \end{cases}$

$\begin{cases} i+j \\ i-j \\ i*j \\ i/j \\ sqrt(j) \\ \sin(j) \\ \cos(j) \\ \tan(j) \\ \text{atan2}(j,i) \\ \exp(j) \end{cases}$

Os valores encontrados com o algoritmo anterior foram dispostos na Tabela 16, onde o tipo de entrada representa o tipo de variável (byte, int, long e float) já que o tipo de variável tem influência na memória ocupada dentro dos algoritmos. O tempo foi dividido posteriormente por 100^2 já que dentro dos dois comandos 'for' a operação é executada diversas vezes. Os resultados poderão ser comparados uma vez que mesmo que os comando 'for' e escrita na porta serial consumam tempo na execução do algoritmo, o tempo será o mesmo em todas as operações.

Tabela 16: Tempo gasto para realizar operações com microcontrolador Atmega 328/P

Operação	Tipo de entrada	Tempo μs	$\left(Tempo/100^2\right)$ μs	Linha de Base
Soma	byte	11.432	1,1432	-
	int	20.948	2,0948	-
	long	39.984	3,9984	-
	float	207.996	20,7996	1
Subtração	byte	11.432	1,1432	-
	int	20.948	2,0948	-
	long	39.984	3,9984	-
	float	213.576	21,3576	1,026827

Multiplicação	byte	13.320	1,3320	-
	int	25.976	2,5976	-
	long	83.368	8,3368	-
	float	226.856	22,6856	1,090675
Divisão	byte	61.104	6,1104	-
	int	155.308	15,5308	-
	long	410.376	41,0376	-
	float	439.120	43,9120	2,111194
Raiz Quadrada	byte	391.816	39,1816	-
	int	406.992	40,6992	-
	long	419.736	41,9736	-
	float	442.284	44,2284	2,126406
Seno	byte	1.209.536	120,9536	-
	int	1.224.712	122,4712	-
	long	1.237.456	123,7456	-
	float	1.286.956	128,6956	6,187407
Cosseno	byte	1.203.452	120,3452	-
	int	1.218.624	121,8624	-
	long	1.231.368	123,1368	-
	float	1.280.868	128,0868	6,158138
Tangente	byte	1.494.900	149,4900	-
	int	1.510.076	151,0076	-
	long	1.522.828	152,2828	-
	float	1.556.076	155,6076	7,481278
Arco Tangente 2	byte	1.907.228	190,7228	-
	int	1.924.296	192,4296	-
	long	1.938.928	193,8928	-
	float	1.936.812	193,6812	9,311775
Exponencial	byte	1.901.908	190,1908	-
	int	1.920.064	192,0064	-
	long	1.932.808	193,2808	-
	float	1.944.164	194,4164	9,347122

Os resultados da Tabela 16 foram dispostos em Gráficos para melhor comparação de resultados. O Gráfico 4 exibe o tempo de cada operação utilizando os diferentes tipos de variáveis, observando-se que quanto maior o tamanho da variável, maior o tempo gasto para execução do algoritmo. O Gráfico 5 exibe um comparativo com a última coluna da tabela anterior, nele a operação de menor custo(soma) no MCU foi tida como base de comparação com as demais operações. Os valores como foram comparados com a variável do tipo ‘float’ são por definição os números de flop de cada operação.

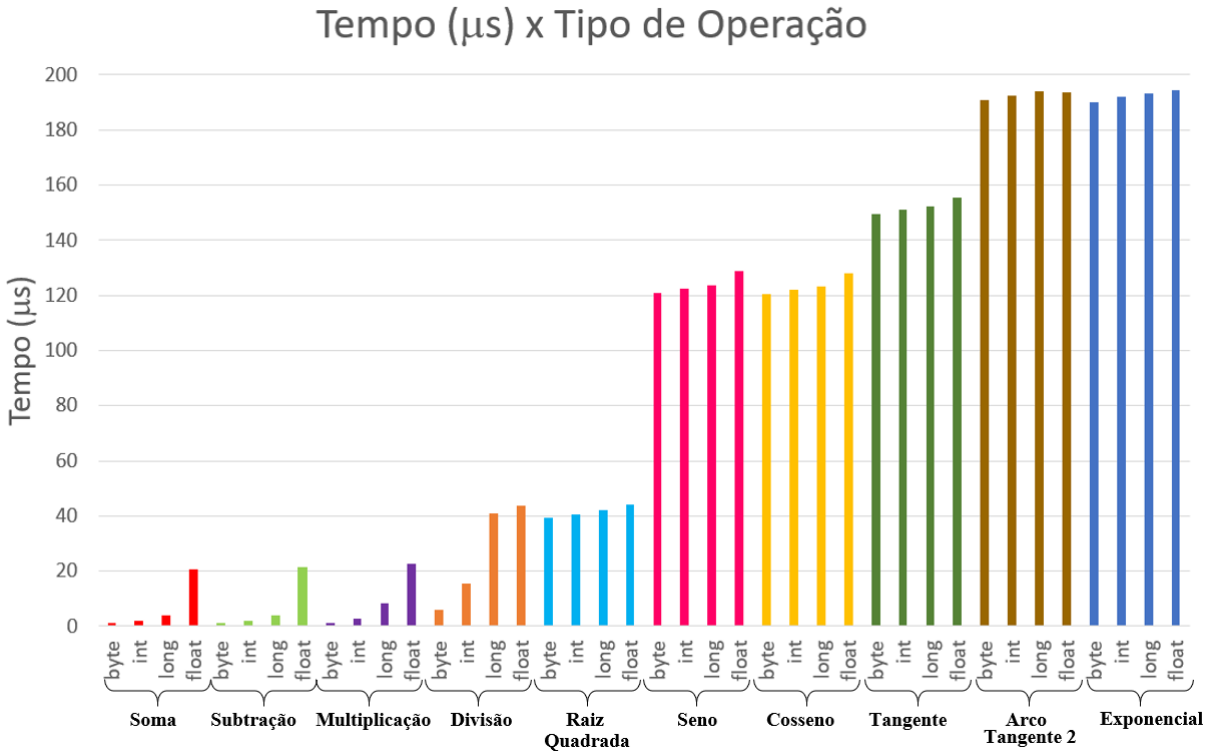


Gráfico 4 : Comparação entre os tempos de operação matemática no MCU

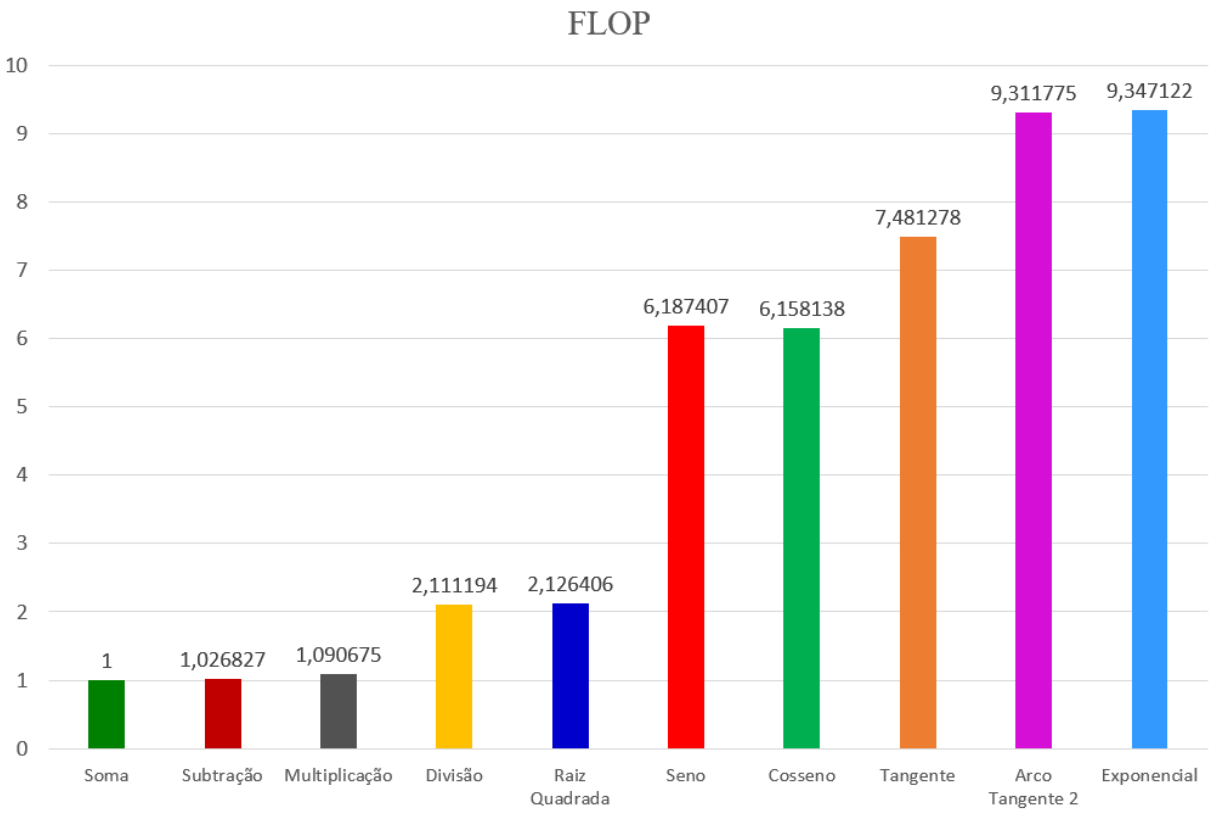


Gráfico 5: Número de Flop para cada operação no MCU

Utilizando de maneira simplificada os dados da Tabela 16 e dos Gráficos 4 e 5 anteriores, cada multiplicação, soma e subtração foi contabilizada como 1 FLOP e as operações trigonométricas foram contabilizadas como 8 FLOP cada (média arredondada entre seno,

cosseno, tangente e arco tangente2). Com essa informação foi possível contabilizar o número total de flop que operações utilizando as MTHs e os QDUs utilizariam ao serem implementados no MCU, os resultados foram exibidos nas Tabelas 17 e 18.

Tabela 17: Número total de flop para Cinemática Direta com MTH e QDU

Cinemática Direta com Simplificações	Operações			FLOP total
	Funções Trigonom.	Soma e Subtrações	Multiplic.	
MTH	8	9	19	92
QDU	12	21	52	169

Tabela 18: Número total de flop para Cinemática Inversa com MTH e QDU

Cinemática Inversa com simplificações	Operações			FLOP total
	Funções Trigonom.	Soma e Subtrações	Multiplic.	
MTH	4	14	22	68
QDU	4	39	100	171

Observa-se que neste teste de número flop, as MTHs superam os QDUs em eficiência temporal, visto que elas gastam menos flop em suas operações de cinemática direta (92 flop para MTH contra 169 flop para QDU) e cinemática inversa (68 flop para MTH contra 171 flop QDU).

3.4.3 Avaliação do tempo por interrupção

Como já mencionado anteriormente, os recursos de interrupção e ‘timers’ foram utilizados na rotina de programação. De modo simplificado, uma interrupção é um desvio da execução das instruções de um programa ocasionados por um evento (neste caso, contagem de tempo). O programa continua executando a função principal ‘main’ para realizar as atividades de movimentação e cálculos de cinemática. Uma segunda parte, denominada de "rotina de timer" deverá conter uma função que periodicamente é chamada pela interrupção para verificar quanto tempo se passou. Ao final, o robô executou uma tarefa específica e o tempo para realizar essa tarefa desde a inicialização do programa é contabilizado. Os programas de Cinemática Direta usando Matriz de transformação Homogênea (Tabela 19) e Cinemática Direta usando Quatérnios Duais (Tabela 20), Cinemática Direta em sua forma compacta (Tabela 21) e Cinemática Inversa forma compacta (Tabela 22) conterão uma função de interrupção. Os respectivos resultados são apresentados nas Figuras 11, 12, 13 e 14. Ao final os tempos foram comparados (Tabelas 23 e 24).

Tabela 19 : Algoritmo MTH interupção

```
#include <TimerOne.h>

unsigned long int cont=0;
/*_____________________________________________________
CINEMÁTICA DIRETA MATRIZ DE TRANSFORMAÇÃO HOMOGÊNEA
                        (MTH)
  _____________________________________________________*/

void setup ()
{
  Serial.begin(115200);
  Serial.println ();
  Timer1.initialize(1);
  Timer1.attachInterrupt(contador);
} // fim setup
/*_____________________________________________________*/
void contador(void)
{
  cont=cont+1;
}

/*_____________________________________________________*/
void loop() {
  MTH(M1,tetha1, d1, a1, alpha1); //Matriz 1
  MTH(M2,tetha2, d2, a2, alpha2); //Matriz 2
  MTH(M3,tetha3, d3, a3, alpha3); //Matriz 3
  MTH(M4,tetha4, d4, a4, alpha4); //Matriz 4
  //multiplica matrizes M1 por M2 e encontra M12:
  mult_matriz(M1, M2, M12);
  //multiplica matrizes M12 por M3 e encontra M13:
  mult_matriz(M12, M3, M13);
  //multiplica matrizes M13 por M4 e encontra M14:
  mult_matriz(M13, M4, M14);
  print_array(M14);// imprime a transform. final
  translacao_mat(M14);
  Serial.print("Contador:");
  Serial.print(cont);
  while (true);
}
```

Interrupção: $1\mu s = 0{,}000001s$

Função Interrupção:

contador

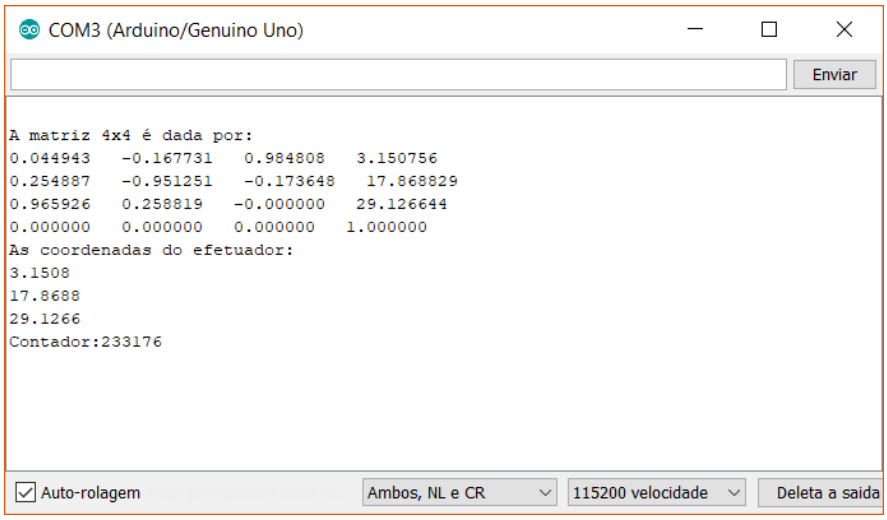


Figura 11: Resultado para algoritmo cinemática direta MTH com interrupção

Tabela 20: Algoritmo QDU interrupção

```
#include <TimerOne.h>

unsigned long int cont=0;
/*
  CINEMÁTICA DIRETA QUATERNIOS DUAIS UNITARIOS
(DQU)
  ____________________________________________*/

void setup ()
{
  Serial.begin(115200);
  Serial.println ();
  Timer1.initialize(1);
  Timer1.attachInterrupt(contador);
} // fim setup
/*_____________________________________________*/
void contador(void)
{
  cont=cont+1;
}
/*_____________________________________________*/
void loop() {
  QDU(Q1,tetha1, d1, a1, alpha1); //Quaternio 1
  QDU(Q2,tetha2, d2, a2, alpha2); //Quaternio 2
  QDU(Q3,tetha3, d3, a3, alpha3); //Quaternio 3
  QDU(Q4,tetha4, d4, a4, alpha4); //Quaternio 4

//multiplica quaternios Q1 por Q2 e encontra Q12:
  mult_quat(Q1, Q2, Q12);

//multiplica quaternios Q12 por Q3 e encontra Q13:
  mult_quat(Q12, Q3, Q13);

//multiplica quaternios Q13 por Q4 e encontra Q14:
  mult_quat(Q13, Q4, Q14);
  print_array(Q14);// imprime a transform. final
  translacao_quat(Q14);
  Serial.print("Contador:");
  Serial.print(cont);
  while (true);
}
```

Interrupção: $1\mu s = 0{,}000001s$

Função Interrupção:

contador

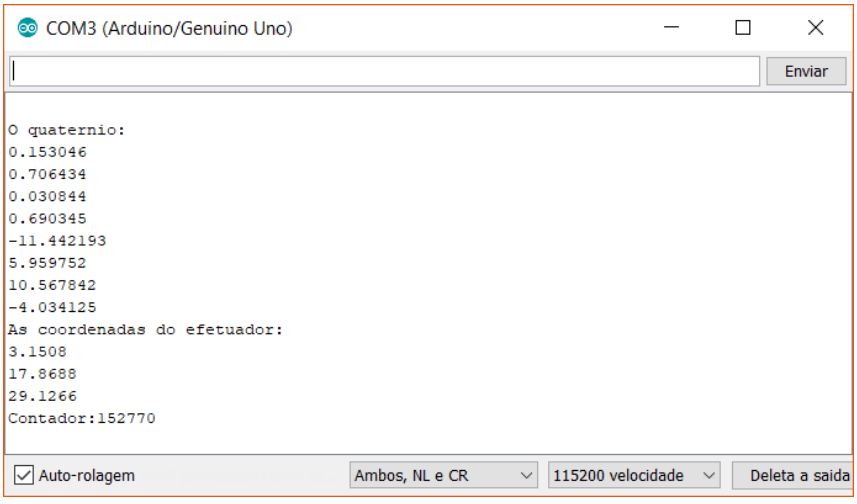


Figura 12: Resultado para algoritmo cinemática direta QDU com interrupção

Tabela 21: Algoritmo Cinemática Direta forma compacta

```
#include <TimerOne.h>
#define pi 3.14159265358979323846264643

unsigned long int cont=0;
float xE,yE, zE, rE;
/*_____________________________________________
  CINEMÁTICA DIRETA
_____________________________________________*/
float tetha1=80*pi/180, tetha2=20*pi/180;
float tetha3=30*pi/180 , tetha4=25*pi/180;
float d1=8.40, d2=0, d3=0, d4=0;
float d=1.13 , a2=9.81 , a3=7.18 , a4=12.29 ;
float alpha1=90*pi/180, alpha2=0*pi/180;
float alpha3=0*pi/180 , alpha4=0*pi/180;

void setup ()
{
  Serial.begin(115200);
  Serial.println ();
  Timer1.initialize(1);
  Timer1.attachInterrupt(contador);
} // fim setup
/*_____________________________________________*/
void Cinematica_direta()
{
  rE= d+a2*cos(tetha2)+a3*cos(tetha2+tetha3)
     +a4*cos(tetha2+tetha3+tetha4);
  xE=rE*cos(tetha1);
  yE=rE*sin(tetha1);
  zE=d1+a2*sin(tetha2)+a3*sin(tetha2+tetha3)
     +a4*sin(tetha2+tetha3+tetha4);

  Serial.println("As coordenadas xE,yE, zE:");
  Serial.println(xE,4);
  Serial.println(yE,4);
  Serial.println(zE,4);
}

/*_____________________________________________*/
void contador(void)
{
  cont=cont+1;
}
/*_____________________________________________*/

void loop() {

  Cinematica_direta();
  Serial.print("Contador:");
  Serial.println(cont);

  while (true); // fica parado
}
```

