

RICARDO SHEIJI YAMAMOTO

PROJETO DE UMA PLATAFORMA MÓVEL
PARA USO EM SIMULADORES
Parte II →

Def (10,0)

São Paulo
1998

RICARDO SHEIJI YAMAMOTO

**PROJETO DE UMA PLATAFORMA MÓVEL
PARA USO EM SIMULADORES
Parte II**

**Trabalho de formatura apresentado
à Escola Politécnica da Universidade
de São Paulo**

**Área de Concentração :
Eng. Mecânica**

**Orientador :
Tarcísio A. H. Coelho**

**São Paulo
1998**

Dedicamos este trabalho aos nossos pais, que sempre nos apoiaram. A eles, que dedicaram parte de sua vida para que nossos sonhos se concretizassem, que se sacrificaram para nos ajudar a sermos quem somos, o nosso sincero agradecimento e profundo reconhecimento.

Agradecimentos

Ao Prof. Dr. Tarcísio Antonio Hess Coelho pela orientação durante o desenvolvimento do trabalho.

Aos funcionários da oficina da Escola Politécnica da USP pelo auxílio fornecido para a construção do protótipo.

Ao amigo Sr. Lucas Chojiro Segawa pelos seus conselhos e incentivos na execução deste trabalho

Ao amigo Sr. Tony Kifua Lin pelo seu auxílio na execução deste trabalho e

A todos que direta ou indiretamente influenciaram na execução deste trabalho.

SUMÁRIO

Lista de Figuras

Lista de Símbolos

Resumo

1. Introdução	1
1.2. Objetivo	1
2. Plataforma de Stewart	2
3. Simuladores de voo e sua Viabilidade	5
4. Plataforma selecionada para estudo	8
5. O Programa de Simulação	11
5.1. Regras gerais	11
5.1.1. Movimento de translação	11
5.1.2. Movimento de rotação	12
5.1.3. Correção dos valores máximos	13
5.2. Barra	13
5.2.1. Cálculo das forças	13
5.3. Plataforma	15
5.3.1 Cálculo das forças	16
5.4. Resultados	18
6. Parâmetros de entrada	21
7. Protótipo	25
8. Resultados	31
9. Programa Working Model	38
10. Conclusões	42
10.1. O projeto	42
10.2. O programa	42
10.3. Sugestões de continuidade do projeto	43
11. Referências Bibliográficas	44

Apêndices

I – Listagem do programa Simula2D

II – Listagem do programa Simula3D

Lista de Figuras

2.1 : Estrutura paralela	2
2.2 : Estrutura serial	3
2.3 : Centro de usinagem	4
3.1 : Simulador de voo	6
3.2 : Detalhe do acionamento da plataforma de um simulador de voo	6
4.1 : Plataforma selecionada para projeto	8
4.2 : Foto dianteira do simulador da Thomson T&S.	9
4.3 : Foto do mecanismo do simulador da Thomson T&S.	9
4.4 : Foto traseira do simulador da Thomson T&S.	9
4.5: Eixos de translação	10
4.6 : Eixos de rotação	10
5.1 : Etapas do movimento de translação	12
5.2 : Forças atuantes na barra	14
5.3 : Forças atuantes na plataforma	15
5.4 : Dados de entrada do programa simula2D	18
5.5 : Dados de entrada da barra	18
5.6 : Trajetória da barra	19
5.7 : Força na atuador 2	19
5.8 : Força no atuador 4	20
5.9 : Força no atuador 7	20
6.1 : Esquema de projeto da plataforma da UBC	22
6.2 : Foto da plataforma projetada pela UBC	24
7.1 : Foto do protótipo	25
7.2 : Foto do protótipo	26
7.3 : Foto do protótipo	26
7.4 : Foto do protótipo	27
7.5 : Foto do protótipo	28
7.6 : Foto do protótipo	29
7.7 : Foto do protótipo	30
8.1 : Dados de entrada do movimento	31

8.2 : Dados da plataforma	32
8.3 : Posição final da plataforma	32
8.4 : Gráfico das forças nos atuadores	33
8.5 : Dados de entrada do movimento	34
8.6 : Dados da plataforma	34
8.7 : Posição final do movimento	35
8.8 : Gráfico da Força A	35
8.9 : Gráfico da Força B	36
8.10 : Gráfico da Força C	36
8.11 : Gráfico da Força D	37
9.1 : Gráfico de forças no atuador A	38
9.2 : Gráfico de forças no atuador B	39
9.3 : Gráfico de forças no atuador C	39
9.4 : Gráfico de forças no atuador D	40
9.5 : Gráfico da posição do ponto P	40
9.6 : Orientação do ponto P	41

Lista de Símbolos

$\vec{a}$ - Vetor aceleração do centro de gravidade da plataforma/barra

$\vec{F}_t$ - Vetor força atuante no ponto t da plataforma/barra

f_x - Força atuante, na direção x, no centro de gravidade da plataforma/barra

f_y - Força atuante, na direção y, no centro de gravidade da plataforma/barra

G - Posição do centro de gravidade da plataforma/barra

GXYZ - Sistema de coordenadas fixo na plataforma/barra

$\vec{I}$ - Versor na direção X

$\vec{J}$ - Versor na direção Y

$\vec{K}$ - Versor na direção Z

J_x - Momento de inércia da plataforma/barra em relação ao eixo X

J_y - Momento de inércia da plataforma/barra em relação ao eixo Y

J_z - Momento de inércia da plataforma/barra em relação ao eixo Z

m - Massa da plataforma/barra

Oxyz - Sistema de coordenadas global

ω_x - Velocidade angular em relação ao eixo X

ω_y - Velocidade angular em relação ao eixo Y

ω_z - Velocidade angular em relação ao eixo Z

Resumo

Os simuladores de voo desempenham uma função importante no treinamento de pilotos profissionais. Na área de entretenimento, os simuladores que agregam movimentos vem aumentando sua participação e se modernizando nos principais parques e centros de lazer.

Este trabalho se baseia na teoria de plataformas Stewart, para o desenvolvimento de um programa de computador. O programa possibilitará o projeto de uma plataforma móvel, com quatro graus de liberdade, para uso em simuladores de voo na área de entretenimento.

O programa, desenvolvido para funcionar em conjunto com o Matlab, é baseado na teoria de cinemática inversa. Ele mostra, através da determinação do posicionamento da plataforma, as forças que os atuadores lineares terão de desenvolver para realizar o movimento necessário. Fica assim possível, o projeto do mecanismo. Os dados de entrada foram baseados em um simulador já construído. A plataforma deve descrever movimentos que transmitam ao usuário sensações reais de velocidade e aceleração.

1. Introdução

Os simuladores têm como objetivo imitar uma situação real e é largamente empregado para treinamento de pilotos. Neste caso, existe uma máquina que simula os mesmos movimentos de um caso real de acordo com as ações do piloto, de modo que este sinta como se estivesse em uma máquina real.

O seu uso para treinamento é muito importante visto que é possível treinar pessoas sem pôr em risco as suas vidas, e também sem perder milhões em dinheiro com o uso de um aparelho real.

Nos últimos anos, com o avanço da tecnologia, o uso de simuladores, por meio de software, tem crescido bastante. Estes imitam, por meio de imagens, uma situação imaginária e pode-se encontrar muitos na área automobilística, de aviões, tanques, helicópteros etc.

Recentemente, o uso de simuladores completo, ou seja, software e máquina, tem sido utilizado também na área recreativa. Porém as máquinas são limitadas de acordo com a situação que se deseja imitar, onde dois ou três graus de mobilidade são suficientes.

1.2. Objetivo

Este trabalho de formatura tem como objetivo projetar uma plataforma móvel, para uso em simuladores, para fins recreativos.

