

Universidade de São Paulo
Escola Politécnica da USP

PMR3405 – Mecanismos para Automação

Robô Manipulador

Ana Cláudia Macedo Vianna Miachon (9403902)
Caio Noboru Asai (9833580)
Gustavo Garcia Frozoni (10333550)
Lucas Satoru Aracena Perez (5968301)
Marcelo Monari Baccaro (8989262)
Mirko del Piero Flores Sanchez (11710458)
Rennan Kusaba (11556391)
Victor Ribeiro Miguez (9350369)

São Paulo
Julho 2020

Conteúdo

1	Introdução	4
2	Trajectoria do objeto carregado	5
2.1	Interpolação da trajetória	5
2.2	Simulação da trajetória	6
3	Síntese do modelo preliminar	7
4	Modelo cinemático	8
4.1	Esquema simplificado do mecanismo	8
4.2	Obtenção da posição, velocidade e aceleração da garra do robô em função das coordenadas generalizadas	9
4.3	Análise da cinemática inversa da trajetória do ponto P	10
4.3.1	Obtenção das coordenadas generalizadas na trajetória	10
4.3.2	Obtenção das velocidades generalizadas na trajetória	10
4.3.3	Obtenção das acelerações generalizadas da trajetória	11
4.4	Resultados numéricos da cinemática inversa	12
5	Dinâmica	14
5.1	Seleção de materiais	14
5.2	Modelo em CAD	14
5.2.1	Elo 1	15
5.2.2	Elo 2	17
5.2.3	Elo 3	19
5.2.4	Elo 4	21
5.2.5	Alcance da garra e região de trabalho	22
5.2.6	Resumo de valores relevantes para os modelos dinâmicos e análises	22
5.2.7	Valores assumidos	23
5.3	Método Iterativo de Newton-Euler	23
5.3.1	Modelo teórico de corpos rígidos	23
5.3.2	Equações da dinâmica do robô PRP	24
5.4	Método de Gibbs-Appell	25
5.4.1	Modelo teórico de corpos rígidos	25
5.4.2	Velocidades e acelerações lineares	26
5.4.3	Aceleração angular	26
5.4.4	Equações da dinâmicas do robô PRP	27
5.5	Resultados de simulação	27
6	Análise dos resultados	28
7	Programas feitos em <i>Matlab</i>	29
7.1	Simulação da trajetória	29
7.2	Cinemática inversa	29
7.3	Plotagem dos gráficos da cinemática inversa	30
7.4	Exportação dos dados da cinemática inversa para o <i>SolidWorks</i>	32
7.5	Equações da dinâmica por <i>Newton-Euler</i>	32
7.6	Equações da dinâmica por <i>Gibbs-Appell</i>	35
7.7	Dinâmica inversa	38

8	Distribuição das tarefas	39
9	Referências	40

1 Introdução

O presente trabalho possuiu como objetivo projetar e desenvolver modelos e simulações de um robô manipulador que opere no espaço tridimensional, conforme a Figura 1.

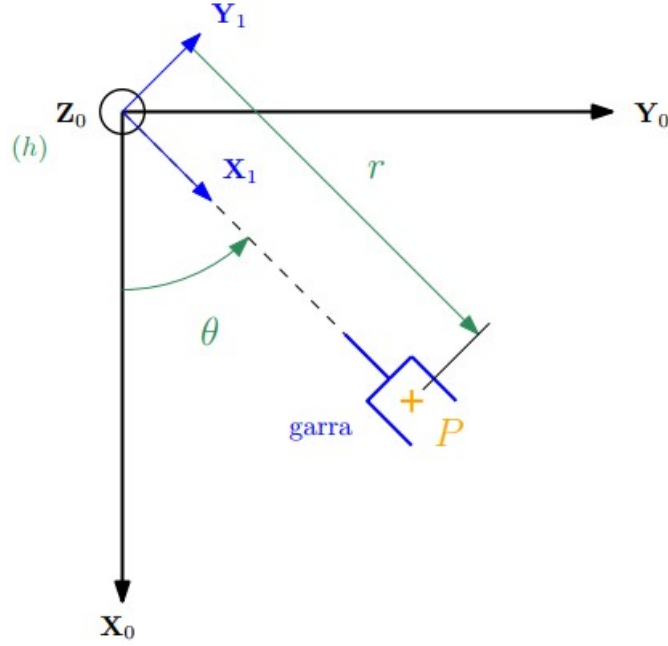


Figura 1: Mecanismo a ser modelado.

A garra do robô, ou efetuator, deve ser capaz de realizar os seguintes deslocamentos: linear h ao longo do eixo Z_0 ; angular θ em torno do eixo Z_0 ; linear r ao longo do eixo X_1 . O mecanismo é de cadeia aberta. É realizado a síntese do tipo, apresenta-se o modelo do robô escolhido e as análises cinemática inversa e dinâmica do modelo.

Para a simulação, o efetuator deverá realizar uma trajetória retilínea com as coordenadas inicial e final especificadas. Admite-se que o robô é capaz de transportar um objeto de massa 1kg. Considerou-se que a direção do eixo Z_0 é a vertical, ou seja, aquela em que atua a força gravitacional.

O mecanismo possui como requisitos os parâmetros abaixo:

- Posição inicial ($P_{inicial}$) = (200,-100,0) [mm]
- Posição final (P_{final}) = (200,100,50) [mm]
- Tempo de simulação (T) = 10 [s]

A trajetória entre $P_{inicial}$ e P_{final} deve ser retilínea e executada no tempo de simulação.

2 Trajetória do objeto carregado

2.1 Interpolação da trajetória

A trajetória foi interpolada por um polinômio do 5º grau. Dessa forma, a função posição em função do tempo é do tipo:

$$y(t) = a_0 t^5 + a_1 t^4 + a_2 t^3 + a_3 t^2 + a_4 t + a_5 \quad (1)$$

As respectivas derivadas para obtenção das expressões da velocidade e aceleração são mostradas abaixo:

Velocidade:

$$y'(t) = 5a_0 t^4 + 4a_1 t^3 + 3a_2 t^2 + 2a_3 t + a_4 \quad (2)$$

Aceleração:

$$y''(t) = 20a_0 t^3 + 12a_1 t^2 + 2a_2 t + 2a_3 \quad (3)$$

As condições iniciais e finais impostas pelo enunciado são:

Condições Iniciais:

$$y(0) = P_{inicial}, \quad y'(0) = 0, \quad y''(0) = 0$$

Condições Finais:

$$y(T) = P_{final}, \quad y'(T) = 0, \quad y''(T) = 0$$

Aplicando as condições iniciais nas equações (1), (2) e (3) obtêm-se os valores de a_3 , a_4 e a_5 como se segue:

$$y(0) = a_5 = P_{inicial}, \quad y'(0) = a_4 = 0, \quad y''(0) = 2a_3 = 0$$

Analogamente, aplicando as condições finais, obtêm-se:

$$y(T) = P_{final} = a_0 T^5 + a_1 T^4 + a_2 T^3 = P_{inicial} \quad (4)$$

$$y'(T) = 5a_0 T^4 + 4a_1 T^3 + 3a_2 T^2 = 0 \quad (5)$$

$$y''(T) = 20a_0 T^3 + 12a_1 T^2 + 6a_2 T = 0 \quad (6)$$

Fazendo $P_{final} - P_{inicial} = L$, as equações (4), (5) e (6) podem ser escritas na forma matricial da seguinte forma:

$$\begin{bmatrix} 1 & 1 & 1 \\ 5 & 4 & 3 \\ 20 & 12 & 6 \end{bmatrix} \begin{bmatrix} a_0 T^3 \\ a_1 T^2 \\ a_2 T \end{bmatrix} = \begin{bmatrix} \frac{L}{T^2} \\ 0 \\ 0 \end{bmatrix} \quad (7)$$

Resolvendo o sistema de equações, chega-se às expressões finais:

Posição:

$$y(t) = L \left(6 \left(\frac{t}{T} \right)^5 - 15 \left(\frac{t}{T} \right)^4 + 10 \left(\frac{t}{T} \right)^3 \right) \quad (8)$$

Velocidade:

$$y'(t) = \frac{L}{T} \left(30 \left(\frac{t}{T} \right)^4 - 60 \left(\frac{t}{T} \right)^3 + 30 \left(\frac{t}{T} \right)^2 \right) \quad (9)$$

Aceleração:

$$y''(t) = \frac{L}{T^2} \left(120 \left(\frac{t}{T} \right)^3 - 180 \left(\frac{t}{T} \right)^2 + 60 \left(\frac{t}{T} \right) \right) \quad (10)$$

2.2 Simulação da trajetória

O comportamento da trajetória foi simulado num programa de *MATLAB*, que se encontra na seção 7.1, e o gráfico abaixo mostra os resultados:

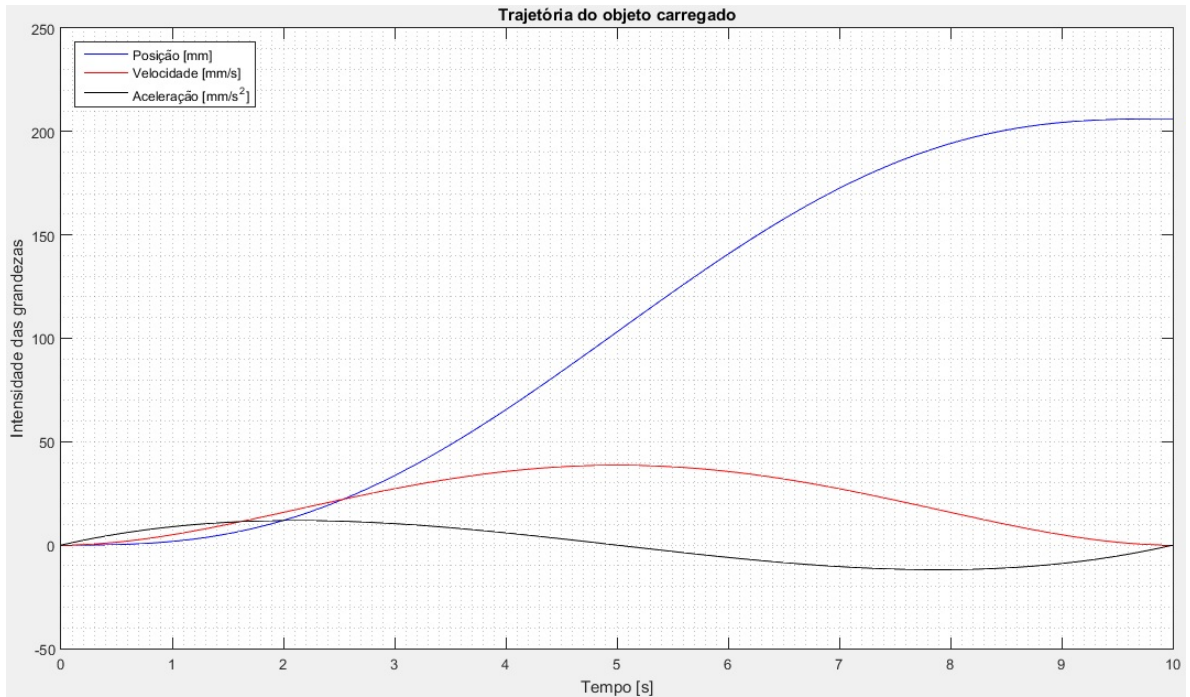


Figura 2: Simulação da trajetória

3 Síntese do modelo preliminar

Segundo o critério de *Kutzbach-Gruebler*, para o espaço de movimentação tridimensional $\lambda = 6$, tem-se que a mobilidade M é dada por:

$$M = 6(N_e - 1) - 5n_1 - 4n_2 - 3n_3 - 4n_4 - n_5$$

em que n_j é número de pares cinemáticos que permitem j graus de liberdade e N_e é o número de elos. Admite-se que a cadeia do mecanismo contém apenas pares cinemáticos de 1 gdl. Como a mobilidade do mecanismo é $M = 3$, pois deve realizar dois movimentos lineares e um angular, obtém-se:

$$6N_e = 5n_1 + 9$$

Dentre as soluções possíveis, a que apresenta o menor número de pares cinemáticos é o caso em que $n_1 = 3$ e $N_e = 4$. Para $n_1 = 3$ há 8 topologias possíveis, desde PPP a RRR. Pelos requisitos de deslocamento do mecanismo a ser modelado, a solução adotada foi do tipo PRP, que apresenta duas juntas prismáticas (P) e uma de revolução (R). Essa topologia considera uma junta de revolução em torno do eixo Z_0 , uma junta prismática ao longo do eixo do braço, na direção do eixo X_1 e uma junta prismática na direção de Z_0 . O modelado mecanimos encontra-se na Figura 3.