Interrupção: $1\mu s = 0{,}000001s$

Função Interrupção:

contador

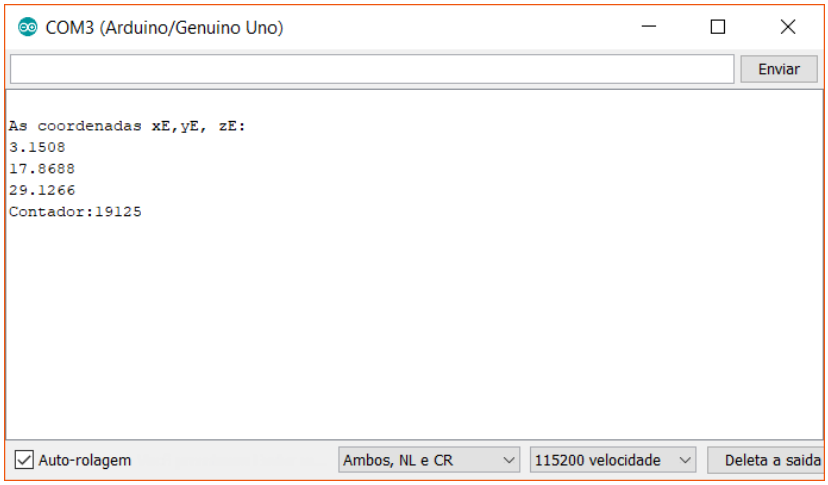


Figura 13: Resultado para algoritmo cinemática direta com interrupção forma compacta

Tabela 22: Algoritmo Cinemática Inversa Forma Compacta

```
include <TimerOne.h>
#define pi 3.14159265358979323846264643

unsigned long int cont=0;
float xE=3.150756875,yE=17.86883020;
float zE=29.12664511;
/*
  CINEMÁTICA INVERSA
                                                    */
float tetha1, tetha2;
float tetha3 , tetha4, fi=-75*pi/180;
float b1, b2, b3, b4, beta;
float d1=8.40, d2=0, d3=0, d4=0;
float d=1.13 , a2=9.81 , a3=7.18 , a4=12.29 ;
float alpha1=90*pi/180, alpha2=0*pi/180;
float alpha3=0*pi/180 , alpha4=0*pi/180;

void setup ()
{
  Serial.begin(115200);
  Serial.println ();
  Timer1.initialize(1);
  Timer1.attachInterrupt(contador);
} // fim setup
/*                                                  */
void Cinematica_inversa()
{
  tetha1=atan2(yE,xE);

  b1=zE-d1+a4*sin(fi);
  b2=sqrt(xE*xE+yE*yE)- a4*cos(fi)-d;
  b3=-a2*a2-a3*a3;
  b4=2*a2*a3;
  beta=(b1*b1+b2*b2+b3)/b4;

  tetha3=atan2(sqrt(1-beta*beta), beta);

  tetha2=atan2(b1,b2)
        -atan2(a3*sin(tetha3), a2+a3*cos(tetha3));
  tetha4= -fi-tetha2-tetha3;
  Serial.println("As coordenadas tetha1,tetha2,
                  tetha3, tetha4:");
```

Interrupção: $1\mu s = 0{,}000001s$

Função Interrupção:

contador

```
  tetha1=tetha1*180/pi;
  tetha2=tetha2*180/pi;
  tetha3=tetha3*180/pi;
  tetha4=tetha4*180/pi;
  Serial.println(tetha1,4);
  Serial.println(tetha2,4);
  Serial.println(tetha3,4);
  Serial.println(tetha4,4);
}

/*_____________________________________________________*/
void contador(void)
{
  cont=cont+1;
}

/*_____________________________________________________*/

void loop() {

  Cinematica_inversa();
  Serial.print("Contador:");
  Serial.println(cont);

  while (true); // fica parado
}
```

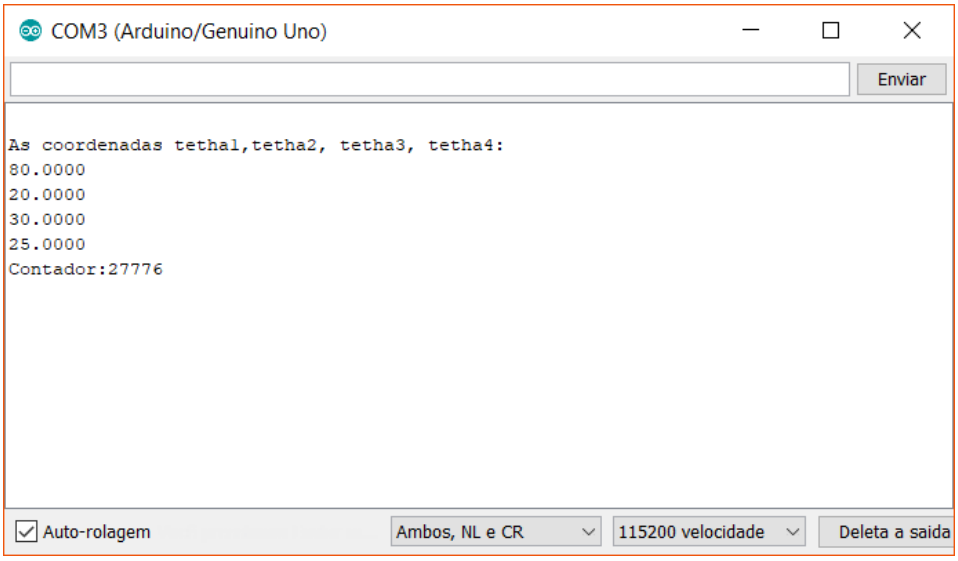


Figura 14: Cinemática Inversa com interrupção forma compacta

Tabela 23: Interrupção aplicada aos algoritmos de cinemática direta MTH e QDU

	Valor do contador	Tempo em segundos
MTH	233.176	0,233s
QDU	152.770	0,152s

Os QDUs são calculados 1,53 vezes mais rápidos do que as MTHs conforme dados mostrados na Tabela 23.

Tabela 24: Interrupção aplicada aos algoritmos de cinemática direta e inversa forma compacta

	Valor do contador	Tempo em segundos
Cinemática Direta	19.125	$0{,}019s$
Cinemática Inversa	27.776	$0{,}027s$

3.4.4 Aplicação do modelo RAM de acesso Máquina

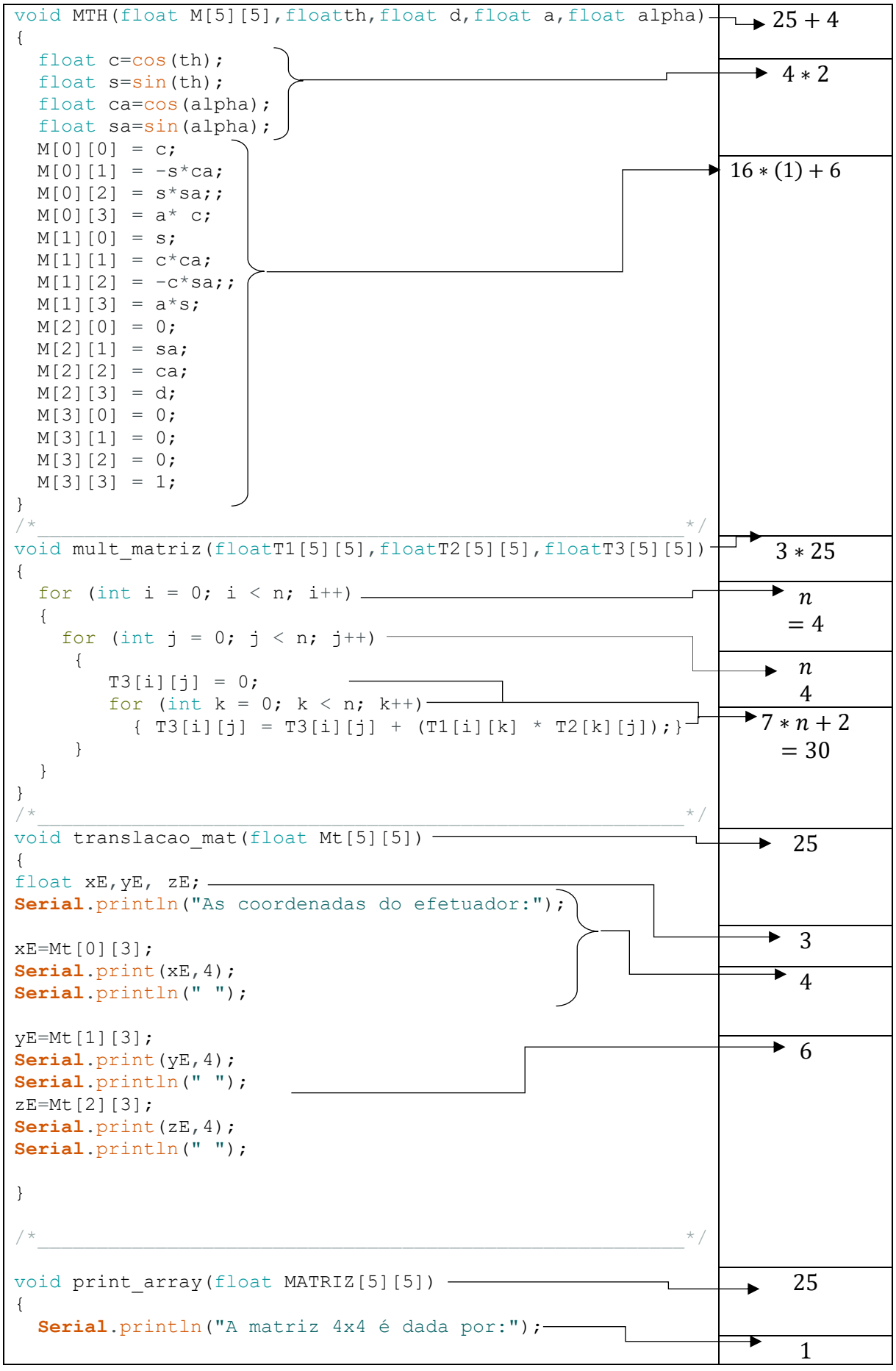
A aplicação do método RAM foi descrita na bibliografia e alguns exemplos de aplicação foram apresentados no Apêndices D. Pelo método RAM, os algoritmos de MTH e QDU foram analisados e a contabilização dos comandos em algoritmos obedecem às regras da literatura conforme a Tabela 25 descrita de maneira simplificada. As Tabelas 26 e 27 mostram os algoritmos analisados para MTH e QDU, respectivamente.

Tabela 25: Contabilização de passos pelo modelo RAM

Comando	Passos
Declarar uma variável de qualquer tipo (int, long, float, double)	1
Atribuir valor à variável (=)	1
Operação simples ($+$, $-$, $*$, $\div$)	1
Comparação ($<$, $>$, $\geq$,$\leq$)	1
Comparação por igualdade ($==$)	1
Escrita de valores das variáveis na tela (*print*)	1
Incremento à uma variável	2
Declarar um array (vetor ou matriz) de tamanho n x m	n x m
Comando 'for' para $i = 0$ até $i < n$	n

Tabela 26: Algoritmo Cinemática direta MTH modelo RAM

	Nº Passos
`/*` `CINEMÁTICA DIRETA MATRIZ DE TRANSFORMAÇÃO HOMOGÊNEA (MTH)` `*/`	
`#define pi 3.14159265358979323846264643`	2
`int n = 4; //tamanho da matriz 4x4 (incluindo posiçao zero)`	2
`// posições matriz [5] --> {0,1, 2, 3, 4}` `float M1[5][5], M2[5][5],M3[5][5], M4[5][5];` `float M12[5][5], M13[5][5], M14[5][5];`	$7 * (25)$
`/*` `*/` `float tetha1=80*pi/180, tetha2=20*pi/180;` `float tetha3=30*pi/180 , tetha4=25*pi/180;` `float d1=8.40, d2=0, d3=0, d4=0;` `float a1=1.13 , a2=9.81 , a3=7.18 , a4=12.29 ;` `float alpha1=90*pi/180, alpha2=0*pi/180;` `float alpha3=0*pi/180 , alpha4=0*pi/180;`	$16 * 2$
`void setup ()` `{` `Serial.begin(115200);` `Serial.println ();` `} // fim setup` `/*` `*/`	1




```
void MTH(float M[5][5],floatth,float d,float a,float alpha)
{
  float c=cos(th);
  float s=sin(th);
  float ca=cos(alpha);
  float sa=sin(alpha);
  M[0][0] = c;
  M[0][1] = -s*ca;
  M[0][2] = s*sa;;
  M[0][3] = a* c;
  M[1][0] = s;
  M[1][1] = c*ca;
  M[1][2] = -c*sa;;
  M[1][3] = a*s;
  M[2][0] = 0;
  M[2][1] = sa;
  M[2][2] = ca;
  M[2][3] = d;
  M[3][0] = 0;
  M[3][1] = 0;
  M[3][2] = 0;
  M[3][3] = 1;
}
/*_____________________________________________________________*/
void mult_matriz(floatT1[5][5],floatT2[5][5],floatT3[5][5])
{
  for (int i = 0; i < n; i++)
  {
    for (int j = 0; j < n; j++)
     {
        T3[i][j] = 0;
        for (int k = 0; k < n; k++)
          { T3[i][j] = T3[i][j] + (T1[i][k] * T2[k][j]);}
     }
  }
}
/*_____________________________________________________________*/
void translacao_mat(float Mt[5][5])
{
float xE,yE, zE;
Serial.println("As coordenadas do efetuador:");

xE=Mt[0][3];
Serial.print(xE,4);
Serial.println(" ");

yE=Mt[1][3];
Serial.print(yE,4);
Serial.println(" ");
zE=Mt[2][3];
Serial.print(zE,4);
Serial.println(" ");

}

/*_____________________________________________________________*/

void print_array(float MATRIZ[5][5])
{
  Serial.println("A matriz 4x4 é dada por:");
```