2. Plataforma de Stewart

A plataforma de Stewart é um tipo de estrutura paralela. Chamamos de estrutura paralela uma estrutura cujos segmentos formam cadeias cinemáticas fechadas (segmentos interligados na forma de elos). Diferentemente das estruturas seriais, que são constituídas de segmentos e juntas dispostos em sequência, formando uma cadeia cinemática aberta. A plataforma de Stewart, no caso mais genérico, consiste de uma base e uma plataforma móvel unidas por segmentos formados de atuadores prismáticos e juntas esféricas.

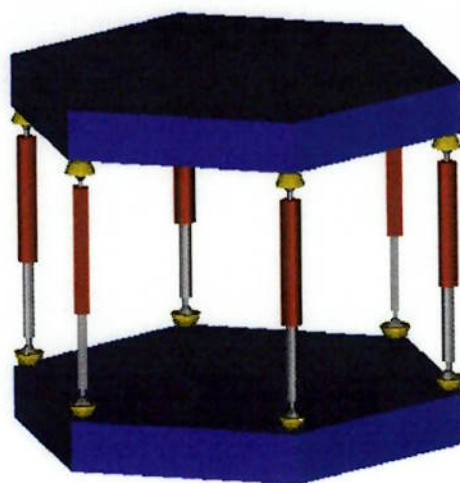


Fig. 2.1 : Estrutura paralela

As vantagens das estruturas paralelas , comparando-as com estruturas seriais, são :

- maior leveza;
- alta rigidez;
- melhor performance dinâmica;
- pequena inércia;

- grande precisão (capacidade de se aproximar do ponto determinado);
- repetibilidade;
- podem operar a altas velocidades;
- e os erros nas articulações não são aditivos.

As desvantagens são :

- menor volume de trabalho;
- e dificuldade de se estudar a cinemática direta (determinação da posição da plataforma tendo como entrada os comprimentos das barras).

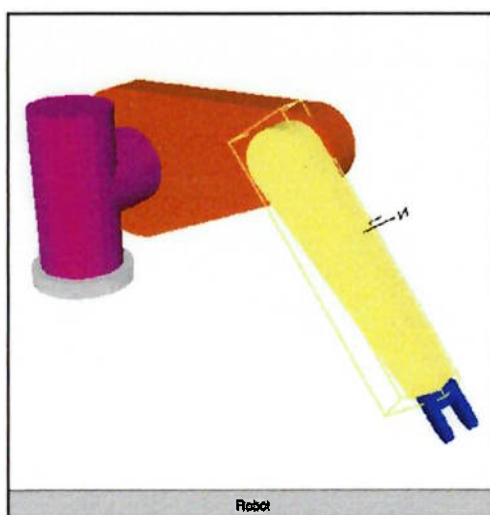


Fig. 2.2 : Estrutura serial

A aplicações da plataforma de estrutura paralela são inúmeras : simuladores de vôo, aparelhos médicos e cirúrgicos, aparelhos aeroespaciais, centros de usinagem etc.

Construtivamente, os segmentos que unem as plataformas tem seus comprimentos variáveis. Pode-se usar , por exemplo, parafusos de esferas ou atuadores hidráulicos, conforme a aplicação.

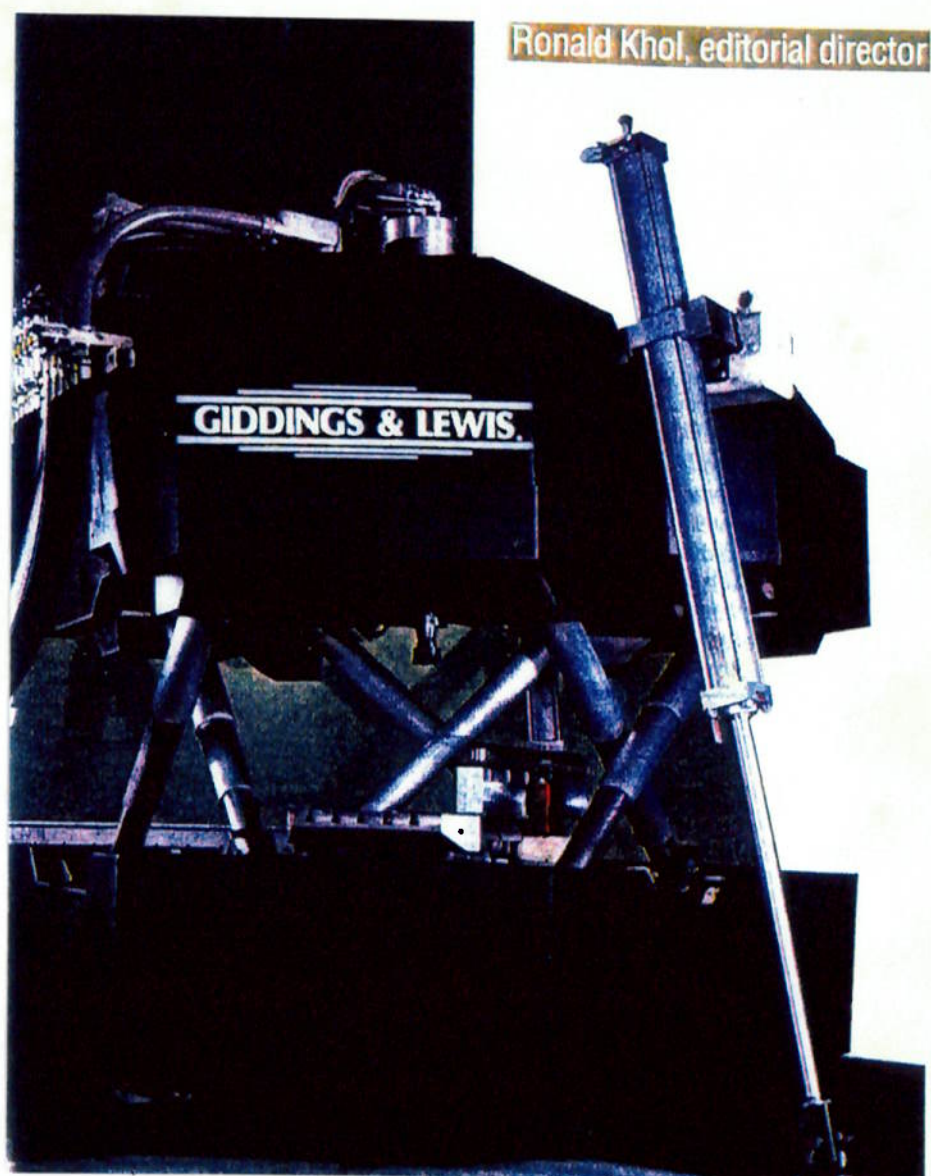


Fig. 2.3 : Centro de usinagem

3. Simuladores de voo e sua viabilidade

Simular significa representar com semelhança. Simuladores são equipamentos que tentam imprimir sensações ou comportamentos de uma situação real. Podemos dividir o campo de utilização de simuladores em dois níveis : o de uso profissional e o de entretenimento.

Os simuladores de uso profissional são muito utilizados para treinamento militar e de aviação. A simulação torna-se verdadeiramente necessária quando existe a necessidade de treinar pessoas em situações que existam risco de vida. O simulador não é um bem de consumo de massa. O custo de um simulador normalmente é inferior ao de se criar situações reais, principalmente em se tratando de vôos aéreos ou espaciais, e situações militares. Mas, mesmo que o custo seja superior, financeiramente, ele se torna desprezível frente a capacidade de se pôr uma vida no limite da sobrevivência e trazê-la de volta com segurança. Percebe-se, então, a necessidade e a viabilidade econômica dos simuladores. No filme "Apollo 13" podemos perceber a importância do papel do simulador no salvamento das vidas da tripulação da nave. Um simulador é o único lugar onde um piloto pode cometer uma falha e discuti-la, dando a ele uma nova chance de descobrir a causa da falha. Um bom instrutor fará o melhor uso de um simulador para assegurar que pilotos reajam a emergências de forma organizada e previsível, tão bem quanto em situações normais.