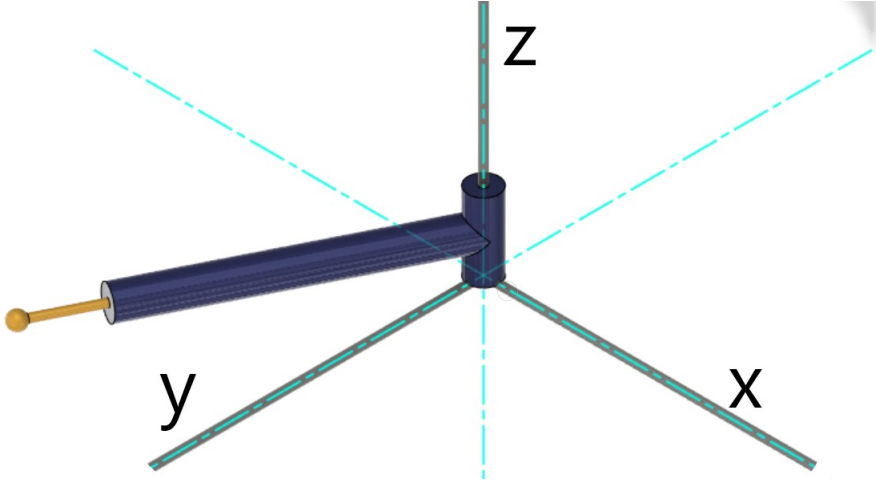


Figura 3: Modelo de mecanismo adotado.

4 Modelo cinemático

4.1 Esquema simplificado do mecanismo

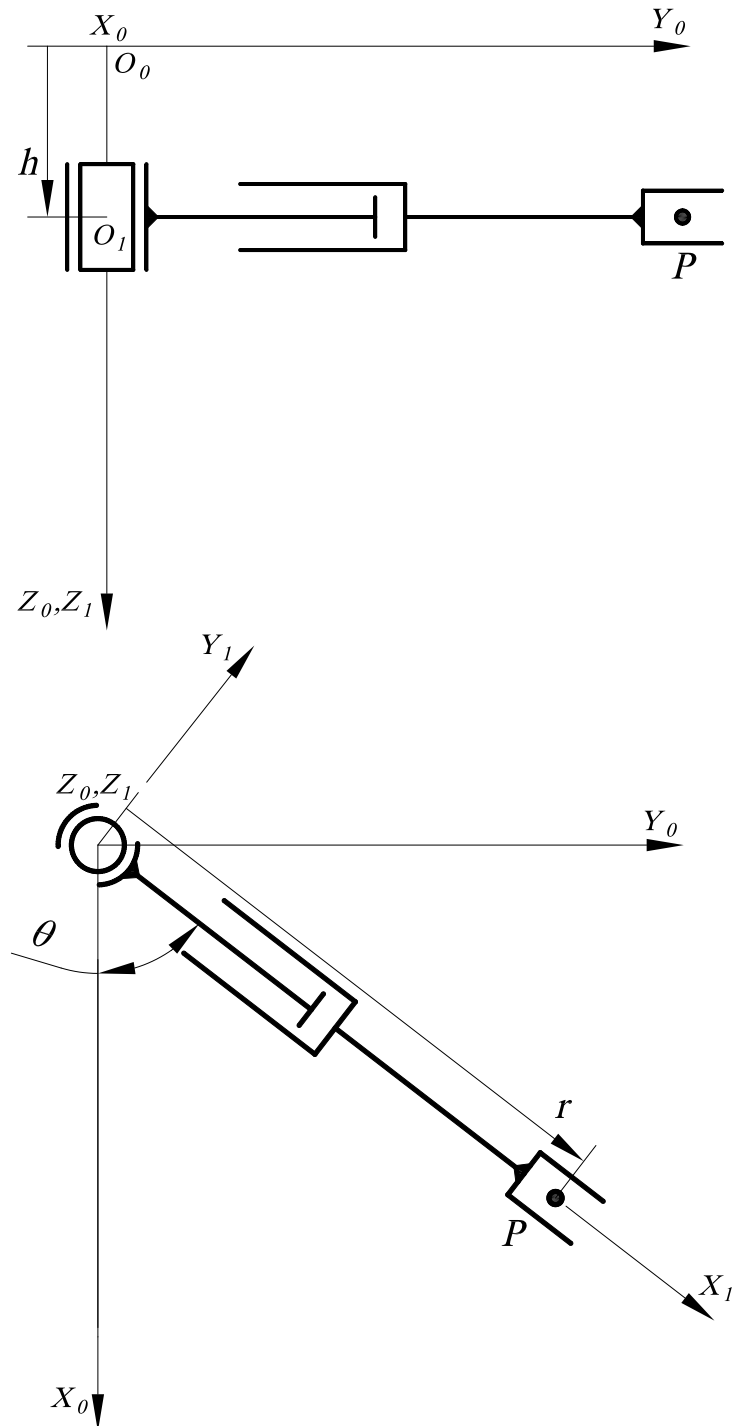


Figura 4: Esquema cinemático do mecanismo.

4.2 Obtenção da posição, velocidade e aceleração da garra do robô em função das coordenadas generalizadas

Para simplificar o modelo cinemático, definimos 3 coordenadas generalizadas: r , θ e h para definir a posição da garra do robô (ponto P). Elas podem ser vistas na Figura 4. Para obter a posição no sistema de coordenadas fixo a terra ($X_0Y_0Z_0$), vamos definir um sistema de coordenadas móvel solidário ao braço do robô ($X_1Y_1Z_1$). Utilizando a matriz de transformação de coordenadas do referencial solidário ao braço do robô com respeito ao referencial inercial (0_1T), obtemos a seguinte relação

$$\begin{bmatrix} {}^0\mathbf{r}_P \\ 1 \end{bmatrix} = {}^0_1T * \begin{bmatrix} {}^1\mathbf{r}_P \\ 1 \end{bmatrix} \quad (11)$$

Seguindo o esquema cinemático da Figura 4, obtemos a matriz de transformação de coordenadas e a posição do ponto P relativo ao sistema móvel ($X_1Y_1Z_1$).

$${}^0_1T = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & h \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^1\mathbf{r}_P = \begin{bmatrix} r \\ 0 \\ 0 \end{bmatrix}$$

Substituindo esses valores na relação (11), obtemos

$$\begin{bmatrix} {}^0\mathbf{r}_P \\ 1 \end{bmatrix} = \begin{bmatrix} r\cos(\theta) \\ r\sin(\theta) \\ h \\ 1 \end{bmatrix}$$

Finalmente, temos a posição da garra com respeito ao sistema fixo a terra (${}^0\mathbf{r}_P = \mathbf{r}_P$) em função das coordenadas generalizadas

$$\mathbf{r}_P = \begin{bmatrix} r\cos(\theta) \\ r\sin(\theta) \\ h \end{bmatrix} \quad (12)$$

Derivando a expressão (12) em relação ao tempo, obtemos a velocidade em função das coordenadas generalizadas e derivadas:

$$\mathbf{v}_P = \frac{d\mathbf{r}_P}{dt} = \begin{bmatrix} \dot{r}\cos(\theta) - r\sin(\theta)\dot{\theta} \\ \dot{r}\sin(\theta) + r\cos(\theta)\dot{\theta} \\ \dot{h} \end{bmatrix} \quad (13)$$

Derivando mais uma vez, obtemos uma expressão para a aceleração em função das coordenadas generalizadas e derivadas:

$$\mathbf{a}_P = \frac{d^2\mathbf{r}_P}{dt^2} = \begin{bmatrix} \ddot{r}\cos(\theta) - 2\dot{r}\sin(\theta)\dot{\theta} - r\cos(\theta)\dot{\theta}^2 - r\sin(\theta)\ddot{\theta} \\ \ddot{r}\sin(\theta) + 2\dot{r}\cos(\theta)\dot{\theta} - r\sin(\theta)\dot{\theta}^2 + r\cos(\theta)\ddot{\theta} \\ \ddot{h} \end{bmatrix} \quad (14)$$

4.3 Análise da cinemática inversa da trajetória do ponto P

Conforme a introdução deste relatório, a trajetória do ponto P é conhecida para 10 segundos, mas não são conhecidas os valores das coordenadas generalizadas, assim temos um problema de cinemática inversa. Ou seja, temos um sistema de equações não lineares para certos pontos da trajetória do ponto P onde as variáveis são r , θ e h . Para resolver este sistema, vamos a utilizar o método numérico de Newton-Raphson.

4.3.1 Obtenção das coordenadas generalizadas na trajetória

Antes de começar a resolução dos sistemas de equações, precisamos obter as posições, velocidades e acelerações dos pontos intermediários da trajetória posto que, assim, vamos utilizar as expressões do polinômio interpolador e suas derivadas obtidas na introdução deste relatório (expressões (8), (9) e (10)):

$$\begin{aligned}y(t) &= L \cdot \left(6 \left(\frac{t}{T} \right)^5 - 15 \left(\frac{t}{T} \right)^4 + 10 \left(\frac{t}{T} \right)^3 \right) \\y'(t) &= \frac{L}{T} \cdot \left(30 \left(\frac{t}{T} \right)^4 - 60 \left(\frac{t}{T} \right)^3 + 30 \left(\frac{t}{T} \right)^2 \right) \\y''(t) &= \frac{L}{T^2} \cdot \left(120 \left(\frac{t}{T} \right)^3 - 180 \left(\frac{t}{T} \right)^2 + 60 \left(\frac{t}{T} \right) \right)\end{aligned}$$

Utilizando a definição da reta no espaço tridimensional, temos

$$\mathbf{r}_P = \begin{bmatrix} r_{p\ x} \\ r_{p\ y} \\ r_{p\ z} \end{bmatrix} = \mathbf{r}_{P\ inicial} + y(t) \cdot \mathbf{v} \quad (15)$$

Onde \mathbf{v} é o vetor unitário da reta de comprimento L que define a trajetória do ponto P , e é definido da seguinte maneira:

$$\mathbf{v} = \frac{1}{L} \cdot (\mathbf{r}_{P\ final} - \mathbf{r}_{P\ inicial})$$

Igualando termo a termo as equações vetoriais (12) e (15), obtemos:

$$\begin{cases} r \cos(\theta) = r_{p\ x} \\ r \sin(\theta) = r_{p\ y} \\ h = r_{p\ z} \end{cases}$$

Onde os valores da direita são conhecidos posto que foram encontrados com a expressão (15). Esse é um sistema não linear com variáveis r , θ e h , o qual pode ser resolvido pelo método de Newton-Raphson.

4.3.2 Obtenção das velocidades generalizadas na trajetória

Derivando a equação (15) em relação ao tempo, obtemos a seguinte relação para a velocidade:

$$\mathbf{v}_P = \frac{d\mathbf{r}_P}{dt} = \begin{bmatrix} v_{p\ x} \\ v_{p\ y} \\ v_{p\ z} \end{bmatrix} = y'(t) \cdot \mathbf{v} \quad (16)$$

Igualando termo a termo a expressão (13) com a (16), obtemos o seguinte sistema de equações:

$$\begin{cases} \dot{r}\cos(\theta) - r\sin(\theta)\dot{\theta} = v_{p\ x} \\ \dot{r}\sin(\theta) + r\cos(\theta)\dot{\theta} = v_{p\ y} \\ \dot{h} = v_{p\ z} \end{cases}$$

Onde os valores da direita são conhecidos posto que foram encontrados com a expressão (16). Assim, temos um sistema linear onde as variáveis são \dot{r} , $\dot{\theta}$ e \dot{h} , sendo que os valores de r , θ e h já foram encontrados na subseção anterior. Na forma matricial do sistema, temos:

$$\begin{bmatrix} \cos(\theta) & -r\sin(\theta) & 0 \\ \sin(\theta) & r\cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{r} \\ \dot{\theta} \\ \dot{h} \end{bmatrix} = \begin{bmatrix} v_{p\ x} \\ v_{p\ y} \\ v_{p\ z} \end{bmatrix}$$

Aplicando a inversa para obter as velocidades generalizadas:

$$\begin{bmatrix} \dot{r} \\ \dot{\theta} \\ \dot{h} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -r\sin(\theta) & 0 \\ \sin(\theta) & r\cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} v_{p\ x} \\ v_{p\ y} \\ v_{p\ z} \end{bmatrix}$$

4.3.3 Obtenção das acelerações generalizadas da trajetória

Derivando a equação (16) obtemos a seguinte relação para a aceleração:

$$\mathbf{a}_P = \frac{d^2 \mathbf{r}_P}{dt^2} = \begin{bmatrix} a_{p\ x} \\ a_{p\ y} \\ a_{p\ z} \end{bmatrix} = \mathbf{y}''(t) \cdot \mathbf{v} \quad (17)$$

Igualando termo a termo a expressão (14) com a (17), obtemos o seguinte sistema de equações:

$$\begin{cases} \ddot{r}\cos(\theta) - 2\dot{r}\sin(\theta)\dot{\theta} - r\cos(\theta)\dot{\theta}^2 - r\sin(\theta)\ddot{\theta} = a_{p\ x} \\ \ddot{r}\sin(\theta) + 2\dot{r}\cos(\theta)\dot{\theta} - r\sin(\theta)\dot{\theta}^2 + r\cos(\theta)\ddot{\theta} = a_{p\ y} \\ \ddot{h} = a_{p\ z} \end{cases}$$