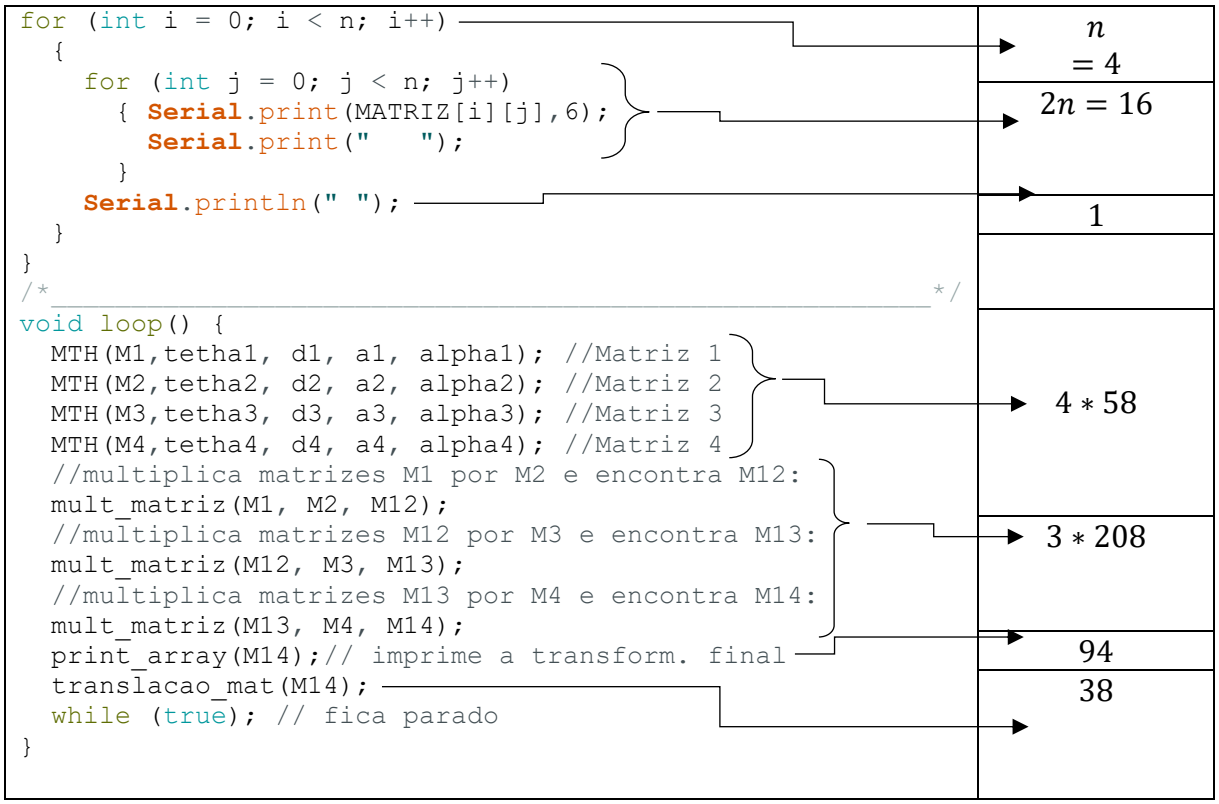
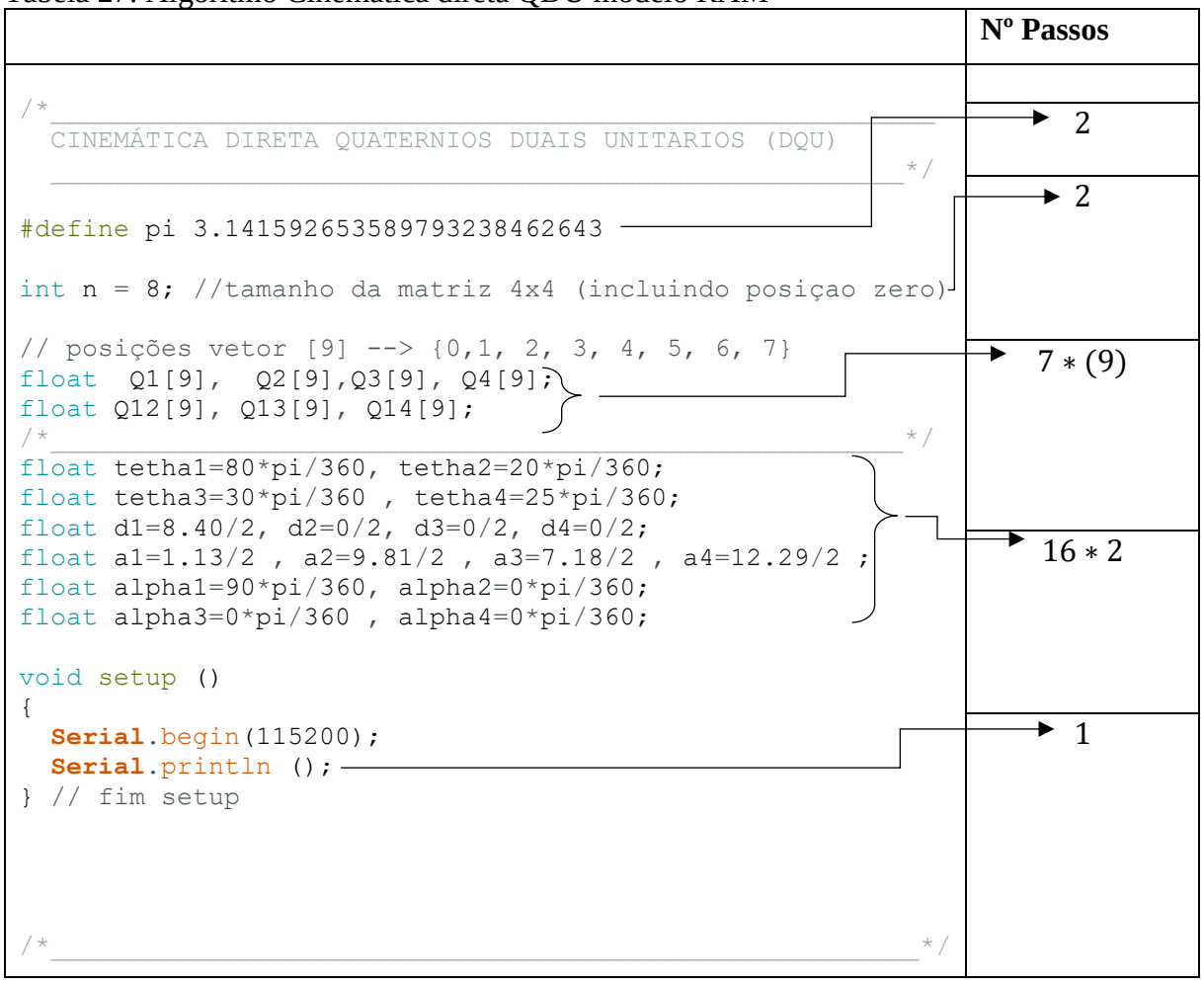
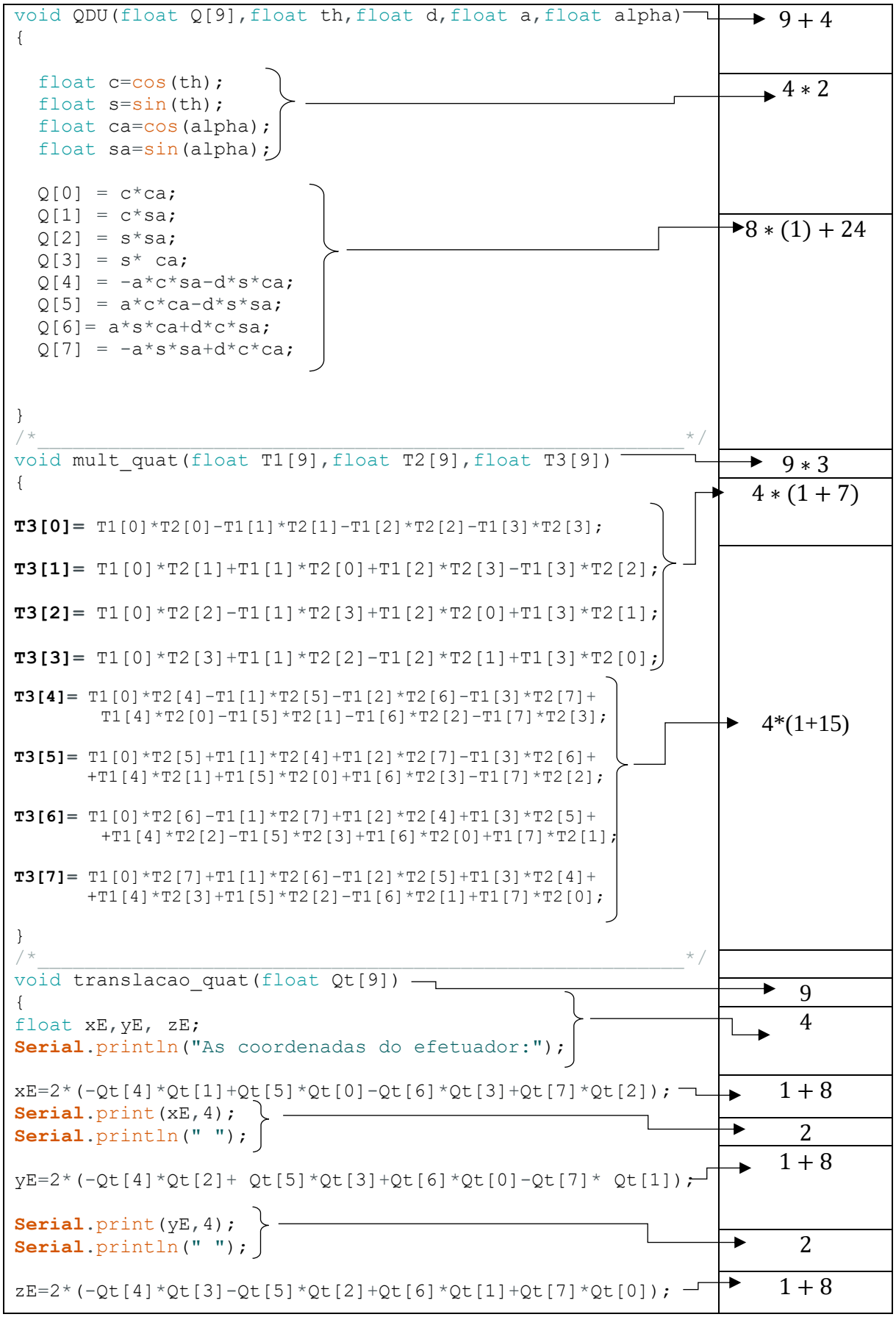
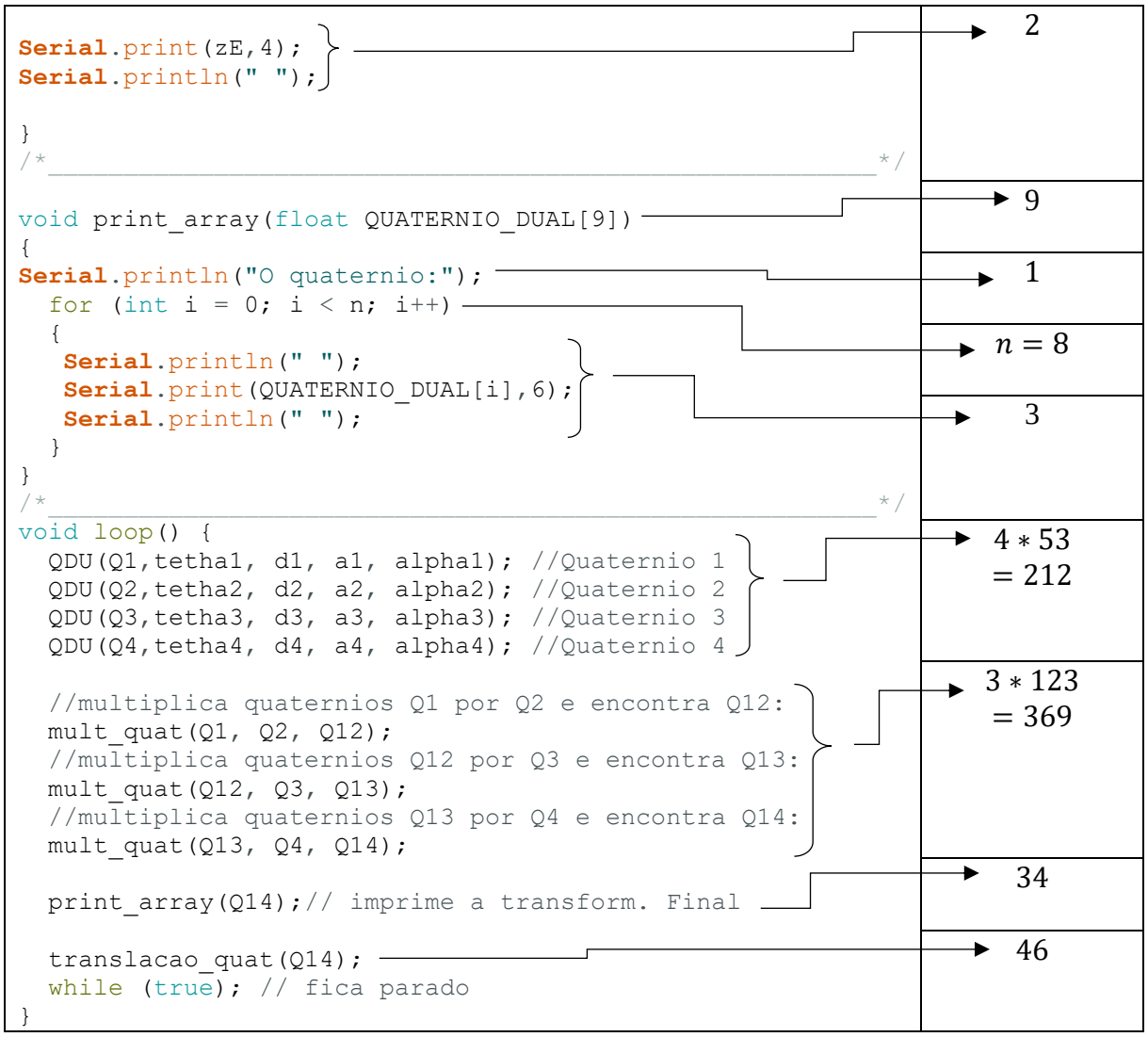


Tabela 27: Algoritmo Cinemática direta QDU modelo RAM






```
void QDU(float Q[9],float th,float d,float a,float alpha)
{

  float c=cos(th);
  float s=sin(th);
  float ca=cos(alpha);
  float sa=sin(alpha);

  Q[0] = c*ca;
  Q[1] = c*sa;
  Q[2] = s*sa;
  Q[3] = s* ca;
  Q[4] = -a*c*sa-d*s*ca;
  Q[5] = a*c*ca-d*s*sa;
  Q[6]= a*s*ca+d*c*sa;
  Q[7] = -a*s*sa+d*c*ca;


}
/*                                                              */
void mult_quat(float T1[9],float T2[9],float T3[9])
{

T3[0]= T1[0]*T2[0]-T1[1]*T2[1]-T1[2]*T2[2]-T1[3]*T2[3];

T3[1]= T1[0]*T2[1]+T1[1]*T2[0]+T1[2]*T2[3]-T1[3]*T2[2];

T3[2]= T1[0]*T2[2]-T1[1]*T2[3]+T1[2]*T2[0]+T1[3]*T2[1];

T3[3]= T1[0]*T2[3]+T1[1]*T2[2]-T1[2]*T2[1]+T1[3]*T2[0];

T3[4]= T1[0]*T2[4]-T1[1]*T2[5]-T1[2]*T2[6]-T1[3]*T2[7]+
        T1[4]*T2[0]-T1[5]*T2[1]-T1[6]*T2[2]-T1[7]*T2[3];

T3[5]= T1[0]*T2[5]+T1[1]*T2[4]+T1[2]*T2[7]-T1[3]*T2[6]+
       +T1[4]*T2[1]+T1[5]*T2[0]+T1[6]*T2[3]-T1[7]*T2[2];

T3[6]= T1[0]*T2[6]-T1[1]*T2[7]+T1[2]*T2[4]+T1[3]*T2[5]+
        +T1[4]*T2[2]-T1[5]*T2[3]+T1[6]*T2[0]+T1[7]*T2[1];

T3[7]= T1[0]*T2[7]+T1[1]*T2[6]-T1[2]*T2[5]+T1[3]*T2[4]+
       +T1[4]*T2[3]+T1[5]*T2[2]-T1[6]*T2[1]+T1[7]*T2[0];

}
/*                                                              */
void translacao_quat(float Qt[9])
{
float xE,yE, zE;
Serial.println("As coordenadas do efetuador:");

xE=2*(-Qt[4]*Qt[1]+Qt[5]*Qt[0]-Qt[6]*Qt[3]+Qt[7]*Qt[2]);
Serial.print(xE,4);
Serial.println(" ");

yE=2*(-Qt[4]*Qt[2]+ Qt[5]*Qt[3]+Qt[6]*Qt[0]-Qt[7]* Qt[1]);

Serial.print(yE,4);
Serial.println(" ");

zE=2*(-Qt[4]*Qt[3]-Qt[5]*Qt[2]+Qt[6]*Qt[1]+Qt[7]*Qt[0]);
```



A Tabela 28 faz a comparação entre as funções utilizadas nos algoritmos MTH e QDU para contagem do número de passos pelo método RAM:

Tabela 28: Comparação entre o número de passos para os algoritmos MTH e QDU

Função	Passos do algoritmo MTH	Passos do algoritmo QDU
Declaração Variáveis início do algoritmo	212	100
Quatro Funções de transformação MTH/QDU	232	212
Três Funções de multiplicação mult_matriz / mult_qua	624	369
Função print_array	94	34
Função para obter a translação final translacao_mat/translação_quat	38	46
TOTAL de passos do algoritmo	**1.200**	**761**

4 ANÁLISE DE RESULTADOS

O cálculo da cinemática direta e inversa fazendo uso do software MAPLE utilizando as Matrizes de Transformação Homogênea e os Quatérnios Duais Unitários chegaram aos mesmos resultados conforme esperado para as duas aplicações.

A cinemática direta utilizando microcontrolador Atmel Atmega328/P para comparação entre Matrizes de Transformação Homogênea e Quatérnios Duais Unitários demonstrou que os QDUs foram superiores nos testes de eficiência temporal utilizando interrupção (sendo 1,53 vezes mais rápidos) e no teste modelo RAM para análise da eficiência de memória dinâmica ocupada, onde os QDUs executaram 1,57 passos a menos do que as MTHs.

O teste para contagem de operações e número de Flop mostrou que os QDUs possuem mais operações do que as MTHs devido a dificuldade em realizar simplificações aritméticas e identidades trigonométricas nos resultados com QDU.

A análise utilizando o próprio software Arduino IDE verificou que as MTHs ocupavam menor memória de armazenamento do programa, o que era esperado já que estas executaram menos operações aritméticas conforme demostrado em primeiro teste com contagem de operações e número de Flop, por outro lado, os QDUs ocuparam menor memória dinâmica do microcontrolador explicando o motivo de os QDUs terem tido um melhor desempenho nos testes de interrupção e modelo RAM.

Os resultados para algoritmos aplicados à cinemática direta apontam que os QDUs ocupam 1,48 vezes mais espaço de armazenamento para o programa (*sketch*) do que as MTHs, em contrapartida o QDU utiliza quase 2 vezes (1,93) menos a memória dinâmica do MCU, conforme mostrado na Tabela 29.

Tabela 29: Algoritmos de cinemática direta utilizando MTH e QDU

	Espaço de Armazenamento para o programa (SKETCH)	Variáveis globais e locais, uso de memória dinâmica
Total Atmega 328/P	32256 bytes	2048 bytes
MTH	4780 bytes (14% do total)	958 bytes (46% do total)
QDU	7088 bytes (21% do total)	496 bytes (24% do total)

Um organograma dos resultados pode ser visualizado na Figura 15, mostrando a ordem de procedimentos da presente pesquisa e como a eficiência de execução dos algoritmos foram confrontadas para chegar à comparação final com os dados presentes no próprio software do MCU.

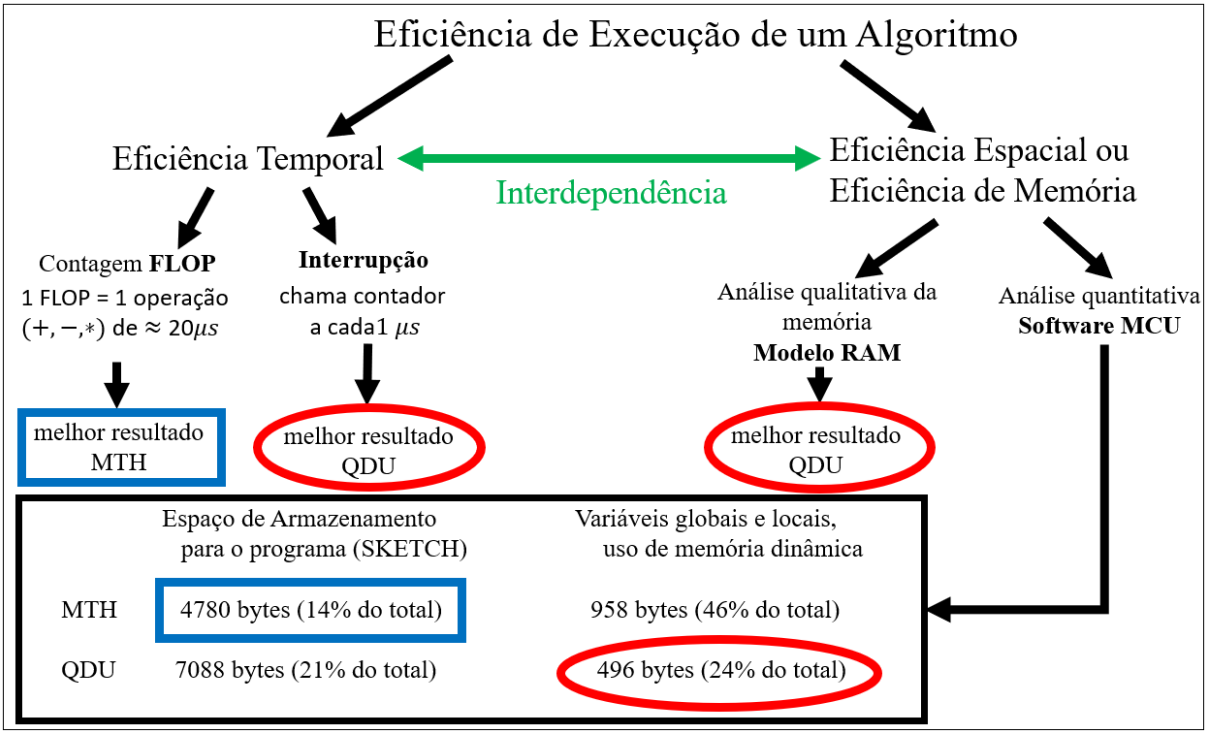


Figura 15: Organograma dos procedimentos e testes realizados

5 CONCLUSÕES

O modelo RAM de acesso máquina foi utilizado para análise da complexidade e comparação entre os algoritmos, apesar de ser um modelo que não está relacionado à análise direta da memória em dispositivos embarcados e, também, por se tratar de um modelo genérico, ele pôde ser usado para uma análise qualitativa da memória dos algoritmos de modo a compreender e interpretar determinados comportamentos do código para efeito de confronto entre os resultados. Esse modelo foi usado em caráter exploratório, portanto não teve o intuito de obter números com resultados precisos, mas através dele foi possível uma indicação de um caminho para tomada de decisão correta onde o QDU se destacou como o algoritmo com maior eficiência de execução. Este modelo qualitativo foi uma ótima alternativa para interpretar melhor os valores quantificados através da implementação no software do próprio MCU.