Os simuladores de uso para entretenimento estão em constante desenvolvimento no mercado atual. Os simuladores utilizados em parques de diversão são considerados investimentos de médio a longo prazo e os custos elevados do equipamento e da manutenção devem ser compensados pela capacidade de atrair o público constantemente. Isso é possível, se conseguirmos a um baixo custo ao usuário, gerar sensações reais de situações improváveis no cotidiano, como pilotar jatos, combater em guerras, participar de corridas de F-1 e "jet-skis".



Fig. 3.1 : Simulador de voo



Fig. 3.2 : Detalhe do acionamento da plataforma de um simulador de voo

O mercado possui diversas empresas neste ramo, como a Namco, Sega, .mdf Craftwerks & Design, Yamaha, Takesugi Engineering e Thomson Training & Simulation, e envolve diferentes tecnologias, como a geração de imagens, o sistema de movimentação, o som, as características do equipamento ou situação a ser simulada e a integração dessas tecnologias.

Os simuladores para uso profissional necessitam de seis graus de mobilidade para representar os seis graus de liberdade de um corpo no espaço. Os simuladores para entretenimento podem ter um número variado de graus de mobilidade. Corridas de carro ou equipamentos aquáticos podem ser simulados com apenas dois graus de mobilidade, com rotação em dois eixos (movimentação limitada sobre um plano). Simuladores de vôo podem ser construídos com três graus de mobilidade. Os atuadores prismáticos são geralmente atuadores pneumáticos ou hidráulicos.

4. Plataforma selecionada para estudo

A plataforma selecionada é baseada num simulador já existente, produzido pela Thomson Training & Simulation. Este simulador consiste em uma cabina sobre uma plataforma sustentada por três atuadores lineares hidráulicos. Nesta cabina, totalmente fechada, são colocadas oito pessoas simultaneamente, e existe uma tela onde é projetado um filme (corrida automobilística) sem a interação do usuário. Os movimentos do veículo no filme são sincronizados com os movimentos da plataforma, criando uma sensação de movimento real com o veículo. Apesar de possuir apenas três graus de mobilidade, esse simulador consegue transmitir praticamente todos os movimentos do automóvel, inclusive de aceleração.

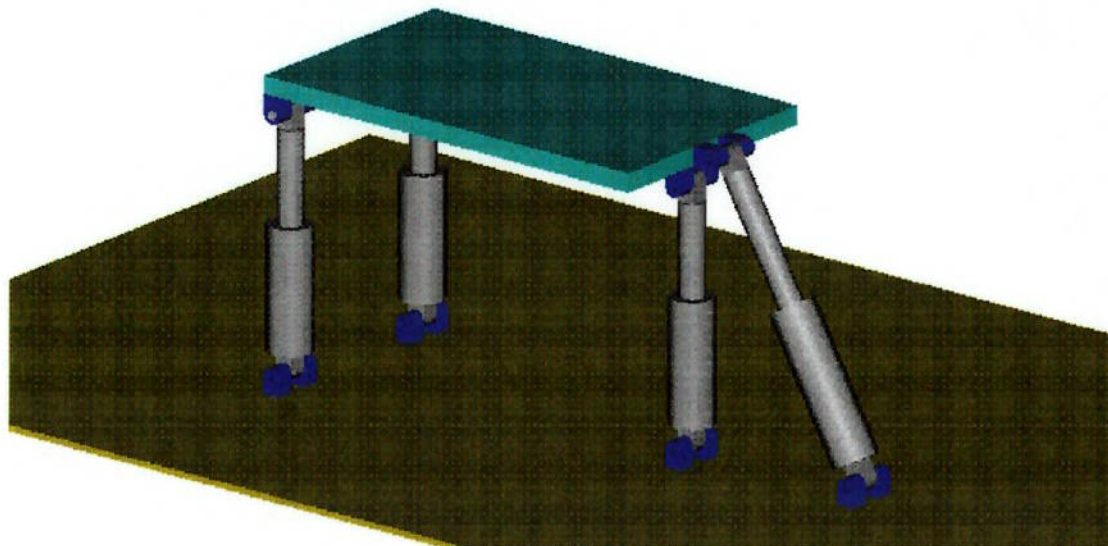


Fig. 4.1 : Plataforma selecionada para projeto



Fig. 4.2 : Foto dianteira do simulador da Thomson T&S.



Fig. 4.3 : Foto do mecanismo do simulador da Thomson T&S.



Fig. 4.4 : Foto traseira do simulador da Thomson T&S.

Através do estudo prévio de um mecanismo plano e este simulador, a plataforma selecionada para estudo e projeto está abaixo demonstrada. A plataforma possui quatro graus de mobilidade, e acredita-se que serão suficientes para transmitir as sensações necessárias como requisito de um simulador. Este grau a mais em relação a plataforma comercial existente permite uma livre movimentação vertical, conforme a figura. A sua exeqüibilidade física é estruturada no simulador já existente.

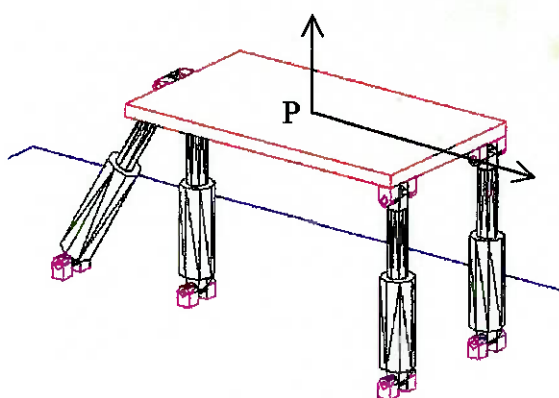


Fig. 4.5: Eixos de translação

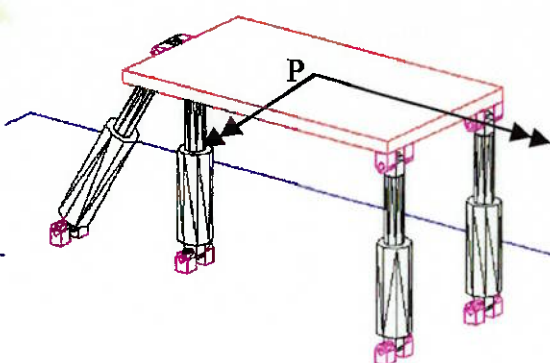


Fig. 4.6 : Eixos de rotação

Definido a plataforma de trabalho, a próxima etapa é a programação para a simulação de um movimento. Para a simulação será utilizado o método da cinemática inversa, ou seja, a partir da posição desejada da plataforma são calculados as forças e comprimentos dos atuadores para a sua execução do movimento.

5. O Programa de Simulação

O objetivo do programa de simulação é calcular as forças necessárias nos atuadores, ao longo do movimento, e também os seus respectivos comprimentos. O programa foi desenvolvido no software MATLAB e teve duas etapas :

- 1) Desenvolvimento da simulação para uma barra em duas dimensões, com três graus de liberdade.
- 2) A segunda etapa foi a evolução do programa para a simulação da plataforma escolhida em três dimensões.

A seguir será dado maiores detalhes das etapas do desenvolvimento do programa.

5.1. Regras gerais

Os movimentos, tanto de translação quanto de rotação, da plataforma / barra são baseados no deslocamento de um ponto da plataforma/barra, sendo que este ponto é o seu centro de gravidade.

Para os seus movimentos de translação e rotação, estes devem obedecer certas leis descritas abaixo.

5.1.1. Movimento de Translação

O movimento de translação possui três etapas :

- 1) Aceleração constante até que a velocidade máxima seja atingida;
- 2) Velocidade constante do centro de gravidade da plataforma e
- 3) Desaceleração constante até o término do movimento.