Onde os valores da direita são conhecidos posto que foram encontrados com a expressão (17). Ordenando o sistema de equações fazendo com que os termos conhecidos, que foram obtidos nas subseções anteriores, fiquem na direita das equações, obtemos:

$$\begin{cases} \ddot{r}\cos(\theta) - r\sin(\theta)\ddot{\theta} = a_{p\ x} + 2\dot{r}\sin(\theta)\dot{\theta} + r\cos(\theta)\dot{\theta}^2 \\ \ddot{r}\sin(\theta) + r\cos(\theta)\ddot{\theta} = a_{p\ y} - 2\dot{r}\cos(\theta)\dot{\theta} + r\sin(\theta)\dot{\theta}^2 \\ \ddot{h} = a_{p\ z} \end{cases}$$

Outra vez temos um sistema linear de equações. Na forma matricial, temos:

$$\begin{bmatrix} \cos(\theta) & -r\sin(\theta) & 0 \\ \sin(\theta) & r\cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \ddot{r} \\ \ddot{\theta} \\ \ddot{h} \end{bmatrix} = \begin{bmatrix} a_{p\ x} + 2\dot{r}\sin(\theta)\dot{\theta} + r\cos(\theta)\dot{\theta}^2 \\ a_{p\ y} - 2\dot{r}\cos(\theta)\dot{\theta} + r\sin(\theta)\dot{\theta}^2 \\ a_{p\ z} \end{bmatrix}$$

Aplicando a inversa para obter as acelerações generalizadas:

$$\begin{bmatrix} \ddot{r} \\ \ddot{\theta} \\ \ddot{h} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -r\sin(\theta) & 0 \\ \sin(\theta) & r\cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} a_{p\ x} + 2\dot{r}\sin(\theta)\dot{\theta} + r\cos(\theta)\dot{\theta}^2 \\ a_{p\ y} - 2\dot{r}\cos(\theta)\dot{\theta} + r\sin(\theta)\dot{\theta}^2 \\ a_{p\ z} \end{bmatrix}$$

4.4 Resultados numéricos da cinemática inversa

As simulações numéricas para determinar os valores das posições, velocidade e acelerações generalizadas foram feitas através de um programa em Matlab, que se encontram nas seções 7.2 e 7.3. Resultados das simulações:

Coordenada generalizada r :

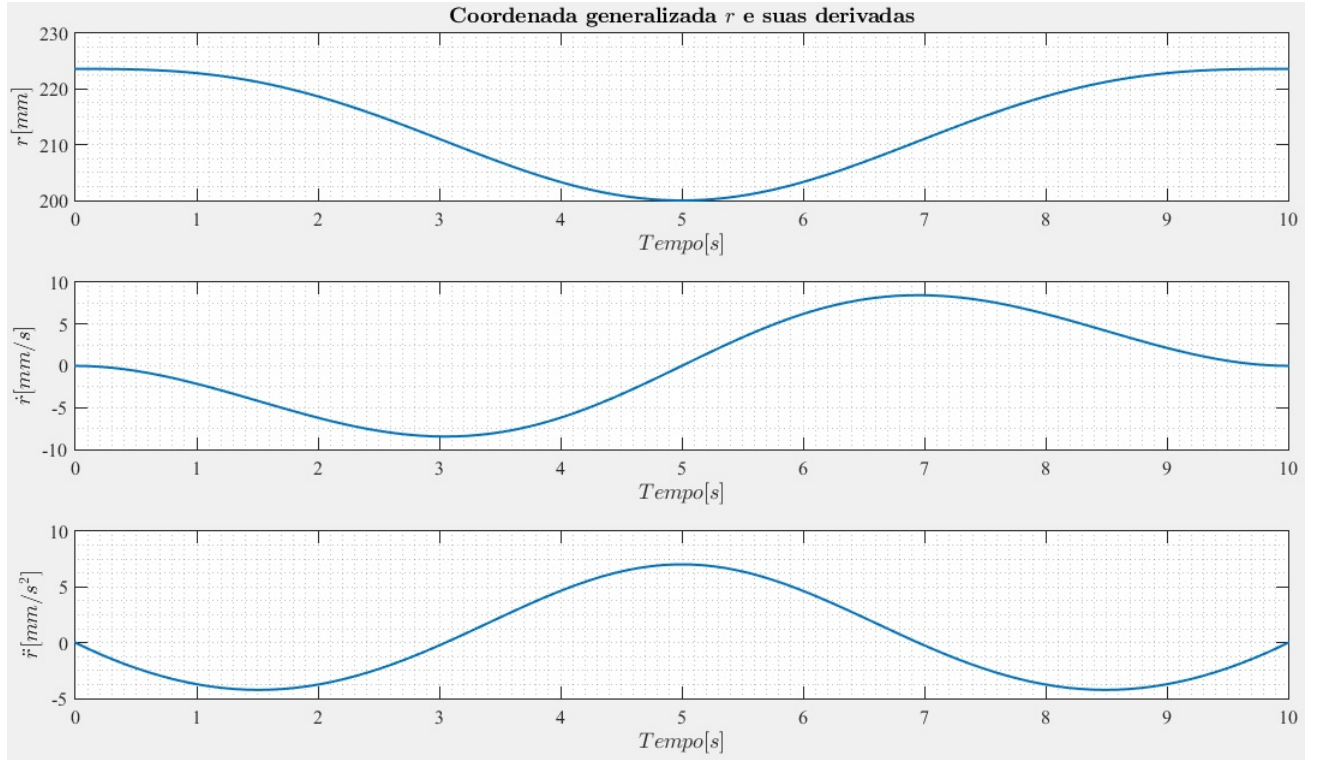


Figura 5: Coordenada generalizada r e suas derivadas em função do tempo

Coordenada generalizada θ :

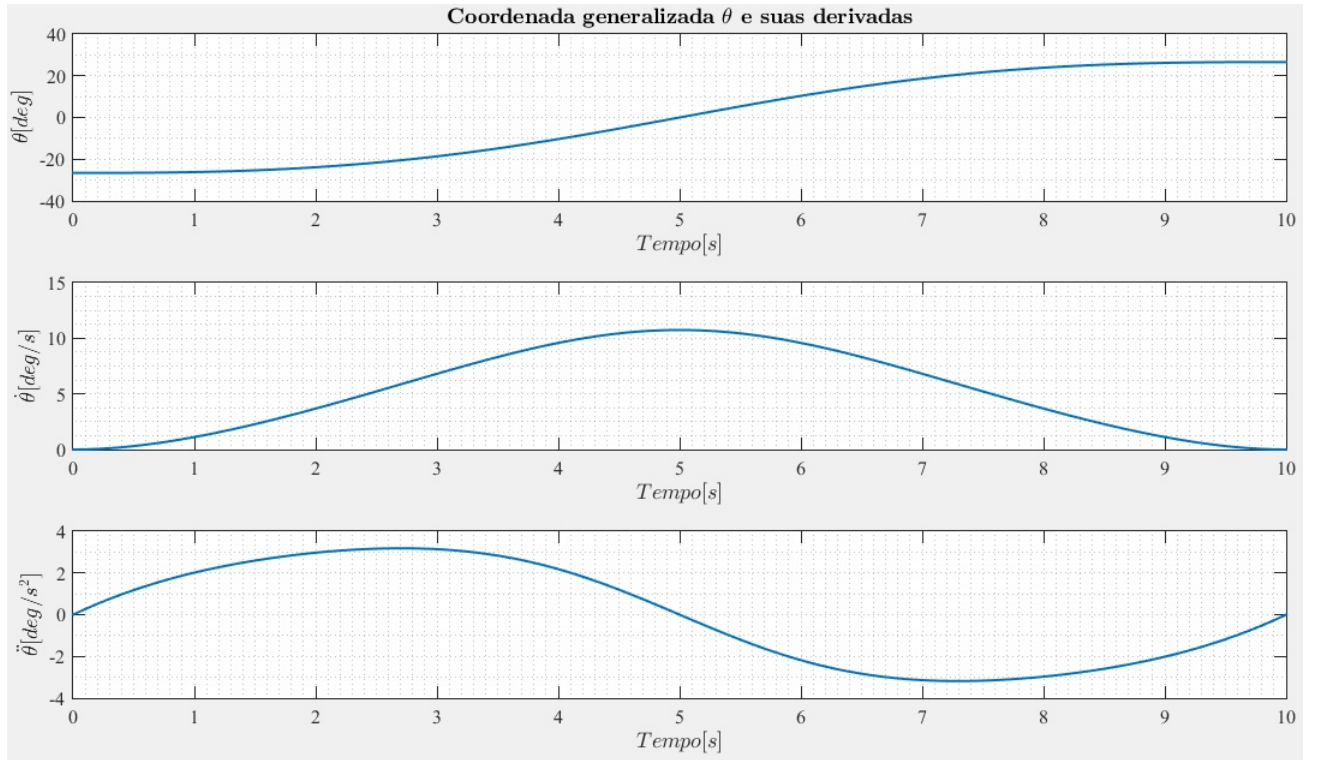


Figura 6: Coordenada generalizada θ e suas derivadas em função do tempo

Coordenada generalizada h :

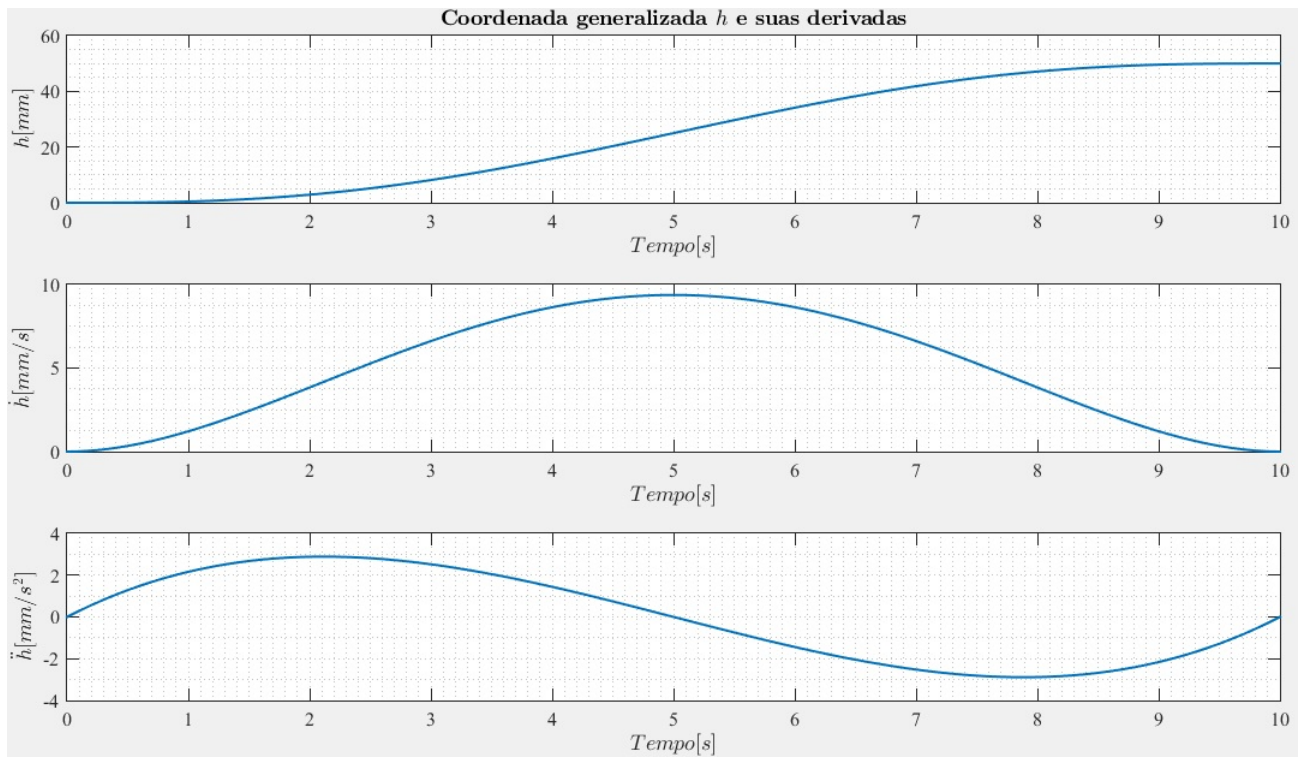


Figura 7: Coordenada generalizada h e suas derivadas em função do tempo

5 Dinâmica

5.1 Seleção de materiais

Os braços robóticos industriais comumente são feitos de um aço de propósito geral, ou seja, algum aço da família AISI 10XX. Dentro desta família, os aços mais utilizados para projeto estrutural são: 1010, 1015, 1018 e 1020. Os sites abaixo mostram que todos eles têm a mesma densidade de $7.87g/cm^3$ ou $7870kg/m^3$, então com relação ao peso da estrutura não faz diferença qual escolher.