Quanto ao tempo de execução em análise por intermédio da interrupção, o algoritmo baseado em QDU levou a um programa mais rápido. Este resultado foi de encontro ao resultado modelo RAM.

O trabalho teve por objetivo promover a familiarização com a álgebra quaterniônica. A aplicação desse elemento na robótica é em decorrência do requisito da facilidade de representação da transformação composta da rotação e da translação no espaço, além disso os testes apontavam para a melhor eficiência de execução com os QDUs e neste trabalho foram esclarecidos o motivo de os QDUs receberem cada vez mais atenção em relação às MTHs. A implementação do algoritmo para cálculo da cinemática direta utilizando MTH levou a um programa com menor ocupação da memória de programa em comparação com algoritmo QDU. Contudo, o programa baseado nas MTHs levaram a uma maior ocupação da memória dinâmica e isso se deve ao fato de a representação com MTHs ocuparem um espaço de 4x4 variáveis enquanto que cada QDU ocupa um espaço 8x1. Cabe ao projetista a utilização das MTHs ou dos QDUs dado que em algumas situações a memória de programa possa ser mais relevante do que a memória dinâmica, neste caso, as MTHs é que se destacariam em relação aos QDUs.

Em última análise as cinemáticas direta e inversa foram implementadas no microcontrolador utilizando as equações finais ou equações geométricas do manipulador, o resultado logicamente foi bem mais rápido já que neste caso não utiliza-se de quatérnios e nem de matrizes, tornando o programa mais leve para o MCU. Para uma cadeia cinemática específica, caso seja possível após aplicar simplificações e identidades trigonométricas para chegar às mesmas equações finais as quais foram chamadas de solução compacta é preferível implementar estas soluções ao MCU com o intuito de obter melhor eficiência. Essas soluções quando implementadas levam a uma execução da ordem de 10 vezes mais rápidas, o que significa que para robôs com equações cinemáticas triviais e bem definidas é mais interessante a implementação das equações em sua forma compacta. A justificativa para permanência da discussão e comparação entre QDUs e MTHs se deve ao fato de que estes resultados poderiam ser implementados em qualquer robô do tipo serial.

6 ABSTRACT

The present work proposes a detailed study of the implementation of a mathematical tool called dual quaternions and a depth analysis of its evolution and potential application on the robotics field. In parallel to this view is presented a comparison between the traditional way of calculating rotation and translation in the kinematics of manipulator robots. Dual quaternions have been massively used in robotics because they are computationally more efficient in representing rotational information than the representation with homogeneous transformation matrices. Thus, this work aims to provide a detailed explanation through a step-by-step for the use of quaternion algebra, in a simplified way, were used examples to better understand the implementation. Although there is a lot of literature about the theoretical aspects of dual quaternions, in a few of them there are practical examples of how their use really works. This work gives a clear notion of the introduction to the dual quaternion theory, in addition to it, this document also demonstrates its application to a 4-DOF serial robotic manipulator. In this dissertation it was possible to make a comparison between the direct and inverse kinematics calculated using the matrix algebra versus the quaternionic algebra, it was verified that the quaternions are computationally more efficient although they do not allocate smaller area of the program memory for the Atmel microcontroller Atmega 328/P. Simulations and experimental tests were performed to prove the results.

Keywords: Unit Dual Quaternion, Matrices of Homogeneous Transformations, Direct Kinematics, Inverse Kinematics.

7 REFERÊNCIAS BIBLIOGRÁFICAS

[ADDISON et. al, 1993] ADDISON,Cliff;ALLWRIGHT,James;BINSTED,Norman; BISHOP,Nigel;CARPENTER,Bryan;DALLOZ,Peter;GEE,David;GETOV,Vladimir;HEY,To ny; HOCKNEY,Roger; LEMKE,Max; MERLIN,John; PINCHES,Mark; SCOTT,Chris; WOLTON, Ivan; *"The Genesis Distributed Memory Benchmarks - I Methodology and General Relativity benchmark with results for the SUPRENUM Computer",* John Wiley & Sons, Ltd, Inglaterra, 1993.

[ADORNO, 2011] ADORNO, B. V. Two-arm Manipulation: From Manipulators to Enhanced Human-Robot Collaboration. PhD thesis, Université Montpellier 2, Montpellier. França, 2011.

[ANTON, 2012] Álgebra linear com aplicações; Howard Anton, Chris Rorres ; tradução técnica: Claus Ivo Doering.10 ed. Porto Alegre : Bookman, 2012.

[ATMEL, 2016] ATMEL, Manual de Instruções Microcontrolador Atmel 328/P; "Atmel-42735B-328/P Datasheet Summary-11/2016", 2016

[ATMEL, 2009] ATMEL, Manual de Instruções Microcontrolador Atmel 16U2; "Microcontroller with 8/16/32K Bytes of ISP Flash and USB Controller - ATmega16U2 Datasheet Summary", 2009. Disponível em:<https://pdf1.alldatasheet.com/datasheet-pdf/view/313554/ATMEL/ATmega16U2.html> Acesso em 01 ago.2018

[BATES, 2008] BATES, Martin P. ; Programming 8-Bit PIC Microcontrollers in C with interactive Hardware Simulation. 2 ed. Reino Unido: Newnes, julho de 2008

[BRANDSTÖTTER,2016] BRANDSTÖTTER, Mathias; Adaptable Serial Manipulators in Modular Design; Instituto de Engenharia de Automação e Controle; Tese de doutorado, Hall in Tirol, 2016

[CADY, 2010] CADY, Fredrick M. Microcontrollers and Microcomputers Principles of software and Hardware Engineering, 2 ed., New York, Oxford University Press, Department of Electrical and Computer Engineering, 2010.

[CHEN, HAN e PENG, 2014] CHEN, Jinbao; HAN,Dong; PENG, Zhuang ; Active Grasping Control of Virtual-Dexterous-Robot Hand with Open Inventor,Hindawi Publishing Corporation Mathematical Problems in Engineering; 2014

[CORMEN et al.,2009] CORMEN, Thomas H.; LEISERSON, Charles E. ;RIVEST, Ronald L., CLIFFORD, Stein; Introduction to Algorithms, 3 ed., Massachusetts Institute of Technology, Inglaterra, 2009

[CRAIG, 2005] CRAIG,John J.;Introduction to Robotics: Mechanics and Control; 3.ed; editora Pearson Education, Estados Unidos, 2005

[CRISP,2004] CRISP, John. Introduction to Microprocessors and Microcontrollers, 2 ed., Estados Unidos:Newnes, 2004

[DOMBRE e KHALIL, 2007] DOMBRE, Etienne; KHALIL, Wisama; Control Systems, Robotics and Manufacturing Series, Robot Manipulators: Modeling, Performance, Analysis and Control; Wiley-ISTE, 2007

[EARL, 2018] EARL, Bill; Memories of an Arduino; Adafruit Learning System, 2018. Disponível em: <learn.adafruit.com/memories-of-an-arduino> Acesso em: 07 ago.2018

[FENG e WAN, 2013] FENG, Xiang; WAN, Wanggen. Dual Quaternion Blending Algorithm and Its Application in Character Animation. TELKOMNIKA, Indonesia., Vol. 11, n.10, p. 5553-5562, October 2013

[FIORE, 2018] FIORE, James M.; "Embedded Controllers Using C and Arduino / 2E"; Versão 2.0.9, 2018

[GE,1998] GE, A. Varshney, J. P. Menon, and C. F. Chang, *"Double quaternions for motion interpolation"* in Proceedings of the ASME Design Engineering Technical Conference, 1998.

[GONNET e BAEZA-YATES] GONNET, G.H. e BAEZA-YATES, R.; Handbook of Algorithms and Data Structures. Addison-Wesley, Reading, Mass., 2 ed. Graham, 1991

[GOUASMI, OUALI e BRAHIM,2012] GOUASMI, Mahmoud; OUALI, Mohammed; BRAHIM, Fernini ; Robot Kinematics Using Dual Quaternions; International Journal of Robotics and Automation (IJRA), Vol. 1, No. 1, pp. 13-30; Blida, Argélia; Março, 2012

[KHALIL e KLEINFINGER, 1986] KHALIL Wisama; KLEINFINGER,J.F; "A new geometric notation for open and closed-loop robots",Laboratorio de Automação de Nantes, NANTES CEDEX - França, Conference Paper, Maio, 1986

[KENWRIGHT,2012] KENWRIGHT, "*A Beginners Guide to Dual-Quaternions: What They Are , How They Work, and How to Use Them for 3D Character Hierarchies*", The 20th International Conference on Computer Graphics, Visualization and Computer Vision, no. June 26–28, pp. 1–10, 2012

[MEI, TIPPER e XU, 2014] MEI Gang; TIPPER John C.; XU Nengxiong; Numerical Robustness in Geometric Computation: An Expository Summary;No. 6, p.2717-2727; International Journal Applied Mathematics & Information Sciences;2014

[NIKU, 2015] NIKU, Saeed Benjamin; *"Introdução à robótica: análise, controle, aplicações"*; tradução e revisão técnica de Sérgio Gilberto Taboada, Rio de Janeiro : LTC, 2ed, 2015

[OZGURA e MEZOUAR, 2016] OZGURA,Erol; MEZOUAR, Youcef; *Kinematic modeling and control of a robot arm using unit dual quaternions*. Jornal: Robotics and Autonomous Systems; p.66–73, França, 2016

[PARK, 2015] PARK, Frank C.; Parallel Robots; Robotics Laboratory, Seoul National University, Seoul, Korea;J. Baillieul, T. Samad (eds.), Encyclopedia of Systems and Control; Springer-Verlag London, 2015

[RADAVELLI et al., 2012] RADAVELLI, Luiz; SIMONIA, Roberto; PIERI, Edson Roberto de; MARTINS, Daniel. A Comparative Study of the Kinematics of Robots Manipulators by Denavit-Hartenberg and Dual Quaternion. Asociación Argentina Mechanical Computational Vol XXXI, p.2833-2848, Salta - Argentina, 13-16 November 2012.

[RAFIQUZZAMAN, 2014] RAFIQUZZAMAN, M., Fundamentals of Digital Logic and Microcontrollers, 6 ed., Nova Jersey e Canadá:John Wiley & Sons, 2014

[ROMANO, 2002] ROMANO, Vitor Ferreira; "Robótica Industrial:Aplicação na Indústria de Manufatura e de Processos"; Editora Edgard Blucher, 1ed., 2002

[SCHILLING, 2003] SCHILLING, Robert J.; Fundamentals of Robotics:Analysis and Control;Prentice-Hall of India, Universidade Clarkson, New Delhi, 5ed. 2003

[SCHILLING, 2011] SCHILLING, "Universally manipulable body models— dual quaternion representations in layered and dynamic MMCs", Autonomous Robots, vol. 30, no. 4, pp. 399–

425, 2011

[SHOEMAKE,1985] SHOEMAKE, “Animating rotation with quaternion curves” in Proceedings of the 12th annual conference on Computer graphics and interactive techniques. ACM Press, pp. 245–254, 1985.

[SICILIANO et al.,2009] SICILIANO, Bruno; SCIAVICCO, Lorenzo; VILLANI, Luigi; ORIOLO, Giuseppe; “Robotics:Modelling, Planning and Control”; Springer Handbook of Robotics. Napoli and Stanford, 2009

[SKIENA,2008] SKIENA, Steven S. ; The Algorithm Design Manual, 2 ed.; Springer; Londres, 2008

[SPONG et al., 2004] SPONG, M. W., Hutchinson, S., & Vidyasagar, M.. *Robot Dynamics and Control*,. 2 edition. January 28, 2004

[SPONG et al., 2005] SPONG, M. W., Hutchinson, S., & Vidyasagar, M. (2005). *Robot Modeling and Control*. 1 edition. van Zutven, P. W. M., Modeling stability and identification of humanoid robots. Master’s thesis, Eindhoven University of Technology, Eindhoven., 2005.

[UEBERHUBER, 1995] UEBERHUBER, Christoph W.; “Numerical Computation 1: Methods, Software, and Analysis”; Springer, Berlin Heidelberg, 1995

[WAGNER e DIACONESCU, 2015] WAGNER, Gerd; DIACONESCU, Mircea; *“Optimize Arduino Memory Usage”* ; Alemanha,2015. Disponível em: <https://www.codeproject.com/Articles/1013667/Optimize-Arduino-Memory-Usage> Acesso em: 07 ago.2018

[WANG e YU,2011] WANG, X.; YU, C. ; Unit-Dual-Quaternion-Based PID Control Scheme for Rigid-Body Transformation; 18º World Congress The International Federation of Automatic Control, Milano, Itália; 2011

[YAVUZ,2009] YAVUZ, Sırma Ç; Kinematic Analysis For Robot Arm; Yildiz Technical University, Grubo Coşkun Yetim; Istanbul,2009

[ZIVIANI, 1999] ZIVIANI, Nivio; Projeto de algoritmos com implementações em Pascal e C, 4 ed.,Pioneira Informática, São Paulo, 1999

Data Types Arduino, n.p., Last Update 15 nov.2017; Disponível em: <www.arduino.cc/reference/en/language/variables/data-types> Acesso em: 07 ago.2018

Transformações Denavit-Hartenberg Disponível em: <commons.wikimedia.org/wiki/File:Denavit-Hartenberg-Transformation> Acesso em: 06 ago.2018

ANEXO

Anexo A – Identidades Trigonométricas Fundamentais

1. $\text{sen}(\boldsymbol{A}+\boldsymbol{B}) = \text{sen}(\boldsymbol{A}) * \cos(\boldsymbol{B}) + \text{sen}(\boldsymbol{B}) * \cos(\boldsymbol{A})$
2. $\text{sen}(\boldsymbol{A}-\boldsymbol{B}) = \text{sen}(\boldsymbol{A}) * \cos(\boldsymbol{B}) - \text{sen}(\boldsymbol{B}) * \cos(\boldsymbol{A})$
3. $\cos(\boldsymbol{A}+\boldsymbol{B}) = \cos(\boldsymbol{A}) * \cos(\boldsymbol{B}) - \text{sen}(\boldsymbol{B}) * \text{sen}(\boldsymbol{A})$
4. $\cos(\boldsymbol{A}-\boldsymbol{B}) = \cos(\boldsymbol{A}) * \cos(\boldsymbol{B}) + \text{sen}(\boldsymbol{B}) * \text{sen}(\boldsymbol{A})$
5. $\cos^2(\boldsymbol{A}) + \text{sen}^2(\boldsymbol{A}) = 1$
6. $\cos^2(\boldsymbol{A}) - \text{sen}^2(\boldsymbol{A}) = \cos(\boldsymbol{2A})$
7. $2 * \text{sen}(\boldsymbol{A}) * \cos(\boldsymbol{B}) = \text{sen}(\boldsymbol{A}+\boldsymbol{B}) + \text{sen}(\boldsymbol{A}-\boldsymbol{B})$
8. $2 * \text{sen}(\boldsymbol{A}) * \cos(\boldsymbol{A}) = \text{sen}(\boldsymbol{2A})$
9. $\cos(-\boldsymbol{A}) = \cos(\boldsymbol{A})$
10. $\text{sen}(-\boldsymbol{A}) = -\text{sen}(\boldsymbol{A})$
11. $\tan(-\boldsymbol{A}) = -\tan(\boldsymbol{A})$
12. $\tan(\boldsymbol{A}+\boldsymbol{B}) = \frac{\tan(\boldsymbol{A})+\tan(\boldsymbol{B})}{\boldsymbol{1}-\tan(\boldsymbol{A})*\tan(\boldsymbol{B})}$
13. $\tan(\boldsymbol{A}-\boldsymbol{B}) = \frac{\tan(\boldsymbol{A})-\tan(\boldsymbol{B})}{\boldsymbol{1}+\tan(\boldsymbol{A})*\tan(\boldsymbol{B})}$

Anexo B – Outras discussões Modelo RAM-Máquina de Acesso Aleatório

A análise de complexidade determina qual o custo desse algoritmo na busca da solução de um problema. Complexidade de um algoritmo (ou complexidade computacional) é a taxa de crescimento dos recursos que um algoritmo requer com relação ao tamanho dos dados que ele processa. Tipicamente, os recursos considerados são o tempo (principalmente) e o espaço de armazenamento despendidos pelo algoritmo para processar os dados de entrada. A complexidade de um algoritmo é também denominada como custo de processamento. O modelo RAM é um modelo genérico para a análise de algoritmos, com ele é possível avaliar a eficiência de um algoritmo de modo independente de hardware e software; ou seja, um algoritmo sob análise sequer precisa ser implementado. O modelo prevê instruções comumente encontradas em computadores reais: operações aritméticas (como adição, subtração, multiplicação, divisão, resto, etc), dados em movimento (carregar, armazenar, copiar) e controle (ramificação condicional e incondicional, chamada de sub-rotina e retorno). A eficiência de execução de um algoritmo com uma entrada específica depende do número de operações, “passos” executados. É conveniente definir a noção de passo que é um valor tão independente da máquina quanto possível. Uma quantidade constante de tempo é necessária para executar cada linha do código, uma linha pode levar um tempo diferente de outra linha dependendo do número de operações presentes em cada uma delas. Cada uma dessas instruções leva uma quantidade de tempo constante. A análise de complexidade com este modelo é bem simples e geralmente este é um excelente preditor de desempenho em máquinas reais, o complicador do uso deste modelo (ou o que dificultaria seu uso) seria sua aplicação em ferramentas matemáticas tais como análise combinatória, teoria da probabilidade e outras que exijam capacidade do programador de identificar os termos mais significativos em uma fórmula matemática (CORMEN et al.,2009, p.23-24).