As três etapas podem ser vistas no gráfico (Velocidade x Tempo) mostrado abaixo :

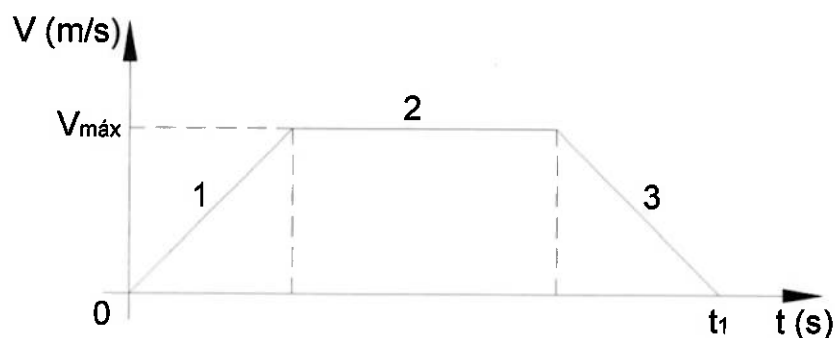


Fig. 5.1 : Etapas do movimento de translação

5.1.2. Movimento de Rotação

Do mesmo modo que o movimento de translação, a rotação possui as mesmas etapas, ou seja :

- 1) Aceleração angular constante até que a velocidade angular máxima seja atingida;
- 2) Velocidade angular constante e
- 3) Desaceleração angular constante até o término do movimento.

Os valores máximos de aceleração e velocidade, tanto linear quanto angular, são dados de entrada do programa, sendo portanto definidos pelo usuário.

Uma outra característica do programa é que os três movimentos devem terminar ao mesmo tempo, e para que isso ocorra é necessário que os valores máximos sejam corrigidos conforme descrito abaixo.

5.1.3. Correção dos Valores Máximos

De posse dos valores dos tempos necessários para terminar cada movimento, obtidos da cinemática, faz-se uma correção dos valores máximos de velocidade e aceleração para que todos os movimentos terminem juntos. Supondo que o movimento de translação demore t_1 segundos para terminar e o movimento de rotação, em torno do eixo y, demore $t_2 < t_1$ segundos, temos que corrigir os valores de aceleração angular e velocidade angular de modo que ambos os movimentos terminem no mesmo instante. No caso, o primeiro deverá ser multiplicado por $(t_2/t_1)^2$ e o segundo por (t_2/t_1) .

5.2. Barra

O programa foi desenvolvido para que a barra tivesse três atuadores, sendo dois em uma ponta da barra e o outro atuador na outra ponta.

Abaixo será demonstrado como foi feito o cálculo das forças nos atuadores.

5.2.1. Cálculo das forças

O cálculo das forças atuantes são calculados a partir do Teorema do Movimento do Baricentro (TMB) e do Teorema do Momento Angular (TMA).

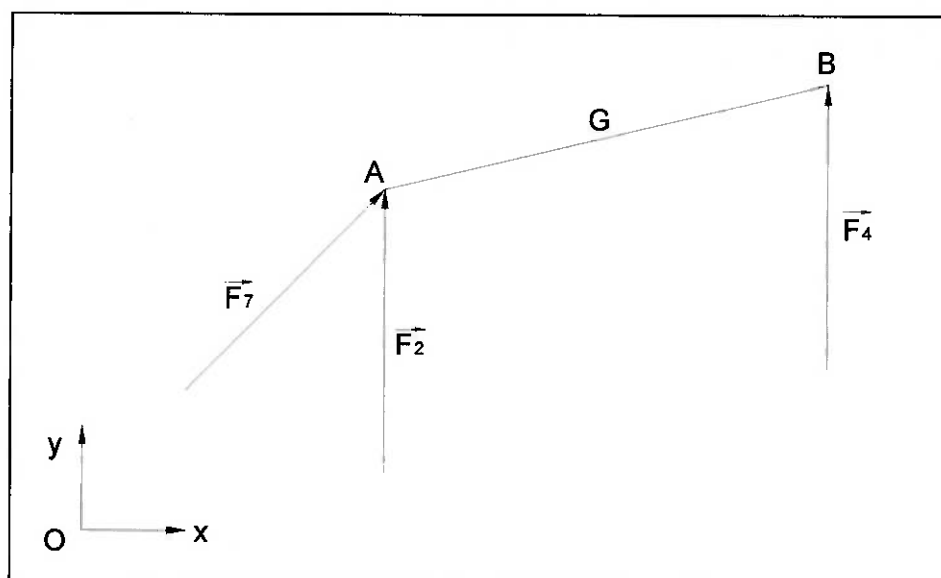


Fig. 5.2 : Forças atuantes na barra

Para a barra, temos a atuação das forças dos atuadores representado na figura 5.2, e portanto a partir do Teorema do Movimento do Baricentro (TMB), temos :

$$\vec{F} = m \cdot \vec{a}$$

na direção x, temos :

$$F_{2x} + F_{4x} + F_{7x} + f_x = m \cdot a_x \quad (1)$$

e na direção y :

$$F_{2y} + F_{4y} + F_{7y} + f_y = m \cdot a_y + m \cdot g \quad (2)$$

Do Teorema do Momento Angular (TMA) temos :

$$\frac{d\vec{H}}{dt} = \vec{M}_G$$

onde $\vec{H} = J_z \cdot \omega_z \cdot \vec{K}$ e

$$\vec{M}_G = (A - G) \wedge \vec{F}_2 + (B - G) \wedge \vec{F}_4 + (A - G) \wedge \vec{F}_7$$

portanto temos :

$$\vec{M}_G = J_z \dot{\omega}_z \quad (3)$$

Com as equações (1), (2) e (3), temos três incógnitas e três equações, o que nos fornece condições de calcular os módulos das forças $\vec{F}_2, \vec{F}_4$ e $\vec{F}_7$

5.3. Plataforma

Tendo escolhido o tipo de plataforma, é necessário calcular as forças dos atuadores em diversos instantes do movimento.

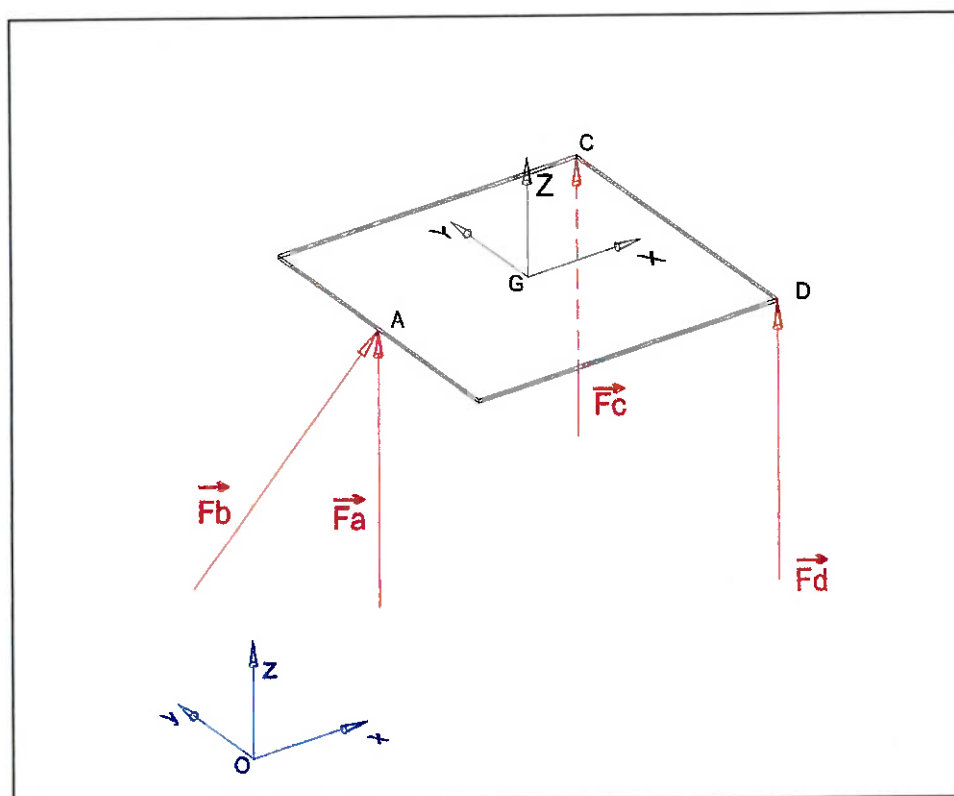


Fig. 5.3 : Forças atuantes na plataforma

Como já foi visto, a plataforma possui quatro graus de liberdade : translação nos eixos x e z, e rotação nos eixos X e y. Todos os seus movimentos são baseados no centro de gravidade da plataforma.