AISI 1010: [⟨https://www.azom.com/article.aspx?ArticleID=6539⟩](https://www.azom.com/article.aspx?ArticleID=6539)

AISI 1015: [⟨https://www.azom.com/article.aspx?ArticleID=6579⟩](https://www.azom.com/article.aspx?ArticleID=6579)

AISI 1018: [⟨https://www.azom.com/article.aspx?ArticleID=6115⟩](https://www.azom.com/article.aspx?ArticleID=6115)

AISI 1020: [⟨https://www.azom.com/article.aspx?ArticleID=9145⟩](https://www.azom.com/article.aspx?ArticleID=9145)

Foi definido que todas as peças mecânicas do braço têm essa mesma densidade. Entretanto, ainda é necessário definir os materiais dos motores de passo. Geralmente a estrutura dos motores é feita de aço inoxidável e o rotor é feito de aço silício (silicon steel ou electrical steel). Como mostra a tabela no site da *Wikipedia* abaixo, a densidade de um aço inoxidável varia de 7.7 a $8.0g/m^3$. Enquanto, como mostra o segundo site da *Wikipedia* abaixo, a densidade do aço silício é aproximadamente $7.65g/m^3$.

Aço inoxidável: [⟨https://en.wikipedia.org/wiki/Stainless_steel⟩](https://en.wikipedia.org/wiki/Stainless_steel)

Aço silício: [⟨https://en.wikipedia.org/wiki/Electrical_steel⟩](https://en.wikipedia.org/wiki/Electrical_steel)

Como todos esses dois valores de densidade são próximos do valor de densidade dos aços de propósito geral e os motores móveis têm um volume pequeno comparado as peças mecânicas móveis, então não é um erro grotesco considerar todos os elementos com a densidade de $7.87g/cm^3$. Assim, no *SolidWorks*, todas as peças são configuradas com o material AISI 1020 Steel Cold Rolled

5.2 Modelo em CAD

Modelo 3D do robô PRP de cadeia aberta e juntas atuadas feito no SolidWorks 2018:

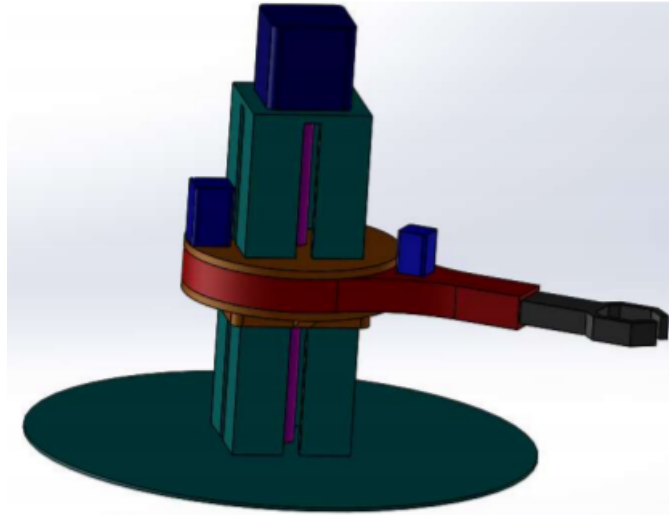


Figura 8: Modelo do robô PRP

5.2.1 Elo 1

A peça do elo 1 é o suporte do mecanismo:

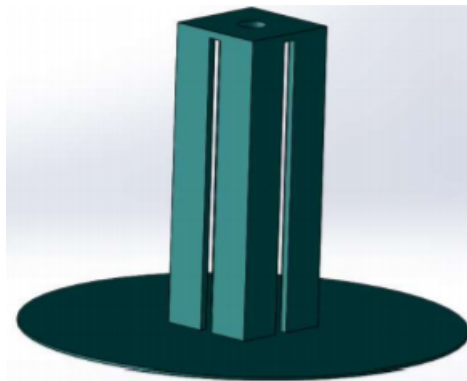


Figura 9: Elo 1

```

Mass properties of Base_Atuador_P1
Configuration: Default
Coordinate system: -- default --

Density = 0.01 grams per cubic millimeter

Mass = 3024.45 grams

Volume = 384300.87 cubic millimeters

Surface area = 247942.47 square millimeters

Center of mass: ( millimeters )
X = 0.00
Y = 0.00
Z = 68.96

Principal axes of inertia and principal moments of inertia: ( grams * square millimeters )
Taken at the center of mass.
lx = ( 0.00, 0.00, 1.00) Px = 15236538.32
ly = ( 0.00, -1.00, 0.00) Py = 23722011.30
lz = ( 1.00, 0.00, 0.00) Pz = 23722011.30

Moments of inertia: ( grams * square millimeters )
Taken at the center of mass and aligned with the output coordinate system.
Lxx = 23722011.30 Lxy = 0.00 Lxz = 0.00
Lyx = 0.00 Lyy = 23722011.30 Lyz = 0.00
Lzx = 0.00 Lzy = 0.00 Lzz = 15236538.32

Moments of inertia: ( grams * square millimeters )
Taken at the output coordinate system.
box = 38105155.88 boxy = 0.00 boxz = 0.00
lyx = 0.00 lyy = 38105155.88 lyz = 0.00
lzx = 0.00 lzy = 0.00 lzz = 15236538.32

```

Figura 10: Propriedades mecânicas do elo 1

Nesta peça está ocochado um motor de passo para transladar o elo 2 em relação ao elo 1 através da rotação de um fuso:

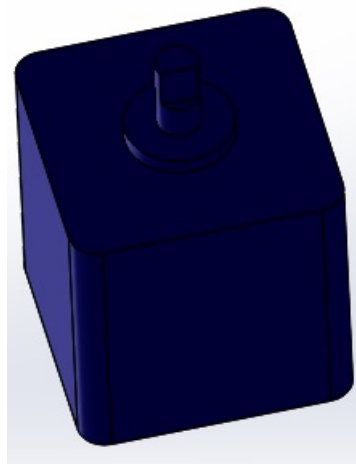


Figura 11: Motor de passo da junta 12

Mass properties of Step_Motor_Big		
Configuration: Default		
Coordinate system: -- default --		
Density = 0.01 grams per cubic millimeter		
Mass = 988.04 grams		
Volume = 125544.84 cubic millimeters		
Surface area = 15081.77 square millimeters		
Center of mass: (millimeters)		
X = 0.00		
Y = 0.00		
Z = 25.39		
Principal axes of inertia and principal moments of inertia: (grams * square millimeters)		
Taken at the center of mass.		
lx = (0.00, 0.00, 1.00)	Px = 400658.65	
ly = (1.00, 0.00, 0.00)	Py = 415227.23	
lz = (0.00, 1.00, 0.00)	Pz = 415231.03	
Moments of inertia: (grams * square millimeters)		
Taken at the center of mass and aligned with the output coordinate system.		
Lxx = 415227.23	Lxy = 0.00	Lxz = 0.00
Lyx = 0.00	Lyx = 415230.72	Lyx = 67.37
Lzx = 0.00	Lzy = 67.37	Lzz = 400658.96
Moments of inertia: (grams * square millimeters)		
Taken at the output coordinate system.		
lxx = 1052053.43	lxy = 0.00	lxz = 0.00
lyx = 0.00	lyy = 1052056.92	lyz = 111.67
lzx = 0.00	lzy = 111.67	lzz = 400658.96

Figura 12: Propriedades mecânicas do motor de passo da junta 12

A distância do parte superior do motor de passo até a base do elo 1 mede 257mm. E o diâmetro da base do elo 1 mede 300mm.

5.2.2 Elo 2

A peça do elo 2 forma uma junta prismática com o elo 1:

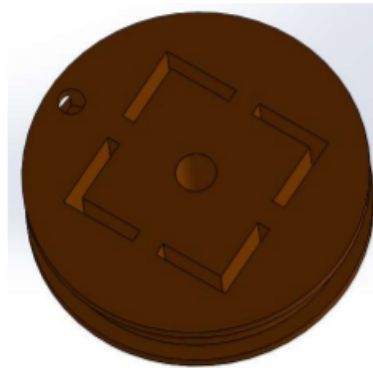


Figura 13: Elo 2

```

Mass properties of Atuador_P1_Base_R
Configuration: Default
Coordinate system: -- default --

Density = 0.01 grams per cubic millimeter

Mass = 2782.89 grams

Volume = 353607.36 cubic millimeters

Surface area = 63695.29 square millimeters

Center of mass: ( millimeters )
X = 0.17
Y = 0.00
Z = 1.68

Principal axes of inertia and principal moments of inertia: ( grams * square m
Taken at the center of mass.
lx = ( 0.00, 1.00, 0.00) Px = 2752654.31
ly = (-1.00, 0.00, 0.00) Py = 2777719.79
lz = ( 0.00, 0.00, 1.00) Pz = 4841499.36

Moments of inertia: ( grams * square millimeters )
Taken at the center of mass and aligned with the output coordinate system.
Lxx = 2777732.52 Lxy = 0.00 Lxz = 5124.86
Lyx = 0.00 Lyy = 2752654.31 Lyz = 0.00
Lzx = 5124.86 Lzy = 0.00 Lzz = 4841486.63

Moments of inertia: ( grams * square millimeters )
Taken at the output coordinate system.
lxx = 2785599.63 lxy = 0.00 lxx = 5907.72
lyx = 0.00 lyy = 2760599.33 lyz = 0.00
lzx = 5907.72 lzy = 0.00 lzz = 4841564.54

```

Figura 14: Propriedades mecânicas do elo 2

Nesta peça está ocoplado um motor de passo para girar o elo 3 em relação ao elo 2:

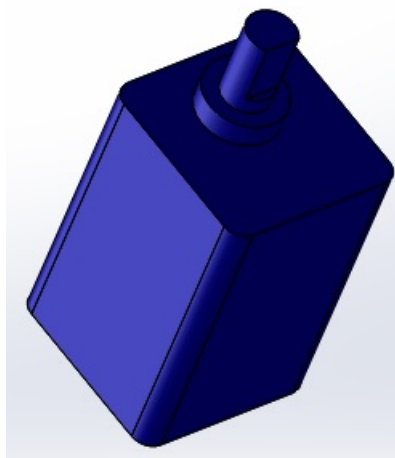


Figura 15: Motor de passo da junta 23

Mass properties of Step_Motor_Medium		
Configuration: Default		
Coordinate system: -- default--		
Density = 0.01 grams per cubic millimeter		
Mass = 148.35 grams		
Volume = 18849.65 cubic millimeters		
Surface area = 4408.86 square millimeters		
Center of mass: (millimeters)		
X = 0.00		
Y = 0.09		
Z = 18.09		
Principal axes of inertia and principal moments of inertia: (grams * square m		
Taken at the center of mass.		
lx = (0.00, 0.03, 1.00)	Px = 12616.86	
ly = (0.00, -1.00, 0.03)	Py = 22992.53	
lz = (1.00, 0.00, 0.00)	Pz = 23031.98	
Moments of inertia: (grams * square millimeters)		
Taken at the center of mass and aligned with the output coordinate system.		
Lxx = 23031.98	Lxy = 0.00	Lxz = 0.00
Lyx = 0.00	Lyy = 22983.38	Lyx = 308.03
Lzx = 0.00	Lzy = 308.03	Lzz = 12626.01
Moments of inertia: (grams * square millimeters)		
Taken at the output coordinate system.		
box = 71553.53	bxy = 0.00	bxz = 0.00
byx = 0.00	byy = 71503.59	byz = 562.40
bzx = 0.00	bzy = 562.40	bzz = 12627.34

Figura 16: Propriedades mecânicas do motor de passo da junta 23

O diâmetro da peça do elo 2 mede 127mm e a distância entre a superfície inferior e a superior mede 40mm.

5.2.3 Elo 3

A peça do elo 3 forma uma junta de rotação com o elo 2:



Figura 17: Elo 3

```

Mass properties of Atuator_R_Base_P2
Configuration: Default
Coordinate system: -- default --

Density = 0.01 grams per cubic millimeter

Mass = 622.67 grams

Volume = 79118.81 cubic millimeters

Surface area = 36101.31 square millimeters

Center of mass: ( millimeters )
X = 0.00
Y = -40.37
Z = -0.03

Principal axes of inertia and principal moments of inertia: ( grams * square m
Taken at the center of mass.
Ix = ( 0.00, 1.00, 0.00) Px = 747232.57
Iy = (-1.00, 0.00, 0.00) Py = 2259824.36
Iz = ( 0.00, 0.00, 1.00) Pz = 2957602.52

Moments of inertia: ( grams * square millimeters )
Taken at the center of mass and aligned with the output coordinate system.
Lxx = 2259824.36 Lxy = 0.00 Lxz = 0.00
Lyx = 0.00 Lyy = 747232.72 Lyz = 585.18
Lzx = 0.00 Lzy = 585.18 Lzz = 2957602.37

Moments of inertia: ( grams * square millimeters )
Taken at the output coordinate system.
lxx = 3274640.05 lxy = 0.00 lxz = 0.00
lyx = 0.00 lyy = 747233.12 lyz = 1221.50
lzx = 0.00 lzy = 1221.50 lzz = 3972417.66

```

Figura 18: Propriedades mecânicas do elo 3

Nesta peça está acoplado um motor de passo para transladar o elo 4 em relação ao elo 3:

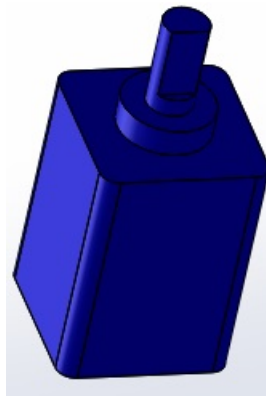


Figura 19: Motor de passo da junta 23

Mass properties of Step_Motor_Small		
Configuration: Default		
Coordinate system: -- default --		
Density = 0.01 grams per cubic millimeter		
Mass = 66.42 grams		
Volume = 8440.07 cubic millimeters		
Surface area = 2611.42 square millimeters		
Center of mass: (millimeters)		
X = 0.00		
Y = 0.01		
Z = 13.31		
Principal axes of inertia and principal moments of inertia: (grams * square millimeters)		
Taken at the center of mass.		
lx = (0.00, 0.00, 1.00)	Px = 3388.49	
ly = (1.00, 0.00, 0.00)	Py = 5886.81	
lz = (0.00, 1.00, 0.00)	Pz = 5887.34	
Moments of inertia: (grams * square millimeters)		
Taken at the center of mass and aligned with the output coordinate system.		
Lxx = 5886.81	Lxy = 0.00	Lxz = 0.00
Lyx = 0.00	Lyx = 5887.32	Lyx = 6.95
Lzx = 0.00	Lzy = 6.95	Lzz = 3388.51
Moments of inertia: (grams * square millimeters)		
Taken at the output coordinate system.		
lxx = 17655.49	lxy = 0.00	lxz = 0.00
lyx = 0.00	lyy = 17655.99	lyz = 11.42
lzx = 0.00	lzy = 11.42	lzz = 3388.52

Figura 20: Propriedades mecânicas do motor de passo da junta 23

O diâmetro interno da peça do elo 3 na junta 23 mede 30mm, a sua espessura mede 20mm e a distância da extremidade até o cento da junta 23 mede 150mm.