Segundo Ziviani (1999, p.6), a medida do custo de execução de um algoritmo depende principalmente do tamanho da entrada dos dados. Por isso é comum considerar o tempo de execução de um programa como uma função do tamanho da entrada. No entanto, é preciso analisar caso a caso, pois existem alguns algoritmos específicos que o custo de execução é uma função da entrada particular dos dados e não apenas do tamanho da entrada como é o

caso de algoritmos para a ordenação de n números (se os dados de entrada já estiverem quase ordenados então o algoritmo pode ter que trabalhar menos). Ziviani (1999, p.12) também explica que a eficiência de execução de um algoritmo será representada por uma função de custo T(n) que representa o número de passos necessários para executar um algoritmo de um problema de tamanho n. É importante enfatizar que T(n) não representa diretamente o tempo de execução, mas o número de vezes que operações relevantes que são executadas.

Também chamado de Registro Geral Máquina (do inglês 'General Register Machine'), o modelo RAM se trata de um modelo padrão para os computadores convencionais (máquinas de processador único). Considera-se que uma unidade central de processamento (CPU) armazena tanto as operações antes quanto os resultados intermediários após uma operação ser realizada, além disso, o modelo pressupõe memória ilimitada compreendendo um número ilimitado de células de memória que pode conter números de qualquer tamanho. Neste modelo, um algoritmo consiste em instruções que são contabilizadas como passos (do inglês 'step'), relativas por exemplo, à adição de um número, alteração no valor atual da variável, a transferência do conteúdo da variável para a memória, carregar um número da memória para a variável, etc. Por simplicidade, o modelo assume a máquina de acesso aleatório executa exatamente uma instrução por etapa computacional e que todas as instruções levam o mesmo tempo, não importa que tipo de operação esteja envolvida (multiplicação, soma, subtração, divisão) ou quão grande as variáveis sejam (byte, int, long, float). Outra consideração sobre o modelo é que as instruções do programa serão executadas sequencialmente mesmo que existam no algoritmo condições de ramificação (por exemplo um comando 'if/else' de condicional). Uma vez que a unidade de tarefa tenha sido especificada para um algoritmo em particular, o número de operações necessárias para completar o algoritmo de um problema de tamanho n pode ser determinado (contado). O índice de complexidade do algoritmo é, neste caso, atribuído ao número de operações requeridas para cada etapa de problema, assume-se que todas as operações demoram igualmente, independentemente dos operandos e do tipo específico de operação. Desta forma, operações igualmente ponderadas são simplesmente contadas e como resultado desta simplificação o custo de execução teórico pode ser usado apenas para uma avaliação qualitativa de como a memória está sendo acessada no algoritmo ou para avaliações quantitativas grosseiras do tempo de execução para o efeito de uma comparação simples entre algoritmos (UEBERHUBER, 1995, p.183-185).

O modelo RAM quando usado para medir o tempo de execução de um algoritmo assume que o algoritmo executa um determinado número de passos por segundo, essa contagem de operações é, então, convertida naturalmente para o tempo de execução real do algoritmo. O modelo é uma simples análise de como os computadores funcionam. Afinal, multiplicar dois números leva mais tempo do que adicionar dois números na maioria dos processadores, o que viola a primeira suposição do modelo RAM. Além disso, o tipo de problema e como os loops e sob-rotinas se desenrolam devem ser analisados caso a caso. Um outro ponto, diz respeito aos tempos de acesso à memória que certamente diferem muito dependendo de qual memória o algoritmo está acessando. No entanto, apesar de as premissas do modelo nem sempre serem verdadeiras, o modelo continua sendo é excelente para entender como um algoritmo funcionará em um computador real. Atinge uma multa equilíbrio, o comportamento de algoritmos, justamente porque é um modelo simples e muito útil na prática. Para explicar a utilidade deste modelo, tome como exemplo o modelo que a Terra é plana. Algumas pessoas poderão argumentar que este é um modelo ruim, já que está razoavelmente bem estabelecido que a Terra redonda. Mas, por outro lado, quando um engenheiro civil pensa na fundação de uma casa, o modelo da Terra plana é suficientemente preciso e confiável. É muito mais fácil manipular um modelo que pressupõe que Terra plana neste caso, já que seria inconcebível que o engenheiro pensasse em uma forma esférica para a fundação de uma casa. A mesma situação é verdadeira com o modelo de computação em RAM. Trata-se de uma abstração que é geralmente muito

útil. É muito difícil projetar um algoritmo de modo que o modelo de RAM lhe dê resultados muito enganosos. A robustez do modelo RAM permite analisar algoritmos em uma máquina independente das abstrações realizadas (SKIENA, 2008, p.32-33)

Anexo C – Microcontrolador, Memória e Compilação de dados

Um microprocessador é um circuito integrado responsável pelo processamento de dados, é a unidade lógica que executa operações aritméticas com diversos registradores especiais, para este componente funcionar existe a necessidade de outros dispositivos externos que contenham memória de leitura e escrita para então haver o armazenamento de dados e programas, no microprocessador os dados são apenas processados, necessitando de dispositivos periféricos que comuniquem com estes dados para o armazenamento permanente, a conversão, a interface, etc. Por outro lado, uma unidade de comando controlado (MCU) ou microcontrolador é considerado mais do que um microprocessador. Além de todas as funções de um microprocessador, o microcontrolador possui memória RAM, memória ROM, temporizadores, contadores, porta serial, conversores e portas de I/O em um só circuito integrado, ou seja, um microcomputador em um único chip (BATES, 2008, p.1-2).

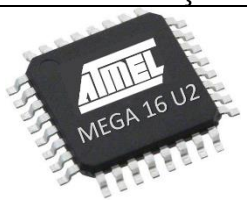
A placa Arduino UNO utilizada no projeto contém dois microcontroladores, sendo o principal deles o modelo Atmega 328/P onde os programas são gravados para posteriormente serem executados. Na mesma placa possui um segundo microcontrolador que é responsável apenas por realizar a comunicação com o computador via USB/Serial. Algumas informações de cada um dos microcontroladores do fabricante Atmel foram coletadas do manual de instruções e apesentados nas Tabelas C1 e C2. Nos microcontroladores da linha Atmega contêm três tipos de memórias EEPROM, FLASH e SRAM, com diferentes aplicações dentro dos microcontroladores (ATMEL,2009 e ATMEL,2016).

Tabela C1: Dados do Microcontrolador Principal utilizado no projeto

Dados do Microcontrolador **ATMEL Atmega 328/P**	
Memória **EEPROM** (não volátil)	1**K** bytes
Memória **FLASH**/ Memória de Programa (não volátil)	32**K** bytes
Memória **SRAM** (volátil)	2**K** bytes

Fonte: ATMEL,2016

Tabela C2: Dados do Microcontrolador Conversor comunicação USB/Serial

Dados do Microcontrolador **ATMEL Atmega 16U2**	
Memória **EEPROM** (não volátil)	512 bytes
Memória **FLASH**/ Memória de Programa (não volátil)	16**K** bytes
Memória **SRAM** (volátil)	512 bytes

Fonte: ATMEL, 2009

Earl(2018, p.7) e Fiore (2018, p.10-12) explicam os três tipos de memórias, suas características e funções dentro do microcontrolador principal (Atmel Atmega328/P) da placa Arduino UNO como:

- Memória EEPROM é uma memória não volátil o que significa que as informações contidas nela continuarão gravadas mesmo quando o sistema for desligado. Os dados nela só podem ser lidos byte-a-byte, e por isso pode ser um pouco complicado usá-la. Essa memória pode sofrer modificações durante a execução do algoritmo, pois pode ser usada para leitura e também gravação durante a execução do programa. Essa memória é mais lenta que a memória SRAM.
- Memória Flash é aquela que possui a mesma tecnologia empregada em pen drives e cartões SD, é chamada pelo fabricante Atmel de memória de programa (*sketch*) e é usada para armazenar o algoritmo em si e quaisquer dos dados (variáveis) que serão utilizados no programa são gravados nesta memória, assim como a memória anterior, esta é uma memória não volátil e mesmo que haja uma queda de energia suas informações continuam inalteradas. O código do programa é executado a partir da memória flash e ela não sofre modificações durante a execução do algoritmo, a modificação desta memória requer alterações no código fonte e nova compilação do programa.
- Memória SRAM (do inglês "Static Random Access Memory") é a memória que pode ser modificada durante a execução do algoritmo, assim como a EEPROM é usada para leitura e escrita de dados a partir da execução do programa, a diferença é que se trata de uma memória volátil e os dados dessa memória são perdidos caso haja queda de energia. É uma memória de grande importância, pois é nela onde se passa todas as ações tais como: alocação estática e dinâmica de variáveis, ponteiros para chamadas de funções, etc. É na memória SRAM que desenvolvedores concentram esforços de otimização, pois é nela onde acontecem a maioria dos problemas como a reinicialização indesejada (do inglês "auto-reset") do microcontrolador e/ou dados corrompidos que exibem resultados imprecisos ou com erros grosseiros retornados pelo MCU.

A Figura C1 mostra de forma esquemática o funcionamento da memória SRAM. Para entende-la melhor é preciso conceituar os três blocos presentes nesta memória: bloco estático (do inglês "Static Data"), bloco de carregamento (do inglês "heap") e bloco de empilhamento (do inglês "Stack"). As funções de cada bloco podem ser descritas como (EARL,2018, p.7-8; WAGNER e DIACONESCU, 2015):

- Bloco Estático ("Static Data"): é um bloco cujo espaço é reservado na SRAM para todas as variáveis globais e estáticas do algoritmo. Para variáveis que contenham valores iniciais previstos no código fonte, durante a execução do algoritmo o sistema simplesmente copia o valor inicial das variáveis que estão na memória Flash para este bloco assim que o

programa é iniciado.

- Bloco de Carregamento (“Heap”): é um bloco para itens e dados alocados dinamicamente. O espaço deste bloco fica logo acima do bloco estático e ele cresce a medida que itens e dados são alocados nele, quanto mais dados a serem alocados dinamicamente maior é a área deste bloco. Embora o uso de alocação de memória dinâmica seja uma boa solução ao programar utilizando um PC que normalmente possui centenas de gigabytes de memória RAM, para o caso de dispositivos embarcados (como os microcontroladores), em geral, é uma má ideia a alocação de memória dinâmica. Isso porque durante a execução do algoritmo essa memória aloca espaço para as variáveis em uso e libera o espaço de outras variáveis, o que resulta na Fragmentação da Memória que consiste na produção de “buracos” de tamanhos diferentes que não podem ser usados em muitos casos para outras variáveis. A medida que esse bloco cresce, a chance de este se chocar com o bloco Stack e corromper os dados executados é maior.
- Bloco de Empilhamento (“Stack”): este bloco serve para variáveis locais e para manter um registro de interrupções e chamadas de função durante a execução do algoritmo. O bloco cresce do topo da memória em direção ao bloco “heap”. Cada interrupção, chamada de função e/ou variável local faz com que este bloco cresça. Ao retornar de uma interrupção ou chamada de função o espaço deste bloco é recuperado para uso em outras funções do algoritmo.

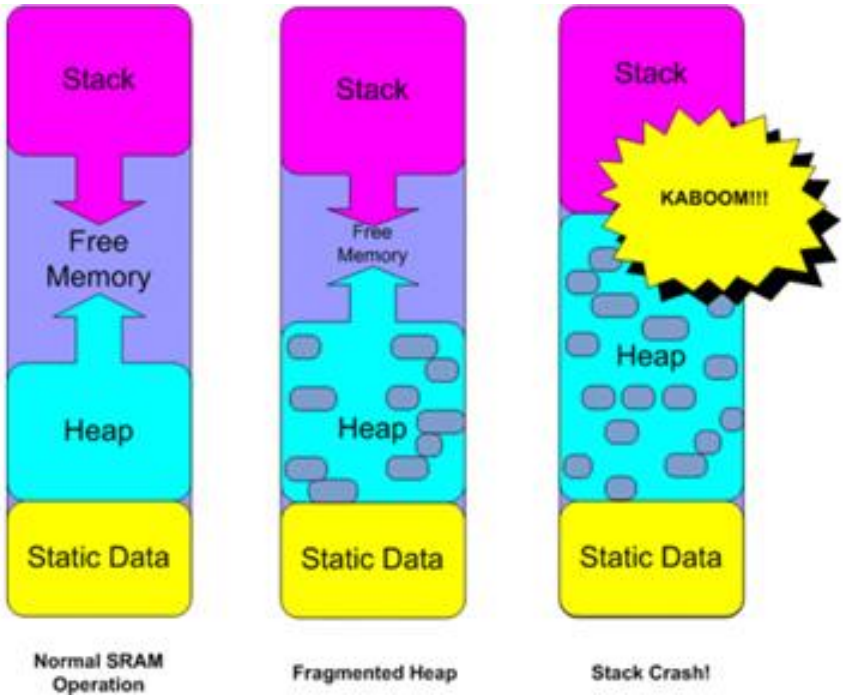


Figura C1: Memória SRAM e seu funcionamento
Fonte: EARL,2018, p.10

O ambiente de desenvolvimento integrado Arduino IDE é um software com múltiplas aplicações, ele inclui um editor de código com recursos de realce de sintaxe, parênteses correspondentes e endentação automática, sendo capaz de compilar e carregar programas para diferentes modelos de placas. Em um código escrito em linguagem C, todas as variáveis que serão utilizadas durante o programa precisam ser declaradas antes de seu primeiro uso no algoritmo (declaradas globalmente ou localmente). É através do tipo de variável que o compilador reserva a quantidade de memória que deve ser utilizada por ela, em outras palavras, quando a variável é declarada separa-se para ela uma quantidade específica de células de

memória necessárias para armazenar o conteúdo da variável. Na declaração é informado o nome da variável, o seu tipo e, opcionalmente, seu valor inicial. Com esta informação, o compilador destina automaticamente o número necessário de células de memória, ou bytes, para armazenar a variável na memória (FIORE, 2018, p.14).

A Tabela C3 contém informações fornecidas pelo próprio fabricante da placa Arduino UNO, é exibido o espaço ocupado por cada tipo de variável em bytes (1 byte = 8bits) e também a extensão do número (termo do inglês “range”) que a variável poderá receber.

Tabela C3: Tipos e tamanhos das variáveis placa Arduino UNO Atmega 328/P

Tipo de Variável	Bytes Usados	Bits Usados	Mínimo	Máximo
byte	1	8	0	255
char	1	8	−128	+127
unsigned char	1	8	0	255
int /short	2	16	−32.768	32.767
unsigned int	2	16	0	65.535
long	4	32	−2.147.483.648 (aprox. -2 bilhões)	2.147.483.647 (aprox. +2 bilhões)
unsigned long	4	32	0	4.294.967.295 (aprox. +4 bilhões)
float	4	32	$-3{,}4 * 10^{38}$	$+3{,}4 * 10^{38}$
double	4	32	$-3{,}4 * 10^{38}$	$+3{,}4 * 10^{38}$

Fonte: www.arduino.cc/reference/en/language/variables/data-types

APÊNDICES

Apêndices A - Álgebra Quaterniônica e Operações elementares com quatérnios duais

A1. Multiplicação Quatérnios Duais por um escalar

$$\lambda * \underline{\boldsymbol{q}} = \lambda * \underline{\boldsymbol{q_p}} + \boldsymbol{\epsilon} * \lambda * \underline{\boldsymbol{q_d}}$$

Exemplo1:

$$\underline{\boldsymbol{q}} = (1 + 1\,\boldsymbol{i} + 1\,\boldsymbol{j} + 1\,\boldsymbol{k}) + \boldsymbol{\epsilon} * (2 + 2\,\boldsymbol{i} + 2\,\boldsymbol{j} + 2\,\boldsymbol{k})$$
$$\lambda = 3$$
$$3 * \underline{\boldsymbol{q}} = 3 * (1 + 1\,\boldsymbol{i} + 1\,\boldsymbol{j} + 1\,\boldsymbol{k}) + \boldsymbol{\epsilon} * 3 * (2 + 2\,\boldsymbol{i} + 2\,\boldsymbol{j} + 2\,\boldsymbol{k})$$
$$3 * \underline{\boldsymbol{q}} = (3 + 3\,\boldsymbol{i} + 3\,\boldsymbol{j} + 3\,\boldsymbol{k}) + \boldsymbol{\epsilon} * (6 + 6\,\boldsymbol{i} + 6\,\boldsymbol{j} + 6\,\boldsymbol{k})$$

A2. Adição de Quatérnios Duais

$$\underline{\boldsymbol{q_1}} + \underline{\boldsymbol{q_2}} = (\underline{\boldsymbol{q_{p1}}} + \underline{\boldsymbol{q_{p2}}}) + \boldsymbol{\epsilon} * (\underline{\boldsymbol{q_{d1}}} + \underline{\boldsymbol{q_{d2}}})$$

Exemplo2:

$$\underline{\boldsymbol{q_1}} = (1 + 1\,\boldsymbol{i} + 1\,\boldsymbol{j} + 1\,\boldsymbol{k}) + \boldsymbol{\epsilon} * (2 + 2\,\boldsymbol{i} + 2\,\boldsymbol{j} + 2\,\boldsymbol{k})$$
$$\underline{\boldsymbol{q_2}} = (3 + 3\,\boldsymbol{i} + 3\,\boldsymbol{j} + 3\,\boldsymbol{k}) + \boldsymbol{\epsilon} * (5 + 5\,\boldsymbol{i} + 5\,\boldsymbol{j} + 5\,\boldsymbol{k})$$
$$\underline{\boldsymbol{q_1}} + \underline{\boldsymbol{q_2}} = (4 + 4\,\boldsymbol{i} + 4\,\boldsymbol{j} + 4\,\boldsymbol{k}) + \boldsymbol{\epsilon} * (7 + 7\,\boldsymbol{i} + 7\,\boldsymbol{j} + 7\,\boldsymbol{k})$$

A3. Subtração de Quatérnios Duais

$$\underline{\boldsymbol{q_1}} - \underline{\boldsymbol{q_2}} = (\underline{\boldsymbol{q_{p1}}} - \underline{\boldsymbol{q_{p2}}}) + \boldsymbol{\epsilon} * (\underline{\boldsymbol{q_{d1}}} - \underline{\boldsymbol{q_{d2}}})$$