As leis de movimento de translação e rotação, descritas nas regras gerais , são aplicadas para a plataforma. E do mesmo modo que para a barra, os três movimentos da plataforma devem terminar ao mesmo tempo, portanto será necessário fazer a correção dos seus valores máximos como já foi descrito.

5.3.1. Cálculo das Forças

O cálculo das forças atuantes são calculados a partir do Teorema do Movimento do Baricentro (TMB) e do Teorema do Momento Angular (TMA).

Do Teorema do Movimento do Baricentro (TMB), temos :

$$\vec{F} = m.\vec{a}$$

na direção x, temos :

$$F_{ax} + F_{bx} + F_{cx} + F_{dx} + f_x = m.a_x \quad (1)$$

e na direção z :

$$F_{az} + F_{bz} + F_{cz} + F_{dz} + f_z = m.a_z + m.g \quad (2)$$

Do Teorema do Momento Angular (TMA) temos :

$$\frac{d\vec{H}}{dt} = \vec{M}_G$$

onde $\vec{H} = J_x \cdot \omega_x \cdot \vec{I} + J_y \cdot \omega_y \cdot \vec{J} + J_z \cdot \omega_z \cdot \vec{K}$ e

$$\vec{M}_G = (A - G) \wedge \vec{F}a + (A - G) \wedge \vec{F}b + (C - G) \wedge \vec{F}c + (D - G) \wedge \vec{F}d$$

portanto na direção de $\vec{I}$, temos :

$$\vec{M}_G \cdot \vec{I} = J_x \cdot \dot{\omega}_x + (J_z - J_y) \cdot \omega_y \cdot \omega_z \quad (3)$$

e na direção de $\vec{J}$, temos :

$$\vec{M}_G \cdot \vec{J} = J_y \cdot \dot{\omega}_y + (J_x - J_z) \cdot \omega_x \cdot \omega_z \quad (4)$$

Com as equações (1), (2), (3) e (4), temos quatro incógnitas e quatro equações, o que nos fornece condições de calcular os módulos das forças $\vec{F}a, \vec{F}b, \vec{F}c$ e $\vec{F}d$

5.4. Resultados

Será apresentado, neste momento, somente os resultados para o programa da barra porque o resultado do programa da plataforma será utilizado posteriormente.

As próximas seis figuras mostram os dados de entrada do programa da barra e as suas saídas. Estas são compostas de quatro gráficos, sendo uma da trajetória da barra e as outras da variação das forças dos atuadores em função do tempo.

Barra

Coordenadas (m):

Número de Interações : **30**

Ao	Bo	Co
X 10	X 30	X 0
Y 0	Y 0	Y 10

Coordenadas Iniciais (m):

Xp **20** Yp **20** Theta **0** °

Coordenadas Finais (m):

Xp **35** Yp **50** Theta **45** °

Limites Máximos :

Aceleração	Velocidade	Aceleração Angular	Velocidade Angular
1 m/s ²	10 m/s	0.1 rad/s ²	20 rad/s ²

Forças Atuantes :

Fx **2** N Fy **3** N

Simular

Sair

Fig. 5.4 : Dados de entrada do programa simula2D

Dados Relativos à Barra

Entrada

Comprimento da Barra : **20** m.