5.2.4 Elo 4

A peça do elo 4 forma uma junta prismática com o elo 3, além disso ela tem uma garra na ponta para segurar o objeto carregado:

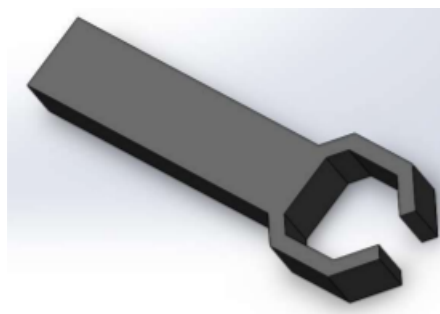


Figura 21: Elo 4

Mass properties of Atuador_P2		
Configuration: Default		
Coordinate system: -- default --		
Density = 0.0079 grams per cubic millimeter		
Mass = 214.2522 grams		
Volume = 27223.9192 cubic millimeters		
Surface area = 17295.8418 square millimeters		
Center of mass: (millimeters)		
X = 70.2348		
Y = 0.0000		
Z = 0.0000		
Principal axes of inertia and principal moments of inertia: (grams * square m		
Taken at the center of mass.		
lx = (1.0000, 0.0000, 0.0000)	Px = 27201.7958	
ly = (0.0000, 0.0000, -1.0000)	Py = 35198.9371	
lz = (0.0000, 1.0000, 0.0000)	Pz = 369708.6102	
Moments of inertia: (grams * square millimeters)		
Taken at the center of mass and aligned with the output coordinate system.		
Lxx = 27201.7958	Lxy = 0.0000	Lxz = 0.0000
Lyx = 0.0000	Lyx = 369708.6102	Lyz = 0.0000
Lzx = 0.0000	Lzy = 0.0000	Lzz = 35198.9371
Moments of inertia: (grams * square millimeters)		
Taken at the output coordinate system.		
lxx = 27201.7958	lxy = 0.0000	lxz = 0.0000
lyx = 0.0000	lyy = 1426600.7074	lyz = 0.0000
lzx = 0.0000	lzy = 0.0000	lzz = 1408891.034

Figura 22: Propriedades mecânicas do elo 4

A distância da extremidade da peça do elo 4 até o objeto carregado é de 120mm e a sua espessura mede 14mm.

5.2.5 Alcance da garra e região de trabalho

O conjunto do alcance radial necessário da garra é $[200, 223.6068]mm$. Quando o elo 4 está completamente retraído no elo 3, o centro da garra, onde seria o centro do objeto, está a 180mm do centro de rotação da junta 23. Quando o elo 4 está completamente distendido do elo 3, o centro da garra está a 270mm do centro de rotação da junta 23. Assim o conjunto de alcance radial do mecanismo é $[180, 270]mm$.

Quando a parte inferior do elo 2 toca a base do elo 1, o objeto está a 27mm do chão. Quando a parte superior do elo 2 toca o limite superior do elo 1, o objevo está a 187mm do chão. Assim, o alcance vertical do mecanismo é $[27, 187]mm$.

A junta 23 de rotação pode girar livremente, assim o alcance da rotação é $[-\pi, +\pi]$.

Juntando os três alcances anteriores obtemos a região de trabalho do mecanismo, que é $[180, 270] \times [27, 187] \times [-\pi, +\pi]mm$. Podemos ver que este conjunto abrange o conjunto radial necessário. O formato geométrico desse volume é de um tubo grosso com raio interno igual a 180mm, raio externo igual a 270mm e com extrusão de 27mm a 187mm.

5.2.6 Resumo de valores relevantes para os modelos dinâmicos e análises

- Massa do elo 1 mais o motor de passo da junta 12: $m_1 = 3.02445 + 0.98804 = 4.01249kg$
- Massa do elo 2 mais o motor de passo da junta 23: $m_2 = 2.78289 + 0.14835 = 2.93124kg$
- Massa do elo 3 mais o motor de passo da junta 34: $m_3 = 0.62267 + 0.06642 = 0.68909kg$
- Massa do elo 4: $m_4 = 0.21425kg$

- Massa total do mecanismo: $m_T = 7.84707kg$
- Momento de inércia do elo 3 mais o motor de passo da junta 34 em relação ao centro da junta 23: $L_{zz_3} = 3972.41766 + 402.323635 = 4374.741295kg \cdot mm^2$
- Momento de inércia do elo 4 em relação ao seu centro de massa: $I_{yy_4} = 369.70861kg \cdot mm^2$
- Distância entre o centro de massa do elo 4 e o objeto carregado: $d_{G_4} = 120 - 70.2348 = 49.7652 [mm]$

5.2.7 Valores assumidos

- Aceleração da gravidade: $g = 9806.65mm/s^2$
- Massa do objeto a ser carregado: $m_{obj} = 1kg$

5.3 Método Iterativo de Newton-Euler

O Método Iterativo de Newton-Euler é dividido em duas partes: cinemática direta e dinâmica inversa. A cinemática direta parte da base do robô, em que está o referencial inercial, para calcular as velocidades e acelerações dos elos consecutivos até chegar no objeto a ser carregado. Enquanto que a dinâmica inversa calcula as forças e torques necessários para atender a cinemática a partir do último elo e se propaga por elos consecutivos até a base.

5.3.1 Modelo teórico de corpos rígidos

Para diminuir o tamanho das equações, além de explicitar a atuação dos motores, são definidos referenciais nas juntas. Para a junta de rotação $[i]$, o referencial i se encontra no centro da junta do elo $(i - 1)$ com o componente z orientado na direção de rotação da junta com sentido anti-horário, além disso a componente y desse referencial aponta na direção e sentido do centro de massa do elo (i) , ou seja, ela mantém o referencial solidário a este elo, e o centro dele está alinhado com este centro de massa na componente x . Para a junta prismática $[i]$, o referencial i se encontra no centro de massa do elo (i) com a componente z orientada na direção da translação com sentido positivo ao afastar o elo da base, além disso as suas componentes x e y são paralelas as componentes x e y da base.

De acordo com a referência [1], as equações abaixo descrevem o processo iterativo da cinemática direta para as juntas de rotação (R) e prismática (P).

Velocidade angular do elo i em relação ao referencial i :

$$\omega_i^{\{i\}} = \begin{cases} (R_{\{i\}}^{\{i-1\}})^T \cdot \omega_{i-1}^{\{i-1\}} + e_z \cdot \dot{q}_i, & \text{se } R \\ (R_{\{i\}}^{\{i-1\}})^T \cdot \omega_{i-1}^{\{i-1\}}, & \text{se } P \end{cases} \quad (18)$$

Aceleração angular do elo i em relação ao referencial i :

$$\dot{\omega}_i^{\{i\}} = \begin{cases} (R_{\{i\}}^{\{i-1\}})^T \cdot \dot{\omega}_{i-1}^{\{i-1\}} + e_z \cdot \ddot{q}_i + ((R_{\{i\}}^{\{i-1\}})^T \cdot \omega_{i-1}^{\{i-1\}}) \times (e_z \cdot \dot{q}_i), & \text{se } R \\ (R_{\{i\}}^{\{i-1\}})^T \cdot \dot{\omega}_{i-1}^{\{i-1\}}, & \text{se } P \end{cases} \quad (19)$$

Aceleração linear do elo i em relação ao referencial i :

$$\ddot{p}_i^{\{i\}} = \begin{cases} (R_{\{i\}}^{\{i-1\}})^T \cdot (\ddot{p}_i^{\{i-1\}} + \dot{\omega}_{i-1}^{\{i-1\}} \times \hat{p}_{\{i\}}^{\{i-1\}} + \omega_{i-1}^{\{i-1\}} \times (\omega_{i-1}^{\{i-1\}} \times \hat{p}_{\{i\}}^{\{i-1\}})), & \text{se } R \\ (R_{\{i\}}^{\{i-1\}})^T (\ddot{p}_i^{\{i-1\}} + \dot{\omega}_{i-1}^{\{i-1\}} \times \hat{p}_{\{i\}}^{\{i-1\}} + \omega_{i-1}^{\{i-1\}} \times (\omega_{i-1}^{\{i-1\}} \times \hat{p}_{\{i\}}^{\{i-1\}})) \\ + 2 \cdot ((R_{\{i\}}^{\{i-1\}})^T \cdot \omega_{i-1}^{\{i-1\}}) \times (e_z \cdot \dot{q}_i) + e_z \cdot \ddot{q}_i, & \text{se } P \end{cases} \quad (20)$$

Aceleração do centro de massa do elo i em relação ao referencial i :

$$\ddot{s}_i^{\{i\}} = \ddot{p}_i^{\{i\}} + \dot{\omega}_i^{\{i\}} \times \hat{s}_i^{\{i\}} + \omega_i^{\{i\}} \times (\omega_i^{\{i\}} \times \hat{s}_i^{\{i\}}) \quad (21)$$

Onde $R_{\{i\}}^{\{i-1\}}$ é a matriz de rotação do referencial $\{i\}$ para $\{i-1\}$, sendo que a transposta desta matriz faz a operação inversa, e_z é o versor da componente z , $\hat{p}_{\{i\}}^{\{i-1\}}$ é o vetor da posição do referencial $\{i\}$ em relação a $\{i-1\}$, q_i é a coordenada generalizada do elo i e $\hat{s}_i^{\{i\}}$ é o vetor da posição do centro de massa do elo i em relação ao referencial $\{i\}$.

De acordo com a mesma referência [1], as equações abaixo descrevem o processo iterativo da dinâmica inversa.

Força resultante do elo i em relação ao referencial $\{i\}$:

$$\hat{f}_i^{\{i\}} = m_i \cdot \ddot{s}_i^{\{i\}} \quad (22)$$

Torque resultante do elo i em relação ao referencial $\{i\}$:

$$\hat{n}_i^{\{i\}} = I_i^{\{i\}} \cdot \dot{\omega}_i^{\{i\}} + \omega_i^{\{i\}} \times (I_i^{\{i\}} \cdot \omega_i^{\{i\}}) \quad (23)$$

Força entre os elos i e $(i-1)$ em relação ao referencial $\{i\}$:

$$f_i^{\{i\}} = R_{i+1}^{\{i\}} \cdot f_{i+1}^{\{i+1\}} + \hat{f}_i^{\{i\}} \quad (24)$$

Torque entre os elos i e $(i-1)$ em relação ao referencial $\{i\}$:

$$n_i^{\{i\}} = R_{i+1}^{\{i\}} \cdot n_{i+1}^{\{i+1\}} + \hat{n}_i^{\{i\}} + \hat{s}_i^{\{i\}} \times \hat{f}_i^{\{i\}} + \hat{p}_{i+1}^{\{i\}} \times (R_{i+1}^{\{i\}} \cdot f_{i+1}^{\{i+1\}}) \quad (25)$$

5.3.2 Equações da dinâmica do robô PRP

As equações da dinâmica do robô PRP através do Método de Iterativo de Newton-Euler foram derivadas através de um programa de matemática simbólica feito no Matlab, que se encontra na seção 7.5. E o resultado obtivo foi:

```
>> f4_4(3)

ans =

(m4*(DG4 - r) - m5*r)*dt^2 + ddr*(m4 + m5)

>> n3_3(3)

ans =

(m3*DG3^2 + Iyy4_4 + Izz3_3 + r*(m5*r - m4*(DG4 - r)) + DG4*m4*(DG4 - r))*ddt + (r*(2*m4 + 2*m5) - 2*DG4*m4)*dt*dr

>> f2_2(3)

ans =

(m2 + m3 + m4 + m5)*ddh + g*m2 + g*m3 + g*m4 + g*m5
```

Figura 23: Equações dos esforços dos 3 motores de passo

Organizando estas equações, trocando m_5 por m_{obj} e juntando termos de momento de inércia do elo 3 através do Teorema de Eixos Paralelos chegamos em:

$$F_{M1} = (m_2 + m_3 + m_4 + m_{obj}) \cdot (\ddot{h} + g) \quad (26)$$

$$T_{M2} = (L_{zz3}^3 + I_{yy4}^4 + m_4 \cdot (r - d_{G4})^2 + m_{obj} \cdot r^2) \cdot \ddot{\theta} + 2 \cdot (m_4 \cdot (r - d_{G4}) + m_{obj} \cdot r) \cdot \dot{\theta} \cdot \dot{r} \quad (27)$$

$$F_{M3} = (m_4 + m_{obj}) \cdot \ddot{r} - (m_4 \cdot (r - d_{G4}) + m_{obj} \cdot r) \cdot \dot{\theta}^2 \quad (28)$$

5.4 Método de Gibbs-Appell

Além do Método Iterativo de Newton-Euler, o modelo dinâmico do mecanismo também foi obtido através do Método de Gibbs-Appell. Assim como o método anterior, este método começa com a cinemática direta, entretanto todas as velocidades e acelerações são observadas em relação referencial inercial, ou seja, a base do robô no elo 1. Em seguida, a construção das matrizes $[C]$ e $[D]$, e do vetor f definidos abaixo determina os esforços dos motores, que devem coincidir com os resultados do método anterior. Os torques e forças externos também são observados em relação ao referencial inercial. Os referências nas juntas são definidos da mesma maneira como na seção anterior.