Exemplo3:

$$\underline{\boldsymbol{q_1}} = (8 + 8\,\boldsymbol{i} + 8\,\boldsymbol{j} + 8\,\boldsymbol{k}) + \boldsymbol{\epsilon} * (9 + 9\,\boldsymbol{i} + 9\,\boldsymbol{j} + 9\,\boldsymbol{k})$$
$$\underline{\boldsymbol{q_2}} = (3 + 3\,\boldsymbol{i} + 3\,\boldsymbol{j} + 3\,\boldsymbol{k}) + \boldsymbol{\epsilon} * (5 + 5\,\boldsymbol{i} + 5\,\boldsymbol{j} + 5\,\boldsymbol{k})$$
$$\underline{\boldsymbol{q_1}} - \underline{\boldsymbol{q_2}} = (5 + 5\,\boldsymbol{i} + 5\,\boldsymbol{j} + 5\,\boldsymbol{k}) + \boldsymbol{\epsilon} * (4 + 4\,\boldsymbol{i} + 4\,\boldsymbol{j} + 4\,\boldsymbol{k})$$

A4. Conjugado de um Quatérnio Dual

$$\underline{\boldsymbol{q}}^* = (\underline{\boldsymbol{q_p}}^*) + \boldsymbol{\epsilon} * (\underline{\boldsymbol{q_d}}^*)$$

Exemplo4:

$$\underline{\boldsymbol{q}} = (+1 + 1\,\boldsymbol{i} + 1\,\boldsymbol{j} + 1\,\boldsymbol{k}) + \boldsymbol{\epsilon} * (+2 + 2\,\boldsymbol{i} + 2\,\boldsymbol{j} + 2\,\boldsymbol{k})$$
$$\underline{\boldsymbol{q}}^* = (+1 - 1\,\boldsymbol{i} - 1\,\boldsymbol{j} - 1\,\boldsymbol{k}) + \boldsymbol{\epsilon} * (+2 - 2\,\boldsymbol{i} - 2\,\boldsymbol{j} - 2\,\boldsymbol{k})$$

A5. Multiplicação de dois Quatérnios Duais

$$\underline{\boldsymbol{p}} * \underline{\boldsymbol{q}} = (\underline{\boldsymbol{p_p}} * \underline{\boldsymbol{q_p}}) + \boldsymbol{\epsilon} * (\underline{\boldsymbol{p_p}} * \underline{\boldsymbol{q_d}} + \underline{\boldsymbol{p_d}} * \underline{\boldsymbol{q_p}})$$

$$\underline{\boldsymbol{p}} = (p_{p0} + p_{pi}\,\boldsymbol{i} + p_{pj}\,\boldsymbol{j} + p_{pk}\,\boldsymbol{k}) + \boldsymbol{\epsilon} * (p_{d0} + p_{di}\,\boldsymbol{i} + p_{dj}\,\boldsymbol{j} + p_{dk}\,\boldsymbol{k})$$
$$\underline{\boldsymbol{q}} = (q_{p0} + q_{pi}\,\boldsymbol{i} + q_{pj}\,\boldsymbol{j} + q_{pk}\,\boldsymbol{k}) + \boldsymbol{\epsilon} * (q_{d0} + q_{di}\,\boldsymbol{i} + q_{dj}\,\boldsymbol{j} + q_{dk}\,\boldsymbol{k})$$

$$\underline{\boldsymbol{p}} * \underline{\boldsymbol{q}} = (\underline{\boldsymbol{p_p}} * \underline{\boldsymbol{q_p}}) + \boldsymbol{\epsilon} * (\underline{\boldsymbol{p_p}} * \underline{\boldsymbol{q_D}} + \underline{\boldsymbol{p_D}} * \underline{\boldsymbol{q_p}})$$
$$\underline{\boldsymbol{p}} * \underline{\boldsymbol{q}} = (\underline{\mathbf{A}}) + \boldsymbol{\epsilon} * (\underline{\mathbf{B}} + \underline{\mathbf{C}})$$

$$\boldsymbol{A} = \begin{bmatrix} \left[(p_{p_0} * q_{p_0}) - (p_{p_i} * q_{p_i}) - \left(p_{p_j} * q_{p_j}\right) - (p_{p_k} * q_{p_k})\right] \; + \\ \left[(p_{p_0} * q_{p_i}) + (p_{p_i} * q_{p_0}) + \left(p_{p_j} * q_{p_k}\right) - (p_{p_k} * q_{p_j})\right] \boldsymbol{i} + \\ \left[\left(p_{p_0} * q_{p_j}\right) - (p_{p_i} * q_{p_k}) + \left(p_{p_j} * q_{p_0}\right) + (p_{p_k} * q_{p_i})\right] \boldsymbol{j} \; + \\ \left[(p_{p_0} * q_{p_k}) + \left(p_{p_i} * q_{p_j}\right) - \left(p_{p_j} * q_{p_i}\right) + (p_{p_k} * q_{p_0})\right] \; \boldsymbol{k} \end{bmatrix}$$

$$\boldsymbol{B} = \begin{bmatrix} \left[(p_{p_0} * q_{D_0}) - (p_{p_i} * q_{D_i}) - \left(p_{p_j} * q_{D_j}\right) - (p_{p_k} * q_{D_k})\right] \; + \\ \left[(p_{p_0} * q_{D_i}) + (p_{p_i} * q_{D_0}) + \left(p_{p_j} * q_{D_k}\right) - (p_{p_k} * q_{D_j})\right] \boldsymbol{i} + \\ \left[\left(p_{p_0} * q_{D_j}\right) - (p_{p_i} * q_{D_k}) + \left(p_{p_j} * q_{D_0}\right) + (p_{p_k} * q_{D_i})\right] \boldsymbol{j} \; + \\ \left[(p_{p_0} * q_{D_k}) + \left(p_{p_i} * q_{D_j}\right) - \left(p_{p_j} * q_{D_i}\right) + (p_{p_k} * q_{D_0})\right] \; \boldsymbol{k} \end{bmatrix}$$

$$\boldsymbol{C} = \begin{bmatrix} \left[(p_{D_0} * q_{p_0}) - (p_{D_i} * q_{p_i}) - \left(p_{D_j} * q_{p_j}\right) - (p_{D_k} * q_{p_k})\right] \; + \\ \left[(p_{D_0} * q_{p_i}) + (p_{D_i} * q_{p_0}) + \left(p_{D_j} * q_{p_k}\right) - (p_{D_k} * q_{p_j})\right] \boldsymbol{i} + \\ \left[\left(p_{D_0} * q_{p_j}\right) - (p_{D_i} * q_{p_k}) + \left(p_{D_j} * q_{p_0}\right) + (p_{D_k} * q_{p_i})\right] \boldsymbol{j} \; + \\ \left[(p_{D_0} * q_{p_k}) + \left(p_{D_i} * q_{p_j}\right) - \left(p_{D_j} * q_{p_i}\right) + (p_{D_k} * q_{p_0})\right] \; \boldsymbol{k} \end{bmatrix}$$

Uma outra notação para a multiplicação de dois quatérnios duais p e q, poderia ser dada por:

$$\begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \\ p_7 \\ p_8 \end{bmatrix} \blacksquare \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \\ q_7 \\ q_8 \end{bmatrix} = \begin{bmatrix} p_1q_1 - p_2q_2 - p_3q_3 - p_4q_4 \\ p_1q_2 + p_2q_1 + p_3q_4 - p_4q_3 \\ p_1q_3 - p_2q_4 + p_3q_1 + p_4q_2 \\ p_1q_4 + p_2q_3 - p_3q_2 + p_4q_1 \\ p_1q_5 - p_2q_6 - p_3q_7 - p_4q_8 + p_5q_1 - p_6q_2 - p_7q_3 - p_8q_4 \\ p_1q_6 + p_2q_5 + p_3q_8 - p_4q_7 + p_5q_2 + p_6q_1 + p_7q_4 - p_8q_3 \\ p_1q_7 - p_2q_8 + p_3q_5 + p_4q_6 + p_5q_3 - p_6q_4 + p_7q_1 + p_8q_2 \\ p_1q_8 + p_2q_7 - p_3q_6 + p_4q_5 + p_5q_4 + p_6q_3 - p_7q_2 + p_8q_1 \end{bmatrix}$$

Exemplo5:

$\underline{\boldsymbol{q_1}} = (1 + 1\,\boldsymbol{i} + 1\,\boldsymbol{j} + 1\,\boldsymbol{k}) + \boldsymbol{\epsilon} * (2 + 2\,\boldsymbol{i} + 2\,\boldsymbol{j} + 2\,\boldsymbol{k})$

$\underline{\boldsymbol{q_2}} = (3 + 3\,\boldsymbol{i} + 3\,\boldsymbol{j} + 3\,\boldsymbol{k}) + \boldsymbol{\epsilon} * (5 + 5\,\boldsymbol{i} + 5\,\boldsymbol{j} + 5\,\boldsymbol{k})$

$\underline{\boldsymbol{q_1}} * \underline{\boldsymbol{q_2}} = (-6 + 6\,\boldsymbol{i} + 6\,\boldsymbol{j} + 6\,\boldsymbol{k}) + \boldsymbol{\epsilon} * (-22 + 22\,\boldsymbol{i} + 22\,\boldsymbol{j} + 22\,\boldsymbol{k})$

A6. Norma ou Magnitude de um Quatérnio Dual

$$\| \underline{\boldsymbol{q}} \|^2 = \underline{\boldsymbol{q}} * \underline{\boldsymbol{q}}^*$$

Exemplo6:

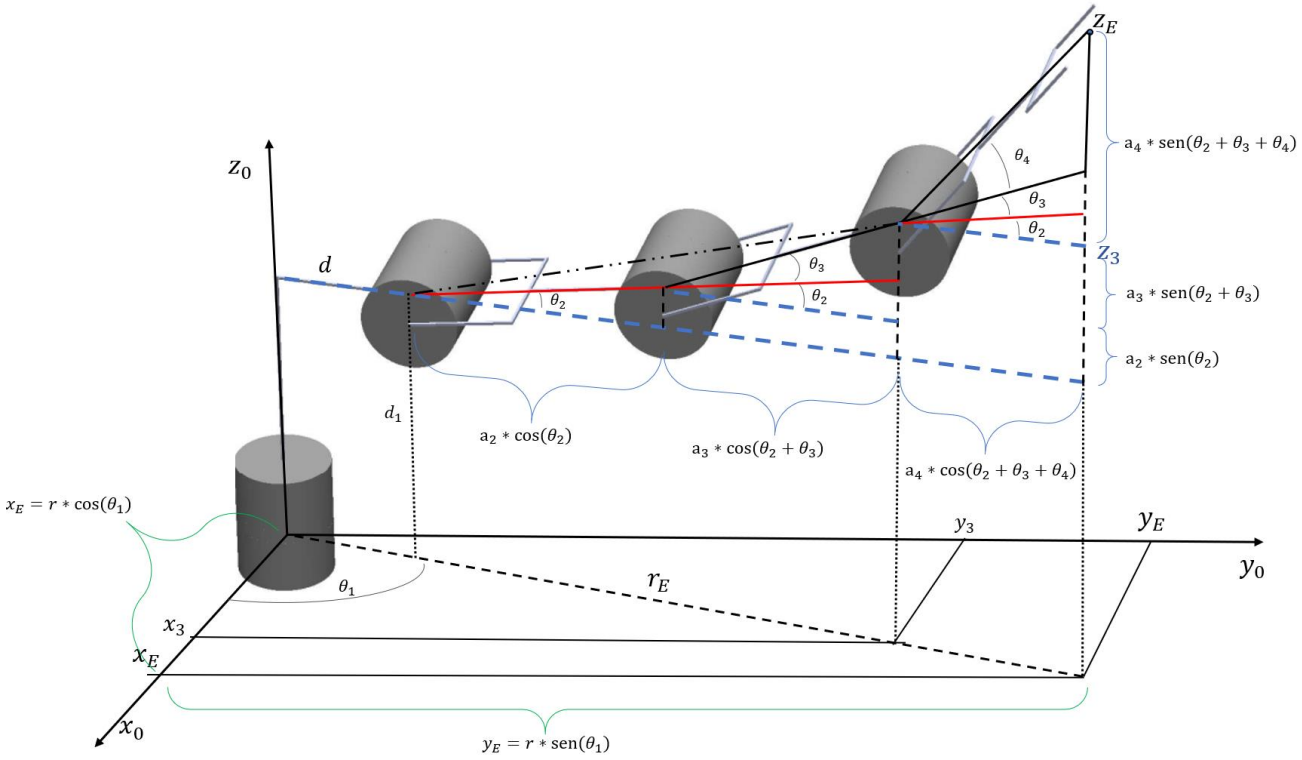
$\underline{\boldsymbol{q}} = (+3 + 2\,\boldsymbol{i} + 2\,\boldsymbol{j} + 2\,\boldsymbol{k}) + \boldsymbol{\epsilon} * (+5 + 7\,\boldsymbol{i} + 7\,\boldsymbol{j} + 7\,\boldsymbol{k})$

$\underline{\boldsymbol{q}}^* = (+3 - 2\,\boldsymbol{i} - 2\,\boldsymbol{j} - 2\,\boldsymbol{k}) + \boldsymbol{\epsilon} * (+5 - 7\,\boldsymbol{i} - 7\,\boldsymbol{j} - 7\,\boldsymbol{k})$

$$\underline{\boldsymbol{q}} * \underline{\boldsymbol{q}^*} = (\sqrt{21} + 0\,\boldsymbol{i} + 0\,\boldsymbol{j} + 0\,\boldsymbol{k}) + \boldsymbol{\epsilon} * (\sqrt{114} + 0\,\boldsymbol{i} + 0\,\boldsymbol{j} + 0\,\boldsymbol{k})$$

Apêndices B – Cálculo da cinemática direta usando o Maple

B1. Cálculo Cinemática Direta geometricamente



$$r_E = d + a_2\cos(\theta_2) + a_3\cos(\theta_2 + \theta_3) + a_4\cos(\theta_2 + \theta_3 + \theta_4)$$
$$x_E = r_E * \cos(\theta_1)$$
$$y_E = r_E * \text{sen}(\theta_1)$$
$$z_E = d_1 + a_2\,\text{sen}(\theta_2) + a_3\,\text{sen}(\theta_2 + \theta_3) + a_4\,\text{sen}(\theta_2 + \theta_3 + \theta_4)$$

$$\begin{cases} x_E = [d + a_2\cos(\theta_2) + a_3\cos(\theta_2 + \theta_3) + a_4\cos(\theta_2 + \theta_3 + \theta_4)] * \cos(\theta_1) \\ y_E = [d + a_2\cos(\theta_2) + a_3\cos(\theta_2 + \theta_3) + a_4\cos(\theta_2 + \theta_3 + \theta_4)] * \text{sen}(\theta_1) \\ z_E = d_1 + a_2\,\text{sen}(\theta_2) + a_3\,\text{sen}(\theta_2 + \theta_3) + a_4\,\text{sen}(\theta_2 + \theta_3 + \theta_4) \end{cases}$$

COMANDOS NO MAPLE:

$x_E := (d + a2\cdot\cos(\theta 2) + a3\cdot\cos(\theta 2 + \theta 3) + a4\cdot\cos(\theta 2 + \theta 3 + \theta 4))\cdot\cos(\theta 1);$

$x_E := -4.732537915$

$y_E := (d + a2\cdot\cos(\theta 2) + a3\cdot\cos(\theta 2 + \theta 3) + a4\cdot\cos(\theta 2 + \theta 3 + \theta 4))\cdot\sin(\theta 1);$

$y_E := 26.83955623$

$z_E := d1 + a2\cdot\sin(\theta 2) + a3\cdot\sin(\theta 2 + \theta 3) + a4\cdot\sin(\theta 2 + \theta 3 + \theta 4);$

$z_E := 17.25541671$

B2. Cálculo Cinemática Direta com Matriz de Transformação Homogênea

COMANDOS NO MAPLE:

$$S := \sin(\theta) :$$
$$C := \cos(\theta) :$$
$$Sa := \sin(alpha) :$$
$$Ca := \cos(alpha) :$$

$$H := \begin{bmatrix} C & -(S\cdot Ca) & (S\cdot Sa) & a\cdot C \\ S & (C\cdot Ca) & -(C\cdot Sa) & a\cdot S \\ 0 & Sa & Ca & \mathrm{d} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H := \begin{bmatrix} \cos(\theta) & -\sin(\theta)\cos(\alpha) & \sin(\theta)\sin(\alpha) & a\cos(\theta) \\ \sin(\theta) & \cos(\theta)\cos(\alpha) & -\cos(\theta)\sin(\alpha) & a\sin(\theta) \\ 0 & \sin(\alpha) & \cos(\alpha) & d \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Tabela Denavit Hartenberg:

Theta	Alpha(∝)	a	d
$\theta 1 := convert(100.0\ degrees, radians)$ $\theta 1 := 1.745329252$ **(1)**	$alpha1 := convert(90.0\ degrees, radians) :$	$d := 1.13 :$ $a1 := d :$	$d1 := 8.40 :$
$\theta 2 := convert(20.0\ degrees, radians)$ $\theta 2 := 0.3490658504$ **(2)**	$alpha2 := convert(0.0\ degrees, radians) :$	$a2 := 9.81 :$	$d2 := 0.0 :$
$\theta 3 := convert(30.0\ degrees, radians)$ $\theta 3 := 0.5235987757$ **(3)**	$alpha3 := convert(0.0\ degrees, radians) :$	$a3 := 7.18 :$	$d3 := 0.0 :$
$\theta 4 := convert(-50.0\ degrees, radians)$ $\theta 4 := -0.8726646261$ **(4)**	$alpha4 := convert(0.0\ degrees, radians) :$	$a4 := 12.29 :$	$d4 := 0.0 :$
$\theta 5 := convert(0.0\ degrees, radians)$ $\theta 5 := 0$ **(5)**	$alpha5 := convert(0.0\ degrees, radians) :$	$a5 := 0.0 :$	$d5 := 0.0 :$

$$H1 := eval(H, \{\theta = \theta 1,\ alph = alpha1, a = a1, d = d1\});$$

$$H1 := \begin{bmatrix} -0.1736481777 & 2.019873996\ 10^{-10} & 0.9848077530 & -0.1962224408 \\ 0.9848077530 & 3.561582832\ 10^{-11} & 0.1736481777 & 1.112832761 \\ 0 & 1. & -2.051033808\ 10^{-10} & 8.40 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H2 := eval(H, \{\theta = \theta 2,\ alph = alpha2,\ a = a2,\ d = d2\});$$

$$H2 := \begin{bmatrix} 0.9396926208 & -0.3420201433 & 0. & 9.218384610 \\ 0.3420201433 & 0.9396926208 & -0. & 3.355217606 \\ 0 & 0 & 1 & 0. \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H3 := eval(H, \{\theta = \theta 3,\ alph = alpha3,\ a = a3,\ d = d3\});$$

$$H3 := \begin{bmatrix} 0.8660254037 & -0.5000000001 & 0. & 6.218062399 \\ 0.5000000001 & 0.8660254037 & -0. & 3.590000001 \\ 0 & 0 & 1 & 0. \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H4 := eval(H, \{\theta = \theta 4,\ alph = alpha4,\ a = a4,\ d = d4\});$$

$$H4 := \begin{bmatrix} 0.6427876096 & 0.7660444432 & -0. & 7.899859722 \\ -0.7660444432 & 0.6427876096 & -0. & -9.414686207 \\ 0 & 0 & 1 & 0. \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H5 := eval(H, \{\theta = \theta 5,\ alph = alpha5,\ a = a5,\ d = d5\});$$

$$H5 := \begin{bmatrix} 1 & 0 & 0 & 0. \\ 0 & 1 & 0 & 0. \\ 0 & 0 & 1 & 0. \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