Massa : **5** kg

Momento de Inercia : **27** kg.m²

Simular

Fig. 5.5 : Dados de entrada da barra

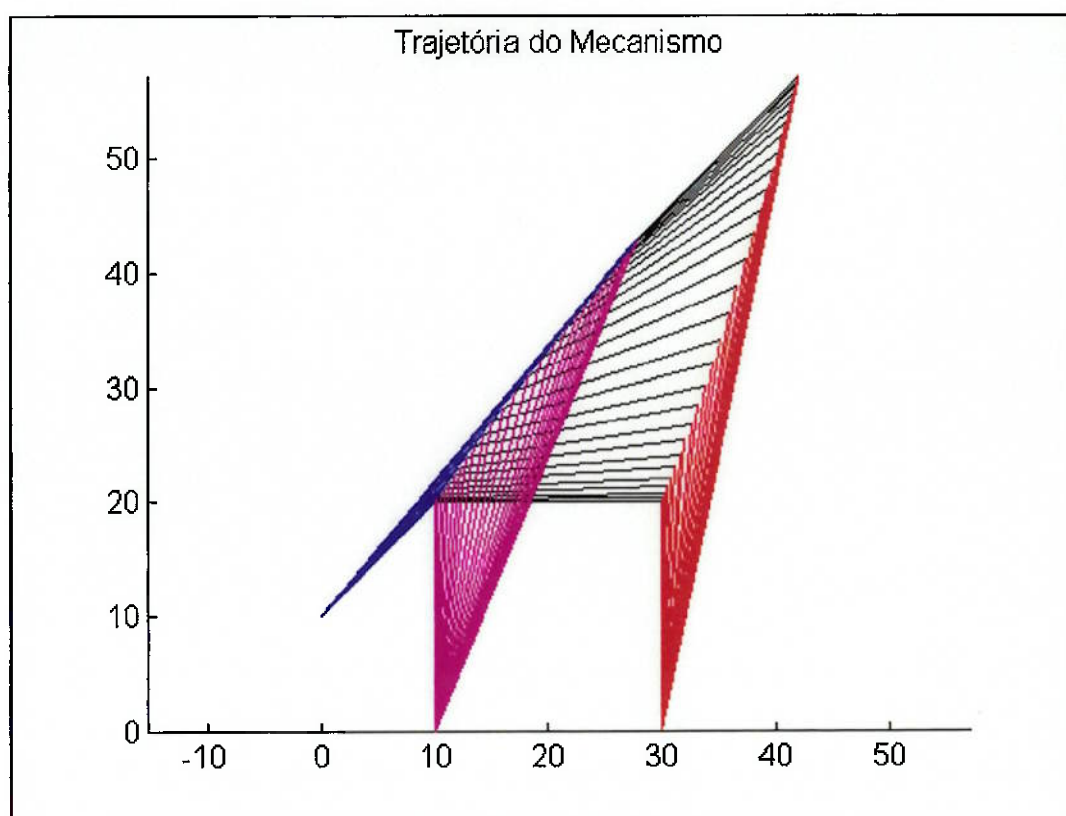


Fig. 5.6 : Trajetória da barra

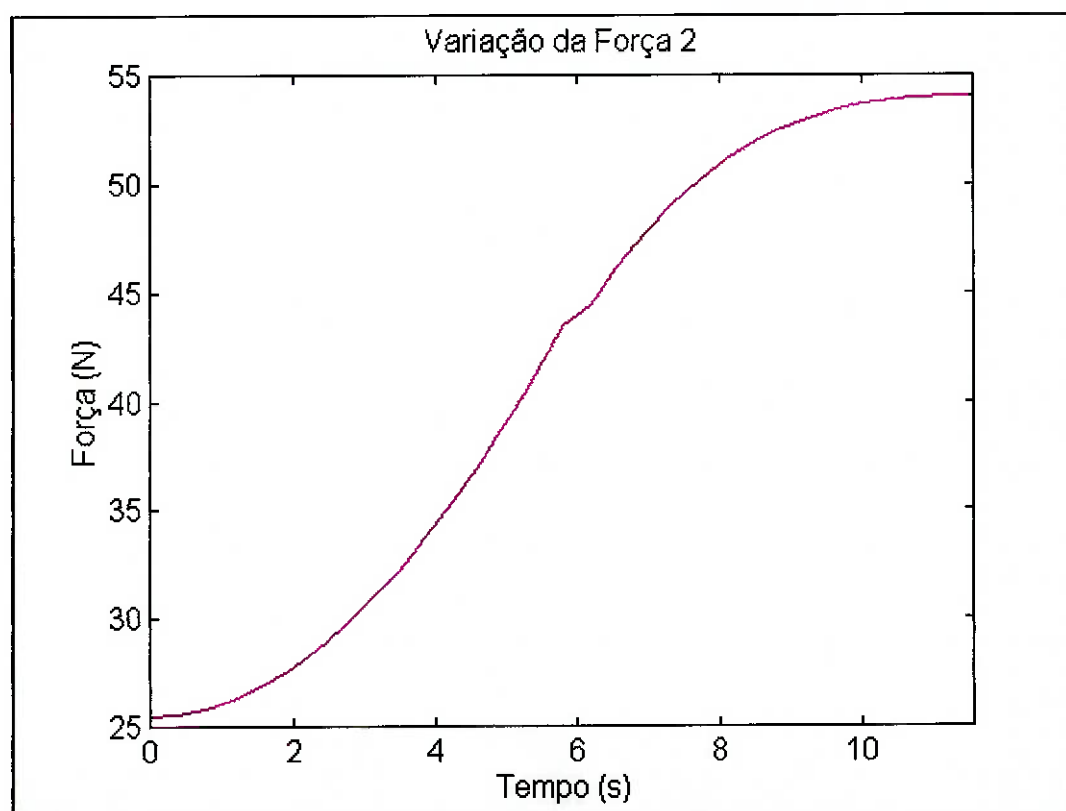


Fig. 5.7 : Força na atuador 2

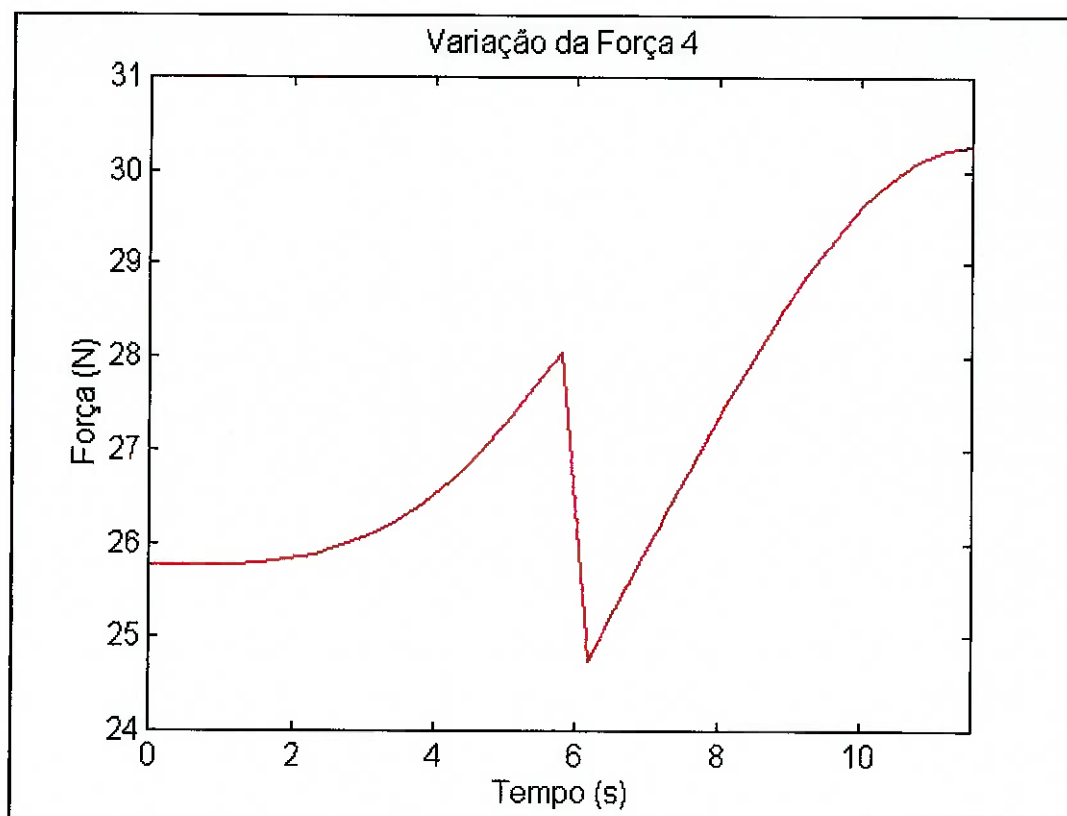


Fig. 5.8 : Força no atuador 4

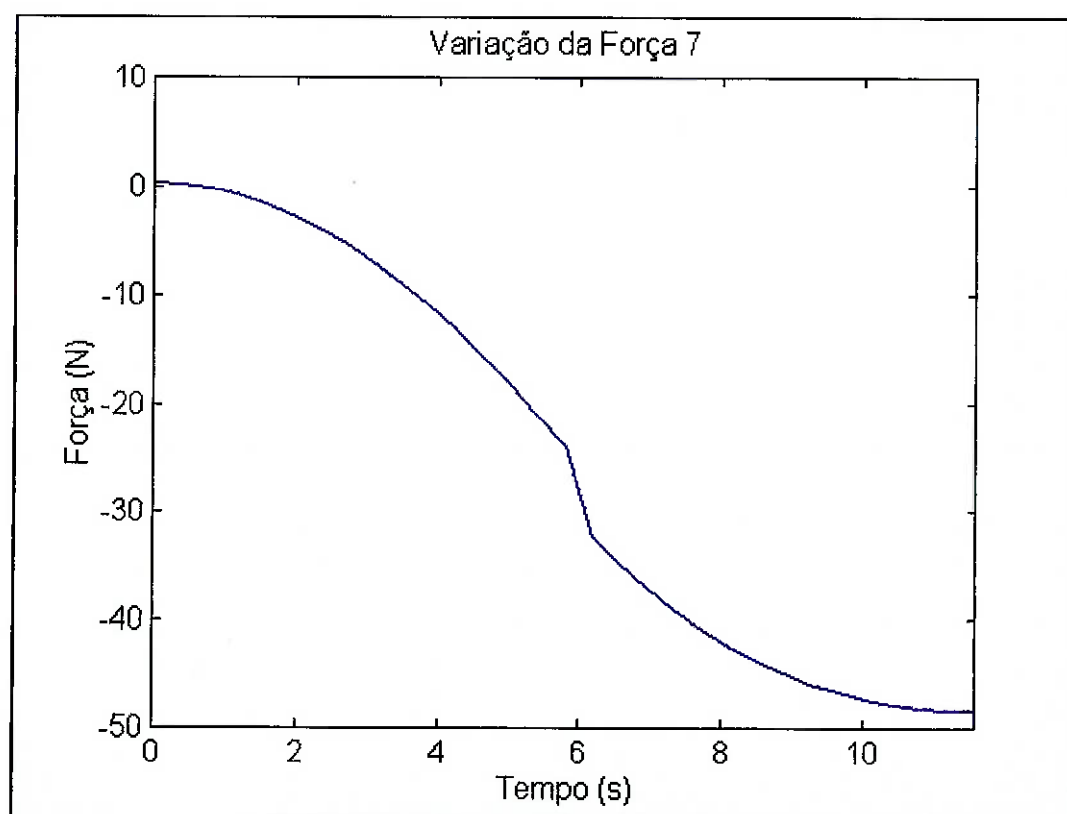


Fig. 5.9 : Força no atuador 7

6. Parâmetros de entrada

Os dados numéricos utilizados para o teste e aferição do programa foram baseados em um simulador construído pelo departamento de engenharia elétrica e de computação da University of British Columbia (UBC).