5.4.1 Modelo teórico de corpos rígidos

Pela referência [2], o modelo dinâmico do Método de Gibbs-Appel é descrito pela equação matricial abaixo:

$$[C]^T \cdot [D] \cdot f = 0 \quad (29)$$

A matriz $[C]$ é definida, simplifcadamente, como:

$$[C] = \begin{bmatrix} 1_{m \times m} \\ J_{n-m \times m} \end{bmatrix} \quad (30)$$

No mecanismo PRP proposto, a matriz $[C]$ será igual a matriz identidade 3×3 , pois o grau de mobilidade do mecanismo é igual ao número de coordenadas generalizadas.

Já a matriz $[D]$ e o vetor f são, genericamente, definidos por:

$$[D] = \begin{bmatrix} \frac{\partial v_{G1}^T}{\partial u_1} & \frac{\partial \omega_{G1}^T}{\partial u_1} & \frac{\partial v_{G2}^T}{\partial u_1} & \frac{\partial \omega_{G2}^T}{\partial u_1} & \cdots & \frac{\partial v_{GN}^T}{\partial u_1} & \frac{\partial \omega_{GN}^T}{\partial u_1} \\ \frac{\partial v_{G1}^T}{\partial u_2} & \frac{\partial \omega_{G1}^T}{\partial u_2} & \frac{\partial v_{G2}^T}{\partial u_2} & \frac{\partial \omega_{G2}^T}{\partial u_2} & \cdots & \frac{\partial v_{GN}^T}{\partial u_2} & \frac{\partial \omega_{GN}^T}{\partial u_2} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{\partial v_{G1}^T}{\partial u_N} & \frac{\partial \omega_{G1}^T}{\partial u_N} & \frac{\partial v_{G2}^T}{\partial u_N} & \frac{\partial \omega_{G2}^T}{\partial u_N} & \cdots & \frac{\partial v_{GN}^T}{\partial u_N} & \frac{\partial \omega_{GN}^T}{\partial u_N} \end{bmatrix} \quad (31)$$

$$f = \begin{bmatrix} \sum F_{at,1} - m_1 \cdot a_{G_1} \\ \sum M_{at,1} - ([I_{1,G_1}] \cdot \dot{\omega}_1 + \omega_1 \times ([I_{1,G_1}] \cdot \omega_1)) \\ \sum F_{at,2} - m_2 \cdot a_{G_2} \\ \sum M_{at,2} - ([I_{2,G_2}] \cdot \dot{\omega}_2 + \omega_2 \times ([I_{2,G_2}] \cdot \omega_2)) \\ \vdots \\ \sum F_{at,N} - m_N \cdot a_{G_N} \\ \sum M_{at,N} - ([I_{N,G_N}] \cdot \dot{\omega}_N + \omega_N \times ([I_{N,G_N}] \cdot \omega_N)) \end{bmatrix} \quad (32)$$

5.4.2 Velocidades e acelerações lineares

A partir das equações algébricas das posições x dos centros de massa dos elos em função das coordenadas generalizadas q em relação ao referencial inercial:

$$x = f(q) \quad (33)$$

É possível encontrar as velocidades lineares dos centros de massa dos elos do mecanismo derivando no tempo as equações das posições e usando a matriz Jacobiana:

$$\dot{x} = [J] \cdot \dot{q} \quad (34)$$

Assim como é possível encontrar as acelerações lineares derivando no tempo a equação das velocidades lineares e usando a matriz Jacobiana e a sua derivada temporal:

$$\ddot{x} = [\dot{J}] \cdot \dot{q} + [J] \cdot \ddot{q} \quad (35)$$

Para o mecanismo PRP as equações algébricas das posições dos 4 elos e do objeto carregado são:

$$r_{G_2}^{\{1\}} = (0, 0, h)^T \quad (36)$$

$$r_{G_3}^{\{1\}} = (-d_{G_3} \cdot \sin(\theta), d_{G_3} \cdot \cos(\theta), h)^T \quad (37)$$

$$r_{G_4}^{\{1\}} = (-(r - d_{G_4}) \cdot \sin(\theta), (r - d_{G_4}) \cdot \cos(\theta), h)^T \quad (38)$$

$$r_{G_{obj}}^{\{1\}} = (-r \cdot \sin(\theta), r \cdot \cos(\theta), h)^T \quad (39)$$

5.4.3 Aceleração angular

De acordo com a referência [3], a velocidade angular com que o elo i gira em relação ao referencial inercial $\{1\}$ é dada por:

$$\Omega_i^{\{1\}} = \Omega_{i-1}^{\{1\}} + R_{\{i-1\}}^{\{1\}} \cdot \Omega_i^{\{i-1\}} \quad (40)$$

Derivando a equação acima, tem-se a aceleração angular:

$$\dot{\Omega}_i^{\{1\}} = \dot{\Omega}_{i-1}^{\{1\}} + R_{\{i-1\}}^{\{1\}} \cdot \dot{\Omega}_i^{\{i-1\}} + \Omega_{i-1}^{\{1\}} \times (R_{\{i-1\}}^{\{1\}} \cdot \Omega_i^{\{i-1\}}) \quad (41)$$

5.4.4 Equações da dinâmica do robô PRP

As equações da dinâmica do robô PRP através do Método de Gibbs-Appell foram deduzidas através de um programa de matemática simbólica feito no Matlab, que se encontra na seção 7.6. E o resultado obtivo foi:

```

Fm1 == (m2 + m3 + m4 + m5)*ddh + g*(m2 + m3 + m4 + m5)
(2*DG4*m4*r)*ddth + (2*DG4*m4)*dth*dr + Tm2 == (Izz3_3 + Izz4_4 + DG3^2*m3 + DG4^2*m4 + m4*r^2 + m5*r^2)*ddth + (2*m4*r + 2*m5*r)*dth*dr
(m4*r + m5*r)*dth^2 + Fm3 == DG4*m4*dth^2 + ddr*(m4 + m5)

```

Figura 24: Equações dos esforços dos 3 motores de passo

Organizando estas equações, trocando m_5 por m_{obj} e juntando termos de momento de inércia do elo 3 através do Teorema de Eixos Paralelos chegamos em:

$$F_{M1} = (m_2 + m_3 + m_4 + m_{obj}) \cdot (\ddot{h} + g) \quad (42)$$

$$T_{M2} = (L_{zz3}^3 + I_{yy4}^4 + m_4 \cdot (r - d_{G4})^2 + m_{obj} \cdot r^2) \cdot \ddot{\theta} + 2 \cdot (m_4 \cdot (r - d_{G4}) + m_{obj} \cdot r) \cdot \dot{\theta} \cdot \dot{r} \quad (43)$$

$$F_{M3} = (m_4 + m_{obj}) \cdot \ddot{r} - (m_4 \cdot (r - d_{G4}) + m_{obj} \cdot r) \cdot \dot{\theta}^2 \quad (44)$$

5.5 Resultados de simulação

A partir dos resultados da cinemática inversa, é possível calcular os esforços dos motores pela dinâmica inversa, cujas equações foram descritas pelos métodos de Newton-Euler e Gibbs-Appell. Como esperado, para o robô PRP, ambos os métodos resultam nas mesmas saídas abaixo, cujo código se encontra na seção 7.7.

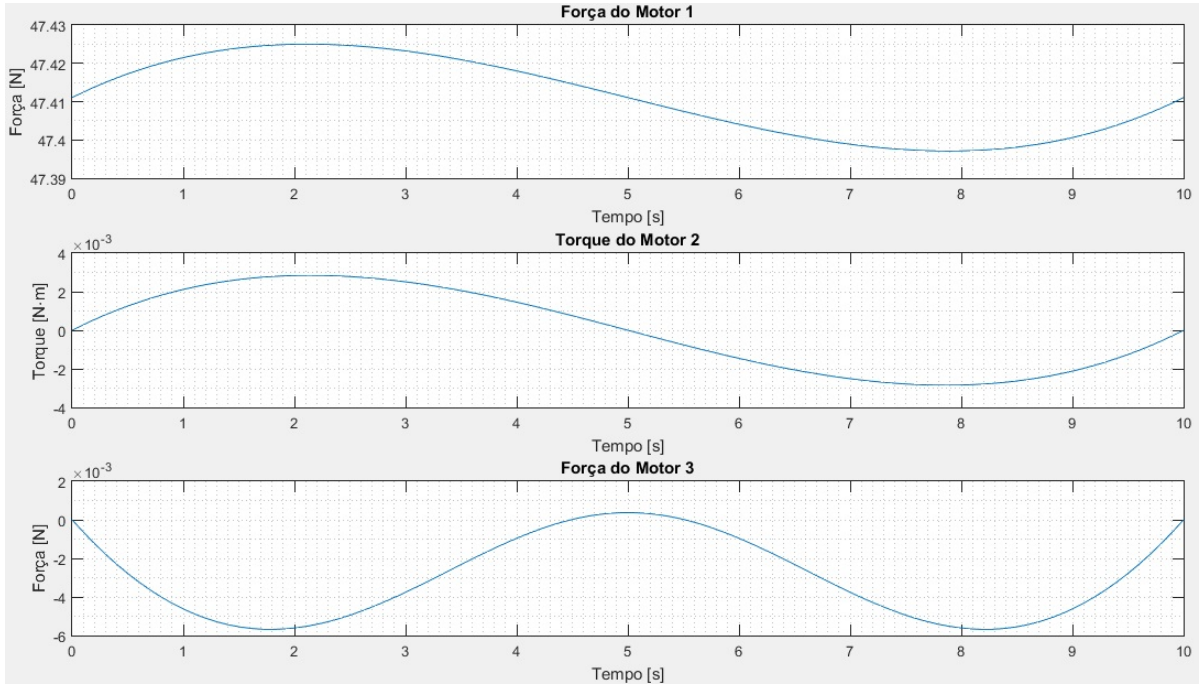


Figura 25: Esforços dos motores

6 Análise dos resultados

A partir da figura 23, que mostra os esforços dos motores, é possível ver que a força do motor de passo da junta 12 é praticamente constante e ele precisa sustentar uma carga de quase 5kg, sendo que 1kg é do objeto carregado. O torque do motor da junta 23 é baixo e maior parte de sua magnitude se deve ao momento de inércia do objeto carregado, pois os elos 3 e 4 têm baixo momento de inércia em relação com o objeto carregado. A força do motor da junta 34 é bem baixo, de um equivalente a uma carga de algumas gramas, pois há força externa aplicada na sua direção, apenas os efeitos das inércias do elo 4 e do objeto carregado.

O mecanismo todo pesa $7.84707kg$. Desta maneira as forças e torques em questão são pequenas quando comparadas com o peso do robô, embora a força do primeiro motor corresponda a mais da metade do peso do mecanismo, ela é praticamente constante. Isto pode indicar que a estrutura do robô e o tamanho dos motores de passo, o que influencia na potência deles, estão hiperdimensionados para o problema. Por outro lado, o mecanismo consegue realizar o acompanhamento retilíneo da trajetória sem problema.