COMANDOS NO MAPLE:

```
H1H2 := evalm(H1&*H2);
```

$$H1H2 := \begin{bmatrix} -0.1631759111 & 0.05939117481 & 0.9848077530 & -1.796978129 \\ 0.9254165784 & -0.3368240888 & 0.1736481777 & 10.19116940 \\ 0.3420201433 & 0.9396926208 & -2.051033808\ 10^{-10} & 11.75521761 \\ 0. & 0. & 0. & 1. \end{bmatrix}$$

```
H1H2H3 := evalm(H1H2&*H3);
```

$$H1H2H3 := \begin{bmatrix} -0.1116188969 & 0.1330222217 & 0.9848077530 & -2.598401808 \\ 0.6330222215 & -0.7544065068 & 0.1736481777 & 14.73626895 \\ 0.7660444432 & 0.6427876096 & -2.051033808\ 10^{-10} & 17.25541671 \\ 0. & 0. & 0. & 1. \end{bmatrix}$$

```
H1H2H3H4 := evalm(H1H2H3&*H4);
```

$$H1H2H3H4 := \begin{bmatrix} -0.1736481777 & 1.8\ 10^{-10} & 0.9848077530 & -4.732537912 \\ 0.9848077530 & 0. & 0.1736481777 & 26.83955623 \\ 0. & 1.000000000 & -2.051033808\ 10^{-10} & 17.25541671 \\ 0. & 0. & 0. & 1. \end{bmatrix}$$

```
H1H2H3H4H5 := evalm(H1H2H3H4&*H5);
```

$$H1H2H3H4H5 := \begin{bmatrix} -0.1736481777 & 1.8\ 10^{-10} & 0.9848077530 & -4.732537912 \\ 0.9848077530 & 0. & 0.1736481777 & 26.83955623 \\ 0. & 1.000000000 & -2.051033808\ 10^{-10} & 17.25541671 \\ 0. & 0. & 0. & 1. \end{bmatrix}$$

```
with(linalg) :
POSICAO := submatrix(H1H2H3H4H5, 1 ..3, 4 ..4)
```

$$POSICAO := \begin{bmatrix} -4.732537912 \\ 26.83955623 \\ 17.25541671 \end{bmatrix}$$

```
X := POSICAO[1, 1];
```

$$X := -4.732537912$$

```
Y := POSICAO[2, 1];
```

$$Y := 26.83955623$$

```
Z := POSICAO[3, 1];
```

$$Z := 17.25541671$$

B3. Cálculo Cinemática Direta com Quatérnios Duais

COMANDOS NO MAPLE:

$$S := \sin\left(\frac{\theta}{2}\right):$$

$$C := \cos\left(\frac{\theta}{2}\right):$$

$$Sa := \sin\left(\frac{alph}{2}\right):$$

$$Ca := \cos\left(\frac{alph}{2}\right):$$

$$Q := \left[\, Ca\cdot C \;\; (Sa\cdot C) \;\; Sa\cdot S \;\; Ca\cdot S \;\; \left(-\frac{a}{2}\cdot Sa\cdot C - \frac{d}{2}\cdot Ca\cdot S\right) \;\; \left(+\frac{a}{2}\cdot Ca\cdot C - \frac{d}{2}\cdot Sa\cdot S\right) \;\; +\frac{a}{2}\cdot Ca\cdot S + \frac{d}{2}\cdot Sa\cdot C \;\; \left(-\frac{a}{2}\cdot Sa\cdot S + \frac{d}{2}\cdot Ca\cdot C\right) \right]:$$

Tabela Denavit Hartenberg:

Theta	Alpha(∝)	a	d
$\theta 1 := convert(100.0\ degrees, radians)$ $\theta 1 := 1.745329252$ (1)	$alpha1 := convert(90.0\ degrees, radians):$	$d := 1.13:$ $a1 := d:$	$d1 := 8.40:$
$\theta 2 := convert(20.0\ degrees, radians)$ $\theta 2 := 0.3490658504$ (2)	$alpha2 := convert(0.0\ degrees, radians):$	$a2 := 9.81:$	$d2 := 0.0:$
$\theta 3 := convert(30.0\ degrees, radians)$ $\theta 3 := 0.5235987757$ (3)	$alpha3 := convert(0.0\ degrees, radians):$	$a3 := 7.18:$	$d3 := 0.0:$
$\theta 4 := convert(-50.0\ degrees, radians)$ $\theta 4 := -0.8726646261$ (4)	$alpha4 := convert(0.0\ degrees, radians):$	$a4 := 12.29:$	$d4 := 0.0:$
$\theta 5 := convert(0.0\ degrees, radians)$ $\theta 5 := 0$ (5)	$alpha5 := convert(0.0\ degrees, radians):$	$a5 := 0.0:$	$d5 := 0.0:$

$with(LinearAlgebra):$

$Q1 := Transpose(Resultado);$

$$Q1 := \begin{bmatrix} 0.7071067811\cos\left(\frac{\theta 1}{2}\right) \\ 0.7071067813\cos\left(\frac{\theta 1}{2}\right) \\ 0.7071067813\sin\left(\frac{\theta 1}{2}\right) \\ 0.7071067811\sin\left(\frac{\theta 1}{2}\right) \\ -0.399515331400000\cos\left(\frac{\theta 1}{2}\right) - 2.969848481\sin\left(\frac{\theta 1}{2}\right) \\ 0.399515331300000\cos\left(\frac{\theta 1}{2}\right) - 2.969848481\sin\left(\frac{\theta 1}{2}\right) \\ 0.399515331300000\sin\left(\frac{\theta 1}{2}\right) + 2.969848481\cos\left(\frac{\theta 1}{2}\right) \\ -0.399515331400000\sin\left(\frac{\theta 1}{2}\right) + 2.969848481\cos\left(\frac{\theta 1}{2}\right) \end{bmatrix} := \begin{bmatrix} 0.4545194776 \\ 0.4545194778 \\ 0.5416752205 \\ 0.5416752203 \\ -2.53183943032290 \\ -2.01823242121243 \\ 2.21502830622832 \\ 1.60293530666053 \end{bmatrix}$$

```
with(LinearAlgebra) :
Q2 := Transpose(Resultado);
```

$$Q2 := \begin{bmatrix} 0.9848077530 \\ 0 \\ 0 \\ 0.1736481777 \\ 0. \\ 4.83048202846500 \\ 0.851744311618500 \\ 0. \end{bmatrix}$$

```
with(LinearAlgebra) :
Q3 := Transpose(Resultado);
```

$$Q3 := \begin{bmatrix} 0.9659258263 \\ 0 \\ 0 \\ 0.2588190451 \\ 0. \\ 3.46767371641700 \\ 0.929160371909000 \\ 0. \end{bmatrix}$$

```
with(LinearAlgebra) :
Q4 := Transpose(Resultado);
```

$$Q4 := \begin{bmatrix} 0.9063077870 \\ 0 \\ 0 \\ -0.4226182617 \\ 0. \\ 5.56926135111500 \\ -2.59698921814650 \\ 0. \end{bmatrix}$$

```
Resultado := eval(Multip_Quater)
```

$$Resultado := [0., 0., 0., 0., -3.12386028156197\ 10^{-9}, -4.73253791154906, 26.8395562329541, 17.2554167076794]$$

```
Rotação := eval(Rotacao);
```

$$Rotação := [\ 0.\ 0.\ 0.\ 0.\ ]$$

```
Translação := eval(Translacao);
```

$$Translação := [-3.12386028156197\ 10^{-9}, -4.73253791154906, 26.8395562329541, 17.2554167076794]$$

COORDENADAS DO PONTO CINEMATICA DIRETA (Dual Quatérnio):

```
x := Translação[2]
```

$$x := -4.73253791154906$$

```
y := Translação[3]
```

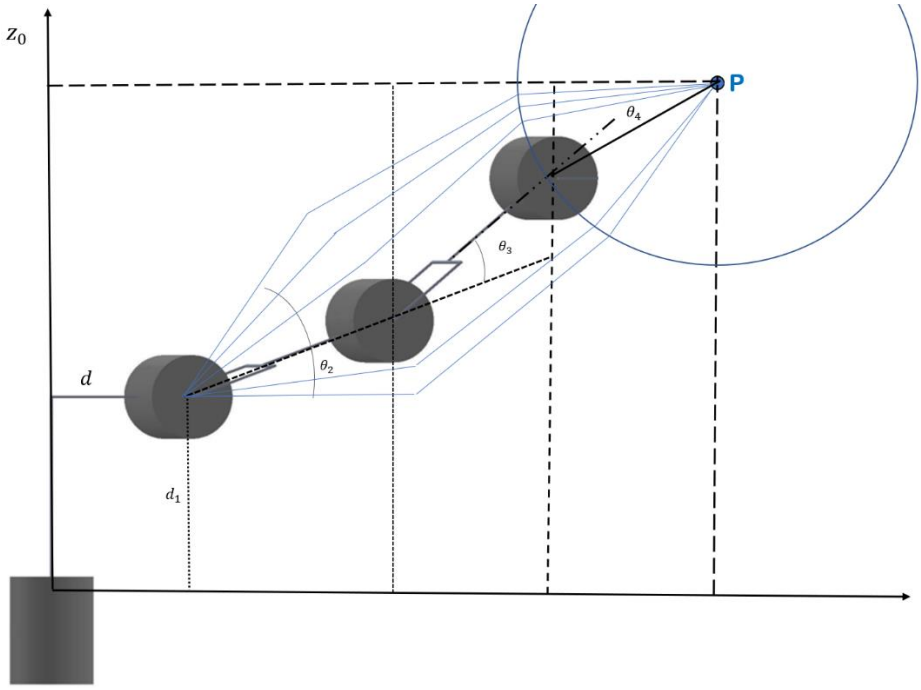
$$y := 26.8395562329541$$

```
z := Translação[4]
```

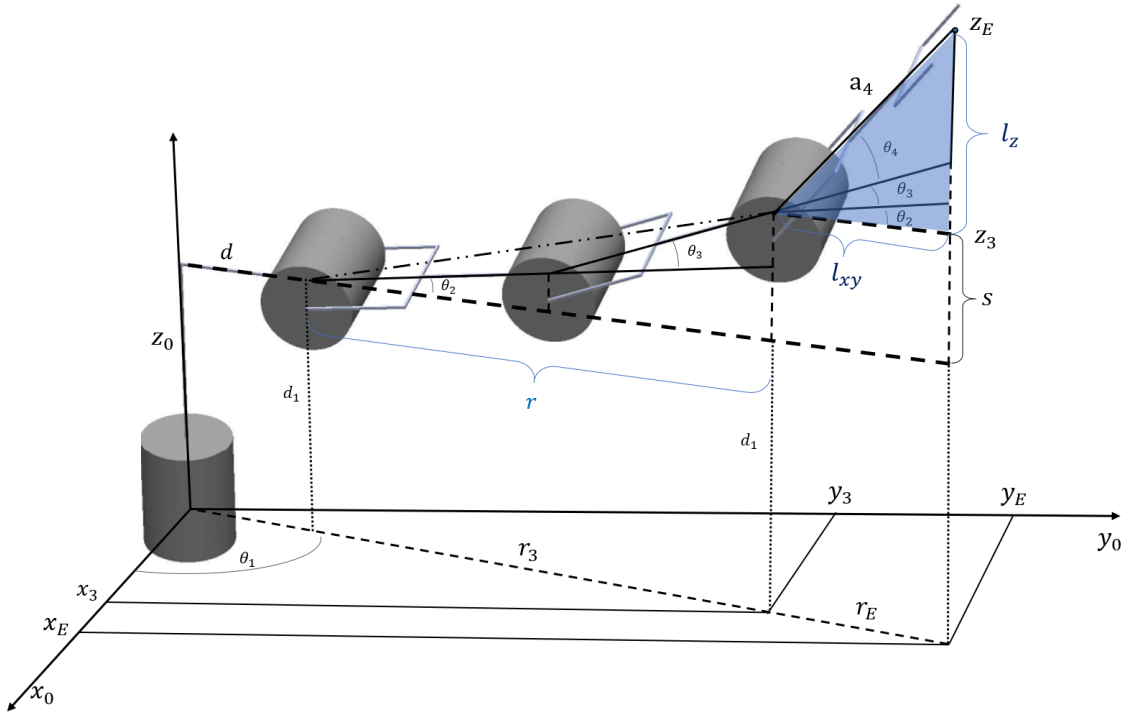
$$z := 17.2554167076794$$

Apêndices C – Cálculo da cinemática inversa geometricamente

Nota-se que para o caso da cinemática inversa não pode ser resolvida, pois há apenas três equações com quatro (4) incógnitas (os quatro ângulos das juntas). De fato, pode-se mostrar facilmente que existem infinitas soluções de ângulos que satisfazem a condição do órgão terminal atingir um dado ponto P no plano (Figura).



É necessário assumir uma condição a mais para que a cinemática inversa tenha uma única solução, a forma a ser estabelecida é fixar a orientação da quarta junta com o ângulo ∅ (com relação à horizontal). Isto irá significar que nem todas as soluções satisfazem as equações, mas somente aquela (ou aquelas) nas quais o ângulo do elo a_4 com relação à horizontal for igual a ∅ (fornecido). Indicando claramente que a posição da junta 4 pode ser determinada (Figura) .

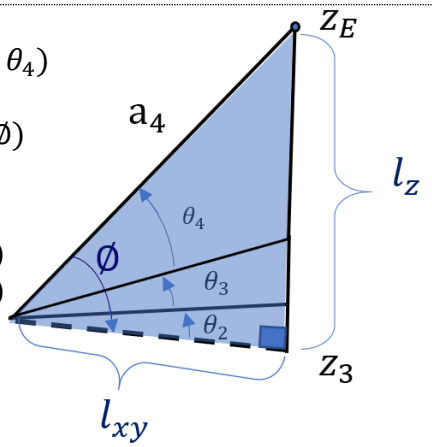


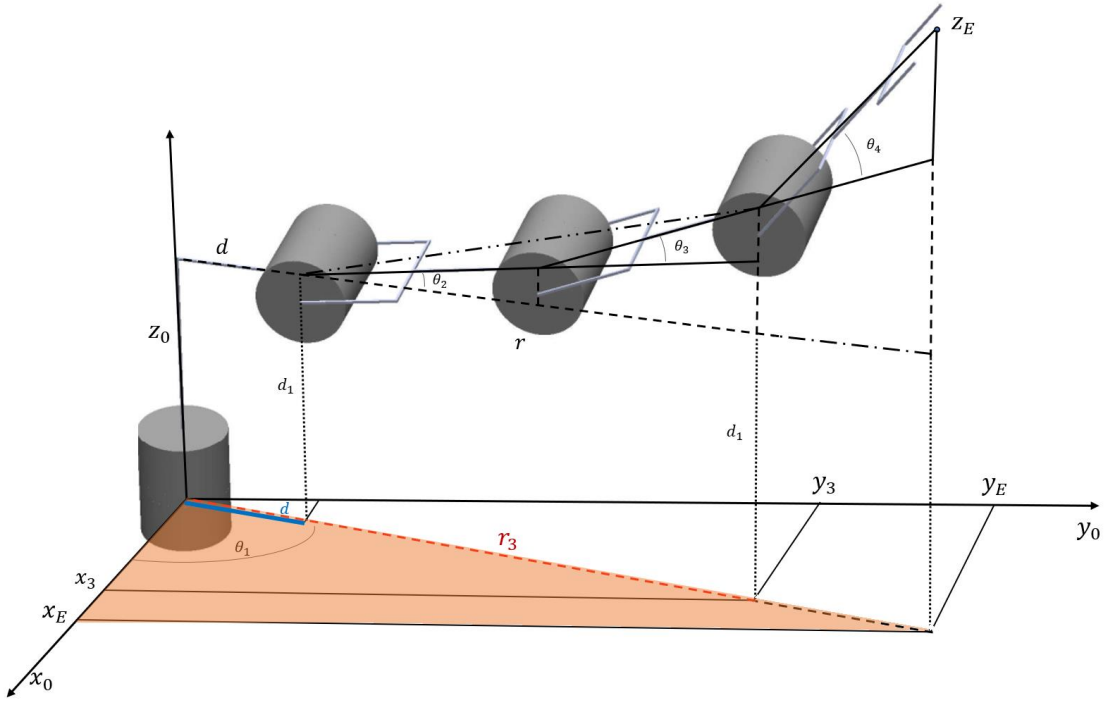
$$s = z_E - d_1 - l_z$$
$$r = \sqrt{{x_E}^2 + {y_E}^2} - l_{xy} - d$$

$$\emptyset = -(\theta_2 + \theta_3 + \theta_4)$$

$$l_{xy} = a_4 * \cos(-\emptyset)$$
$$l_{xy} = a_4 * \cos(\emptyset)$$

$$l_z = a_4 * \sin(-\emptyset)$$
$$l_z = -a_4 * \sin(\emptyset)$$





$$\theta_1 = \arctan2\left(\frac{y_E}{x_E}\right)$$

COMANDOS NO MAPLE:

$x_E := -4.732537915$:

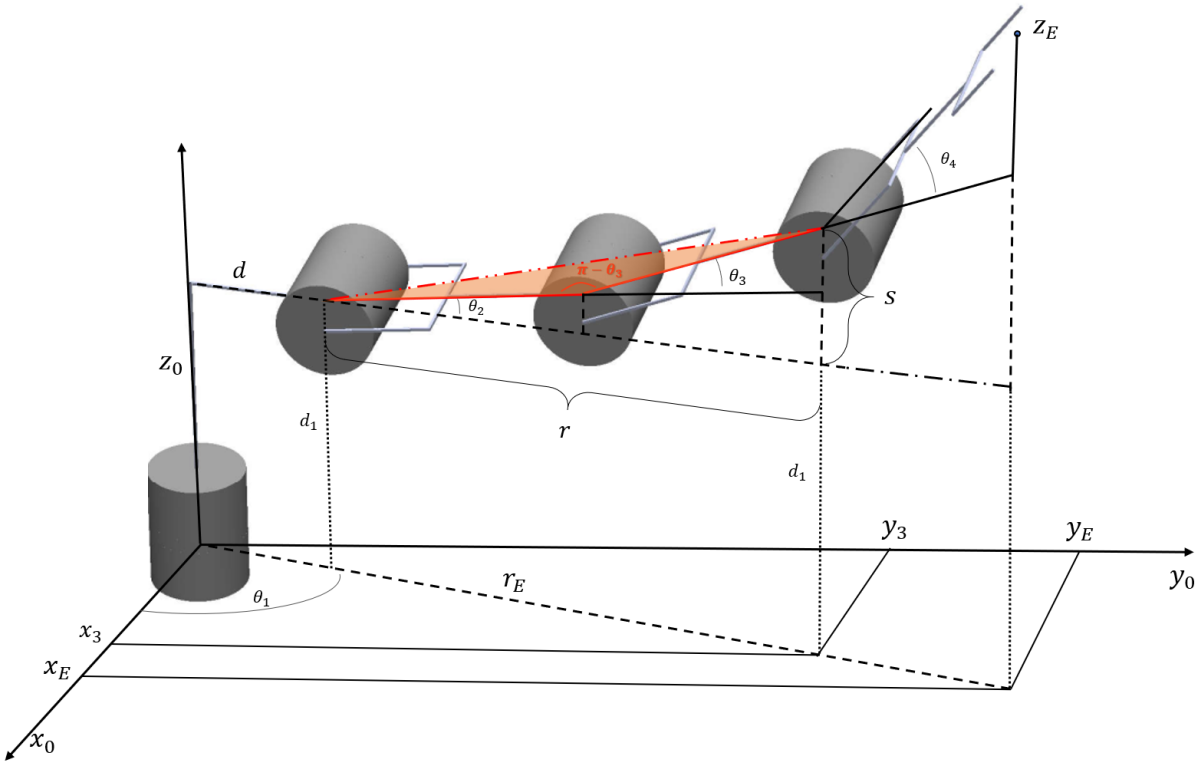
$y_E := 26.83955623$:

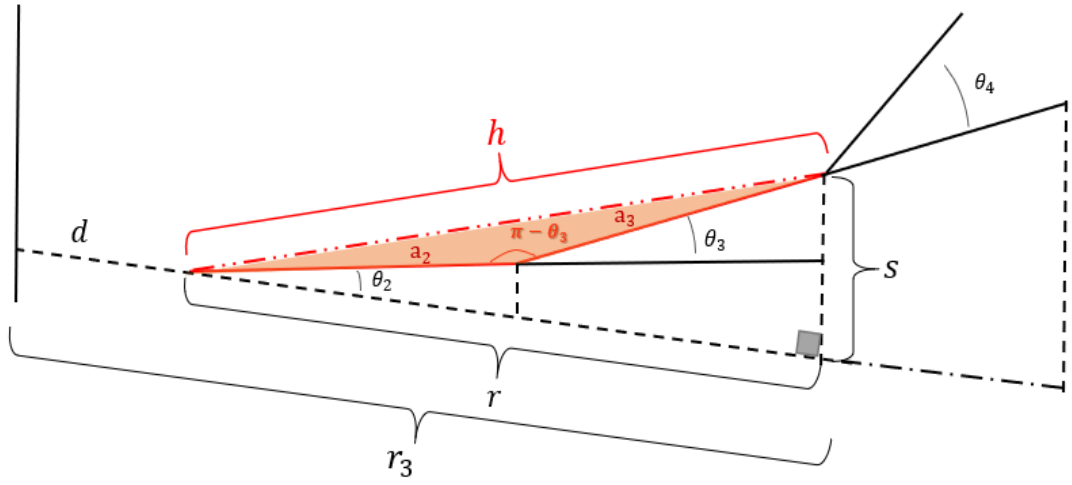
$z_E := 17.25541671$:

$th1\_rad := \arctan(y_E, x_E)$:

$th1 := convert(th1\_rad, degrees)$;

$th1 := 99.99999999\ degrees$





$$\begin{cases} \textbf{(I)}\ h^2 = {a_2}^2 + {a_3}^2 - 2 * a_2 * a_3 * \cos(\pi - \theta_3) \rightarrow \textbf{lei dos cossenos} \\ \textbf{(II)}\ \cos(\pi - \theta_3) = -\cos(\theta_3) \end{cases}$$

Substituindo (II) em (I):

$$\textbf{(III)}\ h^2 = {a_2}^2 + {a_3}^2 + 2 * a_2 * a_3 * \cos(\theta_3)$$

$$\begin{cases} \textbf{(III)}\ h^2 = {a_2}^2 + {a_3}^2 + 2 * a_2 * a_3 * \cos(\theta_3) \\ \textbf{(IV)}\ h^2 = s^2 + r^2 \rightarrow \textbf{pitagoras} \end{cases}$$

Substituindo (IV) em (III):

$$\textbf{(V)}\ s^2 + r^2 - {a_2}^2 - {a_3}^2 = +2 * a_2 * a_3 * \cos(\theta_3)$$

$$\cos(\theta_3) = \frac{s^2 + r^2 - {a_2}^2 - {a_3}^2}{2 * a_2 * a_3}$$

$$\cos(\theta_3) = \frac{(z_E - d_1 - \ l_z)^2 \ + \left(\sqrt{{x_E}^2 + {y_E}^2} - \ l_{xy} - d\right)^2 \ - {a_2}^2 - {a_3}^2}{2 * a_2 * a_3}$$

$$\cos(\theta_3) = \frac{(z_E - d_1 + \mathrm{a}_4 \sin(\emptyset))^2 \ + \left(\sqrt{{x_E}^2 + {y_E}^2} - \ \mathrm{a}_4 \cos(\emptyset) \ - d\right)^2 - {a_2}^2 - {a_3}^2}{2 * a_2 * a_3} = \boldsymbol{\beta}$$

$$\theta_3 = \arctan2\left(\frac{\pm\sqrt{1-\beta^2}}{\beta}\right)$$

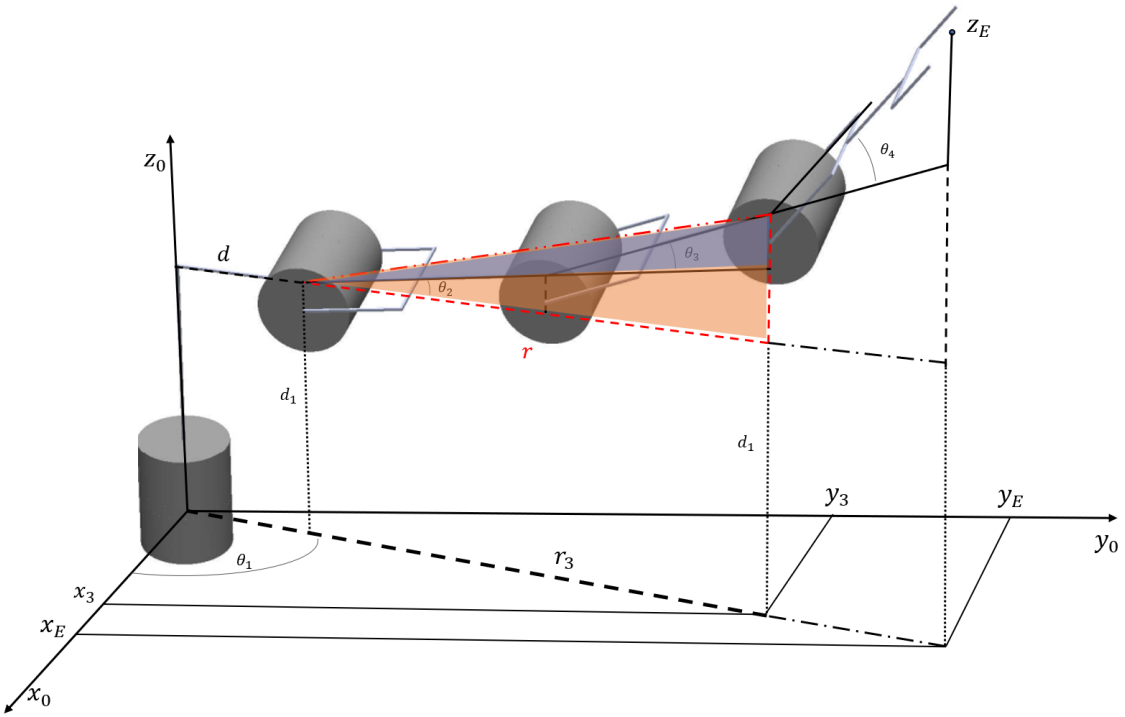
COMANDOS NO MAPLE:

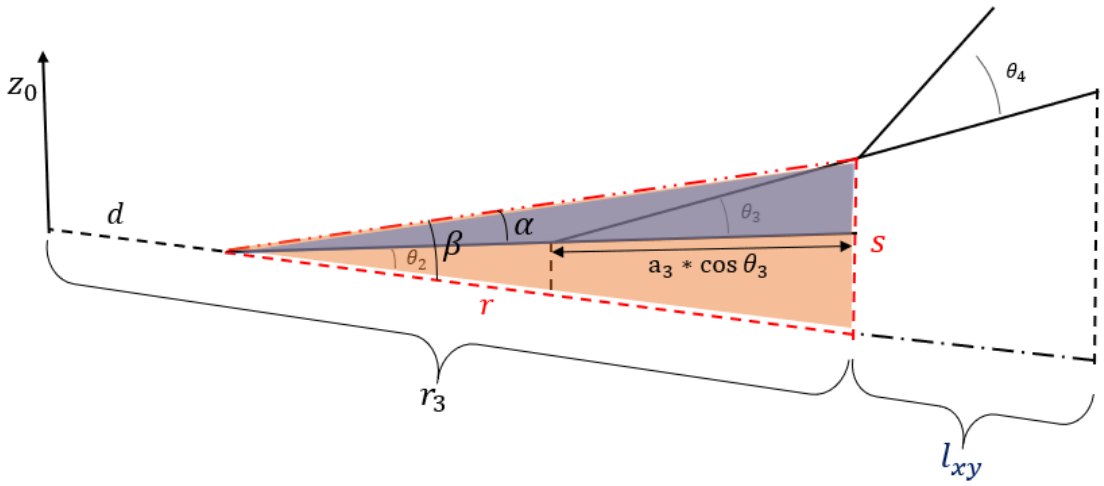
$$\beta := \frac{\left(z_E - d1 + a4 \cdot \sin(\Phi)\right)^2 + \left(\sqrt{(x_E)^2 + (y_E)^2} - a4 \cdot \cos(\Phi) - d\right)^2 - a2^2 - a3^2}{2 \cdot a2 \cdot a3} :$$

$$th3\_rad := \arctan\left(\sqrt{1 - (\beta)^2}, \beta\right) :$$

$$th3 := convert(th3\_rad, degrees);$$

$$th3 := 29.99999990\ degrees$$





$$\begin{cases} \beta = \text{atan2}\left(\frac{s}{r}\right) \\ \alpha = \text{atan2}\left(\frac{a_3 * \sin\theta_3}{a_2 + a_3 * \cos\theta_3}\right) \\ \theta_2 = \beta - \alpha \end{cases}$$

$$\theta_2 = \text{atan2}\left(\frac{s}{r}\right) - \text{atan2}\left(\frac{a_3 * \sin\theta_3}{a_2 + a_3 * \cos\theta_3}\right)$$

$$\theta_2 = \text{atan2}\left(\frac{z_E - d_1 - l_z}{\sqrt{{x_E}^2 + {y_E}^2} - l_{xy} - d}\right) - \text{atan2}\left(\frac{a_3 * \sin\theta_3}{a_2 + a_3 * \cos\theta_3}\right)$$

$$\theta_2 = \text{atan2}\left(\frac{z_E - d_1 + a_4 * \sin(\emptyset)}{\sqrt{{x_E}^2 + {y_E}^2} - a_4 * \cos(\emptyset) - d}\right) - \text{atan2}\left(\frac{a_3 * \sin\theta_3}{a_2 + a_3 * \cos\theta_3}\right)$$

COMANDOS NO MAPLE:

```
th2_rad1 := arctan( (z_E - d1 + a4·sin(Φ)) , (sqrt((x_E)^2 + (y_E)^2) - a4·cos(Φ) - d) )
            -arctan( (a3·sin(th3_rad)) , (a2 + a3·cos(th3_rad)) ) :

th2_1 := convert(th2_rad1 , degrees);

                    th2_1 := 20.00000005 degrees
```

$$\emptyset = -(\theta_2 + \theta_3 + \theta_4)$$

$$\theta_4 = -\emptyset - \theta_2 - \theta_3$$

COMANDOS NO MAPLE:

```
th4_rad := -Φ - th2_rad1 -th3_rad :
th4 := convert(th4_rad , degrees);

                    th4 := -49.99999993 degrees
```

Apêndices D – Exemplos de análise eficiência computacional – Modelo RAM

Exemplo 1:

Algoritmo para Soma de Matrizes	
Algoritmo	nº de passos
for$(i = 0;\ i < n;\ i++)$	$\boldsymbol{n} *$ (valor numérico da variável n= executa o loop n vezes, passando pelo comando 'for')
for$(j = 0;\ j < n; j++)$	$\boldsymbol{n} *$ (valor numérico da variável n= executa o loop n vezes, passando pelo comando 'for')
c[i][j] = a[i][j] + b[i][j];	**2** (1 soma, 1 atribuição)
Número de operações F(n)=	$\boldsymbol{2 * n^2}$

Exemplo 2:

Algoritmo para Multiplicação de Matrizes	
Algoritmo	nº de passos
for$(i = 0;\ i < n;\ i++)$	$\boldsymbol{n} *$ (valor numérico da variável n= executa o loop n vezes, passando pelo comando 'for')
for$(j = 0;\ j < n; j++)$	$\boldsymbol{n} *$ (valor numérico da variável n= executa o loop n vezes, passando pelo comando 'for')
{ c[i][j] = 0;	**1+** (1 atribuição de valor à variável)
for$(k = 0;\ k < n; k++)$	$\boldsymbol{n} *$ (valor numérico da variável n= executa o loop n vezes, passando pelo comando 'for')
c[i][j] = c[i][j] + a[i][k] * b[k][j]; **}**	**3** (1 soma, 1 multiplicação, 1 atribuição)
Número de operações F(n)=	$\boldsymbol{n * n * (1 + 3n)}$= $\boldsymbol{3 * n^3 + n^2}$

Exemplo 3:

Algoritmo para calcular a soma parcial de um número	
Algoritmo	nº de passos
int $soma_{parcial} = 0;$	**2 +** (1 declarar a variável, 1 atribuição de valor)
for$(i = 0;\ i \leq n;\ i++)$	$(\boldsymbol{n + 1}) *$ (valor numérico da variável n= executa o loop n+1 vezes, passando pelo comando 'for')
{ $soma_{parcial} = soma_{parcial} + i * i * i;$	**4 +** (1 atribuição à variável, 1 soma, 2 multiplicações)
print $(soma_{parcial})$ **}**	**1** (1 escrita na memória)
Número de operações F(n)=	$\boldsymbol{2 + (n + 1) * (4 + 1)}$= $\boldsymbol{5 * n + 7}$

Exemplo 4:

Algoritmo Simples para Contagem de zeros em um array	
Algoritmo	nº de passos
int $cont = 0;$	$\mathbf{2} +$ (1 declarar a variável, 1 atribuição de valor)
int $i;$	$\mathbf{1} +$ (1 declarar a variável)
for$(i = 0;\ i \leq n;\ i++)$	$(\boldsymbol{n} + \mathbf{1}) *$ (valor numérico da variável n= executa o loop n+1 vezes, passando pelo comando 'for')
{ if (v[i] == 0)	$\mathbf{1} +$ (1 comparação por igualdade)
$cont = cont + 1;$	$\mathbf{2}$ (1 atribuição de valor, 1 soma)
}	
Número de operações F(n)=	$\mathbf{2 + 1 + (n + 1) * (1 + 2)=}$ $\mathbf{3 * n + 6}$

Exemplo 5:

Algoritmo Fibonacci	
Algoritmo	nº de passos
if $(n == 0)$	$\mathbf{1} +$ (1 comparação por igualdade)
$\boldsymbol{return}\ 0;$	$\mathbf{1} +$ (1 retornar um valor)
$\boldsymbol{else\ if}(n == 1)$	$\mathbf{1} +$ (1 comparação por igualdade)
$\boldsymbol{return}\ 1;$	$\mathbf{1} +$ (1 retornar um valor)
$\boldsymbol{else}$	$\mathbf{2} +$ (1 atribuição de valor, 1 soma)
{	
$\boldsymbol{p_{prev}} = 0;$	$\mathbf{1}$ (1 atribuição)
$\boldsymbol{prev} = 1;$	$\mathbf{1}$ (1 atribuição)
for$(i = 2;\ i \leq n;\ i++)$	$(\boldsymbol{n} - \mathbf{1}) *$ (valor numérico da variável n= executa o loop n+1 vezes, passando pelo comando 'for')
{	
$\boldsymbol{f = p_{prev} + prev};$	$\mathbf{3} +$ (1 soma, 1 multiplicação e 1 atribuição)
$\boldsymbol{p_{prev} = prev};$	$\mathbf{1+}$ (1 atribuição)
$\boldsymbol{prev = f};$	$\mathbf{1}$ (1 atribuição)
}	
}	
$\boldsymbol{return}\ f;$	$\mathbf{1} +$ (1 retornar um valor)
Número de operações F(n)=	$\mathbf{8 + (n + 1) * (3 + 2) + 1=}$ $\mathbf{5 * n + 14}$

Exemplo 6:

Algoritmo encontrar maior número de um array	
Algoritmo	nº de passos
$\boldsymbol{num_{maior}} = A[0];$	$\mathbf{1} +$ (1 atribuição)
for$(i = 1;\ i \leq n - 1;\ i++)$ **{**	$(\boldsymbol{n} + \mathbf{1}) *$ (valor numérico da variável n= executa o loop de 1 até n-1 vezes, passando pelo comando 'for')
if $(A[i] > \boldsymbol{num_{maior}})$	$\mathbf{1} +$ (1 comparação por maior)
$\{\boldsymbol{num_{maior}} = A[i];\}$	$\mathbf{1} +$ (1 atribuição)
}	
return $num_{maior};$	$\mathbf{1} +$ (1 retornar um valor)
Número de operações F(n)=	$\mathbf{1} + (\boldsymbol{n} + \mathbf{1}) * (\mathbf{1} + \mathbf{1}) + \mathbf{1}=$ $\mathbf{2} * \boldsymbol{n} + \mathbf{4}$