Existem algumas semelhanças entre o simulador projetado pela UBC e a plataforma escolhida para este trabalho:

Semelhanças :

- o simulador carrega apenas o operador sentado em uma cadeira;
- os movimentos devem criar, ao usuário, sensações próximas às reais de posição, velocidade e aceleração.

Diferenças :

- o simulador da UBC possui seis graus de mobilidade, contra quatro da plataforma deste trabalho;
- o simulador é montado de forma invertida, com sua base fixa localizada na parte superior, conforme a figura. Segundo a descrição do projeto, este tipo de montagem permite o uso de cilindros hidráulicos mais estreitos e menor fluxo hidráulico, portanto, menor custo.

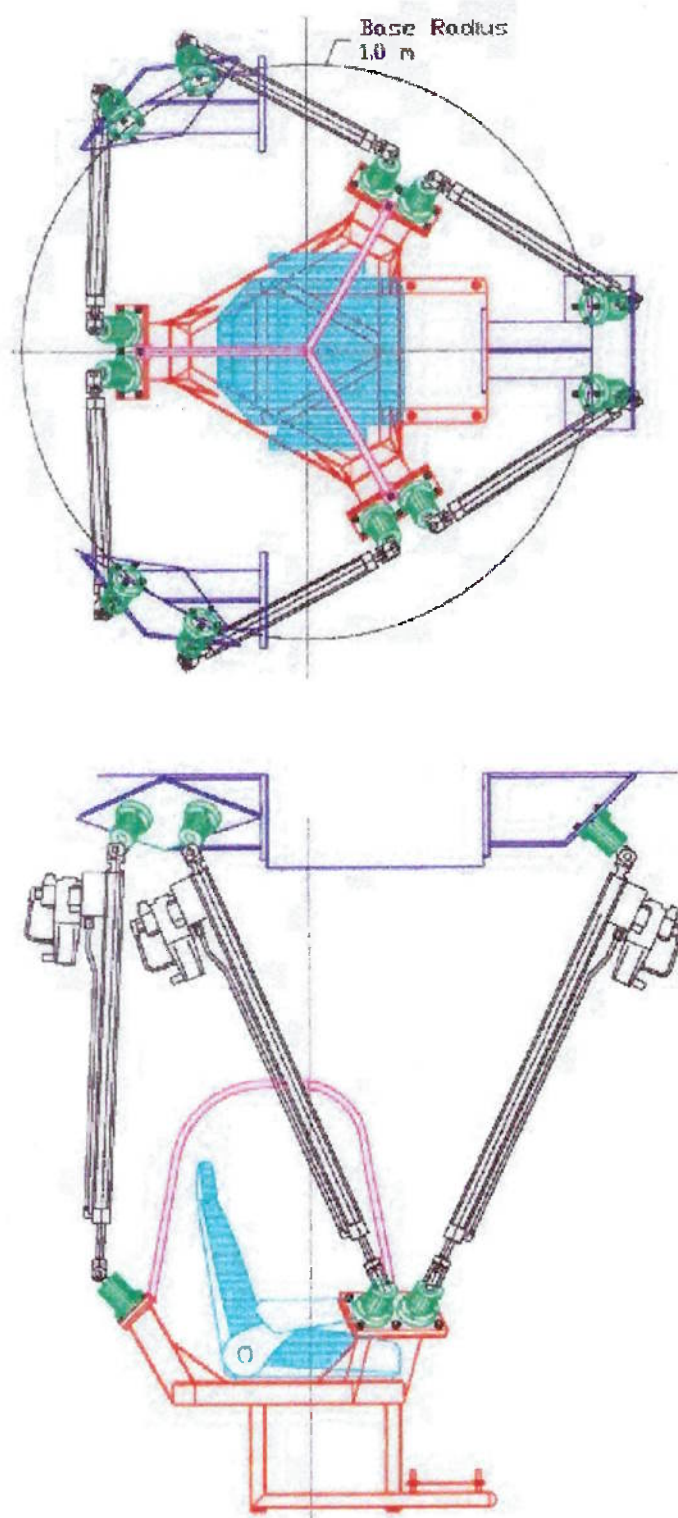


Fig. 6.1 : Esquema de projeto da plataforma da UBC

O simulador de plataforma Stewart da UBC utiliza :

- seis atuadores hidráulicos de 1,5 polegas ($3,81 \cdot 10^{-2}$ m) de diâmetro e 54 polegadas (1,37 m) de curso;
- juntas em U com rolamentos, em 120 graus
- cada atuador possui limite de 4000 N a uma velocidade de 1 m/s e 8000 N a uma velocidade nula

Os parâmetros utilizados deste projeto foram :

- massa da plataforma (carregada) : 250 kg
- aceleração linear : +/- 1g ($9,81 \text{ m/s}^2$)
- aceleração angular : +/- 400 graus/s²
- velocidade linear : +/- 1 m/s
- velocidade angular : +/- 30 graus/s

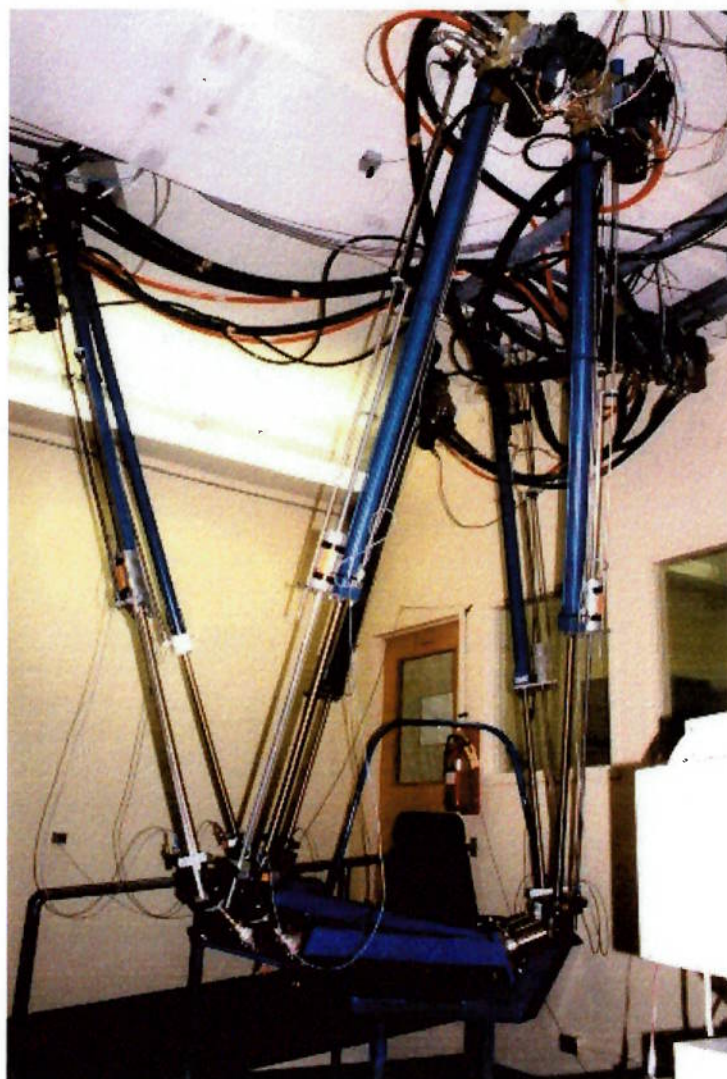


Fig. 6.2 : Foto da plataforma projetada pela UBC

7. Protótipo

Protótipos são usados para avaliações e testes que podem levar a revisões e aperfeiçoamentos, ou seja, o protótipo permitirá que o projeto seja verificado de forma concreta.

No caso da plataforma, o protótipo foi construído com o objetivo de ter um maior contato com os movimentos que a plataforma poderia realizar e também visualizar suas restrições construtivas.

As suas dimensões foram baseadas em uma cadeira, supondo que este fosse o único assento do usuário, e portanto o simulador tivesse apenas um assento e os atuadores.