7 Programas feitos em *Matlab*

7.1 Simulação da trajetória

```
clear, close, clc;
%Definicao dos Parametros
T = 10; %tempo final [s]
x1 = 200; %[mm]
y1 = -100; %[mm]
z1 = 0; %[mm]
x2 = 200; %[mm]
y2 = 100; %[mm]
z2 = 50; %[mm]
L = sqrt(((x2-x1)^2)+((y2-y1)^2)+((z2-z1)^2));

%Equacoes de movimento
t = 0:0.1:T;
y = L*(6*((t/T).^5)-15*((t/T).^4)+10*((t/T).^3));
dy = (L/T)*(30*((t/T).^4)-60*((t/T).^3)+30*((t/T).^2));
ddy = (L/T.^2)*(120*((t/T).^3)-180*((t/T).^2)+60*((t/T)));

%Plotando Grafico
plot(t,y,'b',t,dy,'r',t,ddy,'k');
title('Trajetória do objeto carregado');
xlabel('Tempo [s]');
ylabel('Intensidade das grandezas');
legend('Posição [mm]', 'Velocidade [mm/s]', 'Aceleração [mm/s^2]', 'Location', '
    ↪ northwest');
grid minor;
```

7.2 Cinemática inversa

```
close, clear, clc;
Pi = [200; -100; 0]; % Posição inicial da trajetória [mm]
Pf = [200; 100; 50]; % Posição final da trajetória [mm]
L = norm(Pf - Pi); % Comprimento da trajetória [mm]
v = (Pf - Pi) / L; % Vetor unitário da mesma orientação e sentido da trajetória
    ↪ ria
T = 10; % Tempo total [s]
n = 100; % Número de passos
t = linspace(0, T, n); % Vetor do tempo
x0 = [224; -0.5; 0]; % Vetor da posição inicial

% Polinmio interpolador de 5 grau e suas derivadas
y = L * (6*(t/T).^5 - 15*(t/T).^4 + 10*(t/T).^3);
dy = L / T * (30*(t/T).^4 - 60*(t/T).^3 + 30*(t/T).^2);
```

```

ddy = L / T^2 * (120*(t/T).^3 - 180*(t/T).^2 + 60*t/T);

r_P = zeros(3,n);
Mat_r = zeros(3,n);
Mat_v = zeros(3,n);
Mat_a = zeros(3,n);

for i = 1:n
    % Posições
    r_P(:,i) = Pi + y(i) * v;
    P = @(x) [x(1) * cos(x(2)) - r_P(1,i);
              x(1) * sin(x(2)) - r_P(2,i);
              x(3) - r_P(3,i)];
    Mat_r(:,i) = fsolve(P, x0); clc;
    r = Mat_r(1,:);
    theta = Mat_r(2,:);
    h = Mat_r(3,:);
    x0 = [r(i); theta(i); h(i)];

    % Velocidades
    v_P = dy(i) * v;
    Mat_A = [cos(theta(i)), -r(i)*sin(theta(i)), 0;
             sin(theta(i)), r(i)*cos(theta(i)), 0;
             0, 0, 1];
    Mat_v(:,i) = Mat_A \ v_P;
    dr = Mat_v(1,:);
    dtheta = Mat_v(2,:);
    dh = Mat_v(3,:);

    % Acelerações
    a_P = ddy(i) * v;
    Mat_B = [2*dr(i)*sin(theta(i))*dtheta(i) + r(i)*cos(theta(i))*dtheta(i)^2 +
    ↪ a_P(1);
             -2*dr(i)*cos(theta(i))*dtheta(i) + r(i)*sin(theta(i))*dtheta(i)^2 +
    ↪ a_P(2);
    ↪ a_P(3)
    ↪ ];

    Mat_a(:,i) = Mat_A \ Mat_B;
    ddr = Mat_a(1,:);
    ddtheta = Mat_a(2,:);
    ddh = Mat_a(3,:);
end

```

7.3 Plotagem dos gráficos da cinemática inversa

```

close, clc;
figure('Name','Coordenada generalizada r','NumberTitle','off')

```

```

subplot(3,1,1), plot(t,r,'linewidth',1.5), grid minor
title('\textbf{Coordenada generalizada $r$ e suas derivadas}','Interpreter', '
    ↪ latex')
xlabel('$Tempo [s]$', 'Interpreter', 'latex')
ylabel('$r [mm]$', 'Interpreter', 'latex')
set(gca,'FontSize',12,'FontName','Times New Roman')
subplot(3,1,2), plot(t,dr,'linewidth',1.5), grid minor
xlabel('$Tempo [s]$', 'Interpreter', 'latex')
ylabel('$\dot{r} [mm/s]$', 'Interpreter', 'latex')
set(gca,'FontSize',12,'FontName','Times New Roman')
subplot(3,1,3), plot(t,ddr,'linewidth',1.5), grid minor
xlabel('$Tempo [s]$', 'Interpreter', 'latex')
ylabel('$\ddot{r} [mm/s^2]$', 'Interpreter', 'latex')
set(gca,'FontSize',12,'FontName','Times New Roman')

figure('Name','Coordenada generalizada theta','NumberTitle','off')
subplot(3,1,1), plot(t,theta*180/pi,'linewidth',1.5), grid minor
title('\textbf{Coordenada generalizada $\theta$ e suas derivadas}','
    ↪ Interpreter', 'latex')
xlabel('$Tempo [s]$', 'Interpreter', 'latex')
ylabel('$\theta [deg]$', 'Interpreter', 'latex')
set(gca,'FontSize',12,'FontName','Times New Roman')
subplot(3,1,2), plot(t,dtheta*180/pi,'linewidth',1.5), grid minor
xlabel('$Tempo [s]$', 'Interpreter', 'latex')
ylabel('$\dot{\theta} [deg/s]$', 'Interpreter', 'latex')
set(gca,'FontSize',12,'FontName','Times New Roman')
subplot(3,1,3), plot(t,ddtheta*180/pi,'linewidth',1.5), grid minor
xlabel('$Tempo [s]$', 'Interpreter', 'latex')
ylabel('$\ddot{\theta} [deg/s^2]$', 'Interpreter', 'latex')
set(gca,'FontSize',12,'FontName','Times New Roman')

figure('Name','Coordenada generalizada h','NumberTitle','off')
subplot(3,1,1), plot(t,h,'linewidth',1.5), grid minor
title('\textbf{Coordenada generalizada $h$ e suas derivadas}','Interpreter', '
    ↪ latex')
xlabel('$Tempo [s]$', 'Interpreter', 'latex')
ylabel('$h [mm]$', 'Interpreter', 'latex')
set(gca,'FontSize',12,'FontName','Times New Roman')
subplot(3,1,2), plot(t,dh,'linewidth',1.5), grid minor
xlabel('$Tempo [s]$', 'Interpreter', 'latex')
ylabel('$\dot{h} [mm/s]$', 'Interpreter', 'latex')
set(gca,'FontSize',12,'FontName','Times New Roman')
subplot(3,1,3), plot(t,ddh,'linewidth',1.5), grid minor
xlabel('$Tempo [s]$', 'Interpreter', 'latex')
ylabel('$\ddot{h} [mm/s^2]$', 'Interpreter', 'latex')
set(gca,'FontSize',12,'FontName','Times New Roman')

```

7.4 Exportação dos dados da cinemática inversa para o *SolidWorks*

```
close, clc;
fID = fopen('h.txt','w'); fprintf(fID,'%f,%f\n',[t;h]); fclose(fID);
fID = fopen('dh.txt','w'); fprintf(fID,'%f,%f\n',[t;dh]); fclose(fID);
fID = fopen('ddh.txt','w'); fprintf(fID,'%f,%f\n',[t;ddh]); fclose(fID);

fID = fopen('theta.txt','w'); fprintf(fID,'%f,%f\n',[t;theta*180/pi]); fclose(
    ↪ fID);
fID = fopen('dtheta.txt','w'); fprintf(fID,'%f,%f\n',[t;dtheta*180/pi]);
    ↪ fclose(fID);
fID = fopen('ddtheta.txt','w'); fprintf(fID,'%f,%f\n',[t;ddtheta*180/pi]);
    ↪ fclose(fID);

fID = fopen('r.txt','w'); fprintf(fID,'%f,%f\n',[t;r]); fclose(fID);
fID = fopen('dr.txt','w'); fprintf(fID,'%f,%f\n',[t;dr]); fclose(fID);
fID = fopen('ddr.txt','w'); fprintf(fID,'%f,%f\n',[t;ddr]); fclose(fID);
```

7.5 Equações da dinâmica por *Newton-Euler*

```
close, clear, clc;
%%% Direct Kinematics %%%
g = sym('g', 'real'); % Acceleration of gravity
h = sym('h', 'real'); % height
dh = sym('dh', 'real'); % d/dt (h)
ddh = sym('ddh', 'real'); % d^2/dt^2 (h)
t = sym('t', 'real'); % theta
dt = sym('dt', 'real'); % d/dt (theta)
ddt = sym('ddt', 'real'); % d^2/dt^2 (theta)
r = sym('r', 'real'); % radial distance
dr = sym('dr', 'real'); % d/dt (r)
ddr = sym('ddr', 'real'); % d^2/dt^2 (r)
DG1 = sym('DG1', 'real'); % Distance of the center of mass of LINK 1 to FRAME
    ↪ 1
DG3 = sym('DG3', 'real'); % Distance of the center of mass of LINK 3 to FRAME
    ↪ 3
DG4 = sym('DG4', 'real'); % Distance of the center of mass of LINK 4 to FRAME
    ↪ 4

% Angular velocity of FRAME i relative to FRAME i-1
w_r = @(R,w,dq) transpose(R) * w + [0;0;dq]; % Rotation
w_p = @(R,w) transpose(R) * w; % Prismatic

% Angular acceleration of FRAME i relative to FRAME i-1
dw_r = @(R,dw,w,ddq,dq) transpose(R)*dw + [0;0;ddq] + cross(transpose(R)*w
    ↪ ,[0;0;dq]);
dw_p = @(R,dw) transpose(R) * dw;
```



```

% Linear acceleration of FRAME i relative to FRAME i-1
ddp_r = @(R,dw,w,ddp,ph) transpose(R) * (ddp + cross(dw,ph) + cross(w,cross(w,
    ↪ ph)));
ddp_p = @(R,dw,w,ddp,ph,ddq,dq) transpose(R) * (ddp + cross(dw,ph) ...
    + cross(w,cross(w,ph))) + 2*cross(transpose(R)*w,[0;0;dq]) + [0;0;ddq];

% Acceleration of the center of mass of LINK i relative to FRAME i
dds = @(ddp,dw,w,sh) ddp + cross(dw,sh) + cross(w,cross(w,sh));

%%% FRAME 1 -> Base
w1_1 = [0;0;0];
dw1_1 = [0;0;0];
ddp1_1 = [0;0;g]; % gravity effect
sh1_1 = [0;0;DG1]; % Position of G1 relative to FRAME 1
dds1_1 = dds (ddp1_1, dw1_1, w1_1, sh1_1); % Acceleration of G1

%%% FRAME 2 -> Joint 12 is prismatic
R21 = eye(3); % Rotation matrix between FRAME i and FRAME i-1
ph21 = [0;0;h]; % Position of FRAME i relative to FRAME i-1
w2_2 = w_p (R21, w1_1);
dw2_2 = dw_p (R21, dw1_1);
ddp2_2 = ddp_p (R21, dw1_1, w1_1, ddp1_1, ph21, ddh, dh);
sh2_2 = [0;0;0];
dds2_2 = dds (ddp2_2, dw2_2, w2_2, sh2_2);

%%% FRAME 3 -> Joint 23 is rotational
R32 = [cos(t), -sin(t), 0;
        sin(t), cos(t), 0;
        0, 0, 1];
ph32 = [0;0;0];
w3_3 = w_r (R32, w2_2, dt);
dw3_3 = dw_r (R32, dw2_2, w2_2, ddt, dt);
ddp3_3 = ddp_r (R32, dw2_2, w2_2, ddp2_2, ph32);
sh3_3 = [0;DG3;0];
dds3_3 = dds (ddp3_3, dw3_3, w3_3, sh3_3);

%%% FRAME 4 -> Joint 34 is prismatic
R43 = [1, 0, 0;
        0, 0, 1;
        0, -1, 0];
ph43 = [0;r;0];
w4_4 = w_p (R43, w3_3);
dw4_4 = dw_p (R43, dw3_3);
ddp4_4 = ddp_p (R43, dw3_3, w3_3, ddp3_3, ph43, ddr, dr);
sh4_4 = [0;0;-DG4];
dds4_4 = dds (ddp4_4, dw4_4, w4_4, sh4_4);

%%% FRAME 5 -> Object
R54 = eye(3);

```

```

ph54 = [0;0;0];
w5_5 = w_p (R54, w4_4);
dw5_5 = dw_p (R54, dw4_4);
ddp5_5 = ddp_r (R54, dw4_4, w4_4, ddp4_4, ph54);
sh5_5 = [0;0;0];
dds5_5 = dds (ddp5_5, dw5_5, w5_5, sh5_5);

%%% Inverse Dynamics %%%
m1 = sym('m1', 'real'); % Mass of LINK i
m2 = sym('m2', 'real');
m3 = sym('m3', 'real');
m4 = sym('m4', 'real');
m5 = sym('m5', 'real');
Ixx4_4 = sym('Ixx4_4', 'real'); % Inertia matrix parameters
Iyy4_4 = sym('Iyy4_4', 'real');
Izz4_4 = sym('Izz4_4', 'real');
Ixy4_4 = sym('Ixy4_4', 'real');
Ixx3_3 = sym('Ixx3_3', 'real');
Iyy3_3 = sym('Iyy3_3', 'real');
Izz3_3 = sym('Izz3_3', 'real');
Ixz3_3 = sym('Ixz3_3', 'real');
Ixx2_2 = sym('Ixx2_2', 'real');
Iyy2_2 = sym('Iyy2_2', 'real');
Izz2_2 = sym('Izz2_2', 'real');
Ixy2_2 = sym('Ixy2_2', 'real');
Ixx1_1 = sym('Ixx1_1', 'real');
Iyy1_1 = sym('Iyy1_1', 'real');
Izz1_1 = sym('Izz1_1', 'real');
Ixy1_1 = sym('Ixy1_1', 'real');

% Force and Torque
f = @(R,fa,ddsi,m) R * fa + m * ddsi;
n = @(R,na,I,dw,w,m,sh,ddsi,ph,fa) R*na + I*dw + cross(w,I*w) ...
    + m*cross(sh,ddsi) + cross(ph,R*fa);

%%% FRAME 5 -> Object -> Material Point
I5_5 = zeros(3); % Inertia Matrix
f5_5 = f (eye(3), [0;0;0], dds5_5, m5);
n5_5 = n (eye(3), [0;0;0], I5_5, dw5_5, w5_5, m5, sh5_5, dds5_5, [0;0;0],
    ↪ [0;0;0]);

%%% FRAME 4
I4_4 = [Ixx4_4, Ixy4_4, 0;
        Ixy4_4, Iyy4_4, 0;
        0, 0, Izz4_4];
f4_4 = f (R54, f5_5, dds4_4, m4);

```

```

n4_4 = n (R54, n5_5, I4_4, dw4_4, w4_4, m4, sh4_4, dds4_4, ph54, f5_5);

%%% FRAME 3
I3_3 = [Ixx3_3, 0, Ixz3_3;
        0, Iyy3_3, 0;
        Ixz3_3, 0, Izz3_3];
f3_3 = f (R43, f4_4, dds3_3, m3);
n3_3 = n (R43, n4_4, I3_3, dw3_3, w3_3, m3, sh3_3, dds3_3, ph43, f4_4);

%%% FRAME 2
I2_2 = [Ixx2_2, Ixy2_2, 0;
        Ixy2_2, Iyy2_2, 0;
        0, 0, Izz2_2];
f2_2 = f (R32, f3_3, dds2_2, m2);
n2_2 = n (R32, n3_3, I2_2, dw2_2, w2_2, m2, sh2_2, dds2_2, ph32, f3_3);

%%% FRAME 1 -> Base
I1_1 = [Ixx1_1, Ixy1_1, 0;
        Ixy1_1, Iyy1_1, 0;
        0, 0, Izz1_1];
f1_1 = f (R21, f2_2, dds1_1, m1);
n1_1 = n (R21, n2_2, I1_1, dw1_1, w1_1, m1, sh1_1, dds1_1, ph21, f2_2);

% Organizing
co = [ddh,dh,ddt,dt,ddr,dr];
f1_1 = collect(f1_1, co); n1_1 = collect(n1_1, co);
f2_2 = collect(f2_2, co); n2_2 = collect(n2_2, co);
f3_3 = collect(f3_3, co); n3_3 = collect(n3_3, co);
f4_4 = collect(f4_4, co); n4_4 = collect(n4_4, co);
f5_5 = collect(f5_5, co); n5_5 = collect(n5_5, co);

```

7.6 Equações da dinâmica por *Gibbs-Appell*

```

close, clear, clc;
%%% Direct Kinematics %%%
g = sym('g', 'real'); % Acceleration of gravity
h = sym('h', 'real'); % height
dh = sym('dh', 'real'); % d/dt (h)
ddh = sym('ddh', 'real'); % d^2/dt^2 (h)
th = sym('th', 'real'); % theta
dth = sym('dth', 'real'); % d/dt (theta)
ddth = sym('ddth', 'real'); % d^2/dt^2 (theta)
r = sym('r', 'real'); % radial distance
dr = sym('dr', 'real'); % d/dt (r)
ddr = sym('ddr', 'real'); % d^2/dt^2 (r)
DG3 = sym('DG3', 'real'); % Distance of the center of mass of LINK 3 to FRAME
    ↪ 2

```

```

DG4 = sym('DG4', 'real'); % Distance of the center of mass of LINK 4 to FRAME
    ↪ 5
q = [h; th; r]; % Generalized coordinates
dq = [dh; dth; dr]; % Generalized velocities
ddq = [ddh; dddh; ddr]; % Generalized acceleration

%% Position vector of the centers of mass relative to FRAME 1
rG2_1 = [0; 0; h];
rG3_1 = [-DG3*sin(th); DG3*cos(th); h];
rG4_1 = [-(r-DG4)*sin(th); (r-DG4)*cos(th); h];
rG5_1 = [-r*sin(th); r*cos(th); h];
rG_1 = [rG2_1; rG3_1; rG4_1; rG5_1];

%% Linear velocity vector of the centers of mass relative to FRAME 1
Jv = jacobian(rG_1, q);
vG_1 = Jv * dq;
vG2_1 = vG_1(1:3,1);
vG3_1 = vG_1(4:6,1);
vG4_1 = vG_1(7:9,1);
vG5_1 = vG_1(10:12,1);

%% Linear acceleration vector of the centers of mass relative to FRAME 1
a = sym('h(t)', 'real');
b = sym('th(t)', 'real');
c = sym('r(t)', 'real');
Ja = diff(subs(Jv, [h,th,r], [a,b,c]));
Ja = subs(Ja, [a,b,c,diff(a),diff(b),diff(c)], [h,th,r,dh,dth,dr]);
aG_1 = Ja * dq + Jv * ddq;
aG2_1 = aG_1(1:3,1);
aG3_1 = aG_1(4:6,1);
aG4_1 = aG_1(7:9,1);
aG5_1 = aG_1(10:12,1);

%% Angular velocity vector of the centers of mass relative to FRAME 1
w1_1 = [0;0;0];
w2_2 = [0;0;0];
w3_3 = [0;0;dth];
w4_4 = [0;0;0];
R32 = [cos(th),-sin(th),0; sin(th),cos(th),0; 0,0,1];
R43 = [1,0,0; 0,0,1; 0,-1,0];
R21 = eye(3);
R31 = R21 * R32;
R41 = R31 * R43;
w2_1 = w1_1 + R21 * w2_2;
w3_1 = w2_1 + R31 * w3_3;
w4_1 = w3_1 + R41 * w4_4;

%% Angular acceleration vector of the centers of mass relative to FRAME 1
dw1_1 = [0;0;0];

```

```

dw2_2 = [0;0;0];
dw3_3 = [0;0;ddth];
dw4_4 = [0;0;0];
dw2_1 = dw1_1 + R21 * dw2_2 + cross(w1_1, R21 * w2_2);
dw3_1 = dw2_2 + R31 * dw3_3 + cross(w2_2, R31 * w3_3);
dw4_1 = dw3_3 + R41 * dw4_4 + cross(w3_3, R41 * w4_4);

%%% Inverse Dynamics %%%
m2 = sym('m2', 'real'); % Mass of LINK i
m3 = sym('m3', 'real');
m4 = sym('m4', 'real');
m5 = sym('m5', 'real');
Ixx4_4 = sym('Ixx4_4', 'real'); % Inertia matrix parameters
Iyy4_4 = sym('Iyy4_4', 'real');
Izz4_4 = sym('Izz4_4', 'real');
Ixy4_4 = sym('Ixy4_4', 'real');
Ixx3_3 = sym('Ixx3_3', 'real');
Iyy3_3 = sym('Iyy3_3', 'real');
Izz3_3 = sym('Izz3_3', 'real');
Ixz3_3 = sym('Ixz3_3', 'real');
Ixx2_2 = sym('Ixx2_2', 'real');
Iyy2_2 = sym('Iyy2_2', 'real');
Izz2_2 = sym('Izz2_2', 'real');
Ixy2_2 = sym('Ixy2_2', 'real');
Fm1 = sym('Fm1', 'real'); % Motors forces and torques
Tm2 = sym('Tm2', 'real');
Fm3 = sym('Fm3', 'real');

%%% Matrix C
C = eye(3); % n = m = 3

%%% Matrix D
JV2_1 = jacobian([vG2_1', w2_1'], dq);
JV3_1 = jacobian([vG3_1', w3_1'], dq);
JV4_1 = jacobian([vG4_1', w4_1'], dq);
D = [JV2_1', JV3_1', JV4_1'];

% Inertia Matrices
I2_2 = [Ixx2_2, Ixy2_2, 0;
        Ixy2_2, Iyy2_2, 0;
        0, 0, Izz2_2];
I3_3 = [Ixx3_3, 0, Ixz3_3;
        0, Iyy3_3, 0;
        Ixz3_3, 0, Izz3_3];
I4_4 = [Ixx4_4, Ixy4_4, 0;
        Ixy4_4, Iyy4_4, 0;
        0, 0, Izz4_4];

```

```

%%% Vector F
F5 = m5 * aG5_1; % Object reaction force
M5 = cross(R41 * [0;0;DG4], F5); % Object reaction torque
F2 = [0;0;Fm1-m2*g] - m2*aG2_1;
M2 = [0;0;0] - I2_2*dw2_1 - cross(w2_1,I2_2*w2_1);
F3 = [0;0;-m3*g] - m3*aG3_1;
M3 = [0;0;Tm2] - I3_3*dw3_1 - cross(w3_1,I3_3*w3_1);
F4 = R41*[0;(m4+m5)*g;Fm3] - F5 - m4*aG4_1;
M4 = - M5 - I4_4*dw4_1 - cross(w4_1,I4_4*w4_1);
F = [F2; M2; F3; M3; F4; M4];

%%% Gibbs-Appel Equations
GAE = C' * D * F == zeros(3,1);
GAE = collect(GAE, [ddh,dh,ddth,dth,ddr,dr]); % Organize
GAE = simplify(GAE); % Cancel terms
GAE = collect(GAE, [ddh,dh,ddth,dth,ddr,dr]);

```

7.7 Dinâmica inversa

```

close, clc;
%%% Inverse Dynamics %%%
m2 = 2.78289 + 0.14835; % Atuador_P1_Base_R + Step_Motor_Medium [kg]
m3 = 0.62267 + 0.06642; % Atuador_R_Base_P2 + Step_Motor_Small [kg]
m4 = 0.2142522; % Atuador_P2 [kg]
m5 = 1; % Objeto carregado [kg]
Lzz3_3 = 3972.41766 + 402.323635; % Atuador_R_Base_P2 + Step_Motor_Small [kg*
    ↳ mm2]
Iyy4_4 = 369.7086102; % Atuador_P2 [kg*mm2]
g = 9806.65; % aceleração da gravidade [mm/s2]
DG4 = 120 - 70.2348; % Dist(G4, objeto) [mm]

Fm1 = (m2 + m3 + m4 + m5) * (ddh + g) * 1e-3;
Tm2 = ((Lzz3_3 + Iyy4_4 + m4*(r-DG4).^2 + m5*r.^2) .* ddtheta ...
    + 2 * (m4*(r-DG4) + m5*r) .* dtheta .* dr) * 1e-6;
Fm3 = ((m4 + m5) * ddr - (m4*(r-DG4) + m5*r) .* dtheta.^2) * 1e-3;

figure(1);
subplot(311), plot(t, Fm1), grid minor, title('Força do Motor 1');
xlabel('Tempo [s]'), ylabel('Força [N]');
subplot(312), plot(t, Tm2), grid minor, title('Torque do Motor 2');
xlabel('Tempo [s]'), ylabel('Torque [N\cdot m]');
subplot(313), plot(t, Fm3), grid minor, title('Força do Motor 3');
xlabel('Tempo [s]'), ylabel('Força [N]');

```

8 Distribuição das tarefas

Seção do trabalho	Alunos
Síntese do tipo e elaboração do modelo CAD preliminar	Caio Noboru Asai
	Gustavo Garcia Frozoni
Cinemática direta	Marcelo Monari Baccaro
	Victor Ribeiro Miguez
Cinemática inversa	Mirko del Piero Flores Sanchez
Equações dinâmicas por Newton-Euler	Marcelo Monari Baccaro
Equações dinâmicas por Gibbs-Appell	Lucas Satoru Aracena Perez
	Marcelo Monari Baccaro
Modelo CAD definitivo	Marcelo Monari Baccaro
Simulações do mecanismo no SolidWorks	Marcelo Monari Baccaro
Relatório final	Ana Cláudia Macedo Vianna Miachon
	Marcelo Monari Baccaro
Apresentação final do trabalho	Marcelo Monari Baccaro

9 Referências

- [1] Tsuneo Yoshikawa, *Foundations of Robotics: Analysis and Control*, The MIT Press, 1993.
- [2] Carvalho JCM., Ibrahim RC, Coelho TAH, *Mecanismos, máquinas e robô*, Elsevier, 2018.
- [3] John J. Craig, *Introduction to Robotics: Mechanics and Control*, Pearson, 2014.