A sua construção foi realizada na oficina da Escola Politécnica da USP e abaixo pode-se ver diversas posições que a plataforma permite.



Fig. 7.1 : Foto do protótipo



Fig. 7.2 : Foto do protótipo



Fig. 7.3 : Foto do protótipo

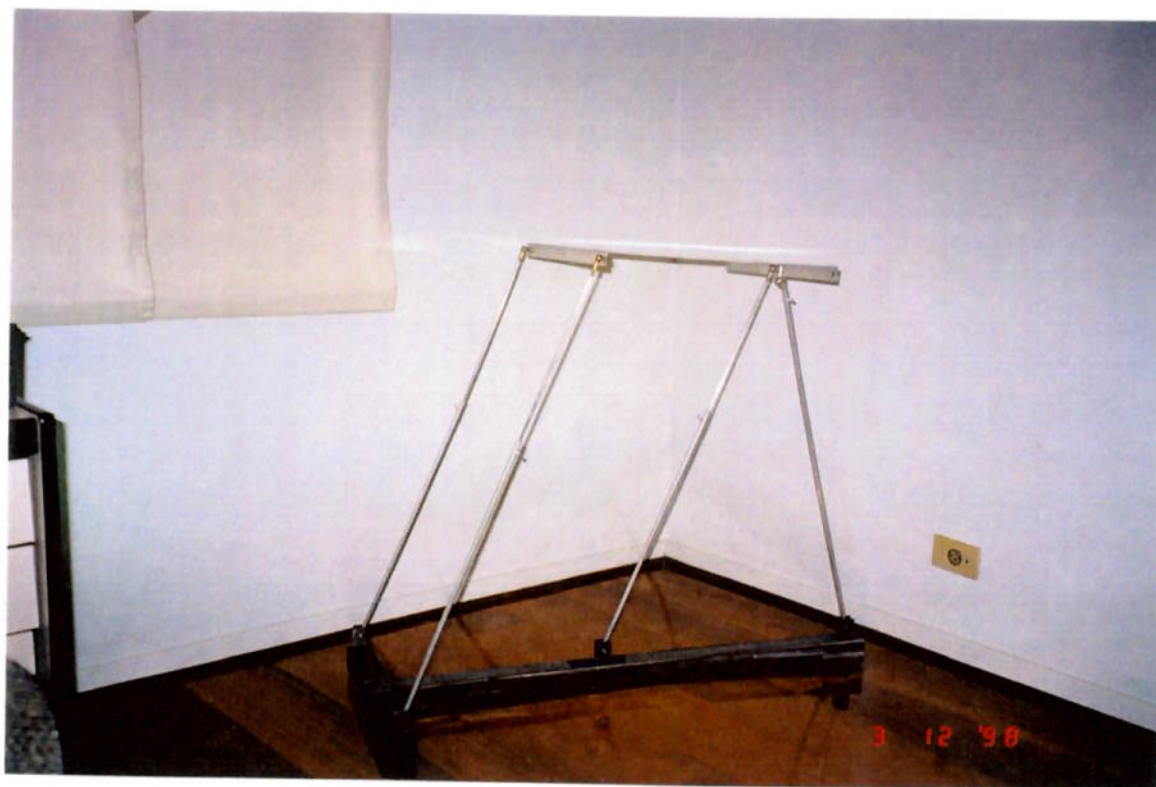


Fig. 7.4 : Foto do protótipo



Fig. 7.5 : Foto do protótipo



Fig. 7.6 : Foto do protótipo



Fig. 7.7 : Foto do protótipo

8. Resultados

Foram realizadas duas simulações para exemplificar o uso do programa. Para melhor visualização, as forças nos atuadores foram reunidas em um único gráfico.

Simulação 1 : movimento vertical da plataforma

Dados de entrada :

Plataforma

Coordenadas (m):

	Ao	Bo	Co	Do
X	0	-0.555	0.45	0.45
Y	0	0	0.225	-0.225
Z	0	0	0	0

Número de Interações : 50

Coordenadas Iniciais (m):

Xp 0.225 Yp 0 Zp 0.8 Theta(*) X 0 Y 0

Coordenadas Finais (m):

Xp 0.225 Yp 0 Zp 1 Theta(*) X 0 Y 0

Limites Máximos :

Aceleração

10 m/s²

Aceleração Angular

X 6.981 rad/s² Y 6.981 rad/s²

Velocidade

1 m/s

Velocidade Angular

X 0.524 rad/s Y 0.524 rad/s

Simular

Sair

Fig. 8.1 : Dados de entrada do movimento

Dados Relativos à Plataforma

Entrada

Dimensões da Plataforma : X m. Y m.

Massa : kg

Momento de Inercia : J_x kg.m² J_y kg.m²
 J_z kg.m²

Forças Atuantes :
F_x N F_z N

Simular

Fig. 8.2 : Dados da plataforma

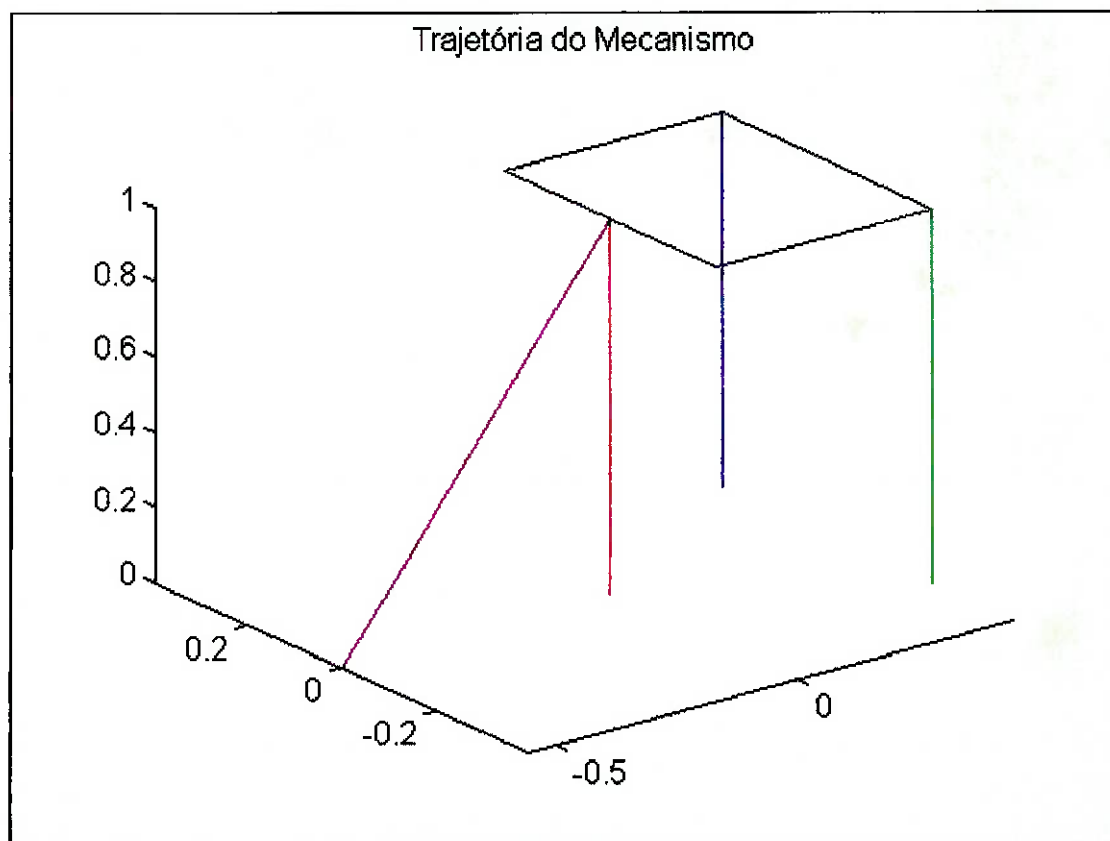


Fig. 8.3 : Posição final da plataforma

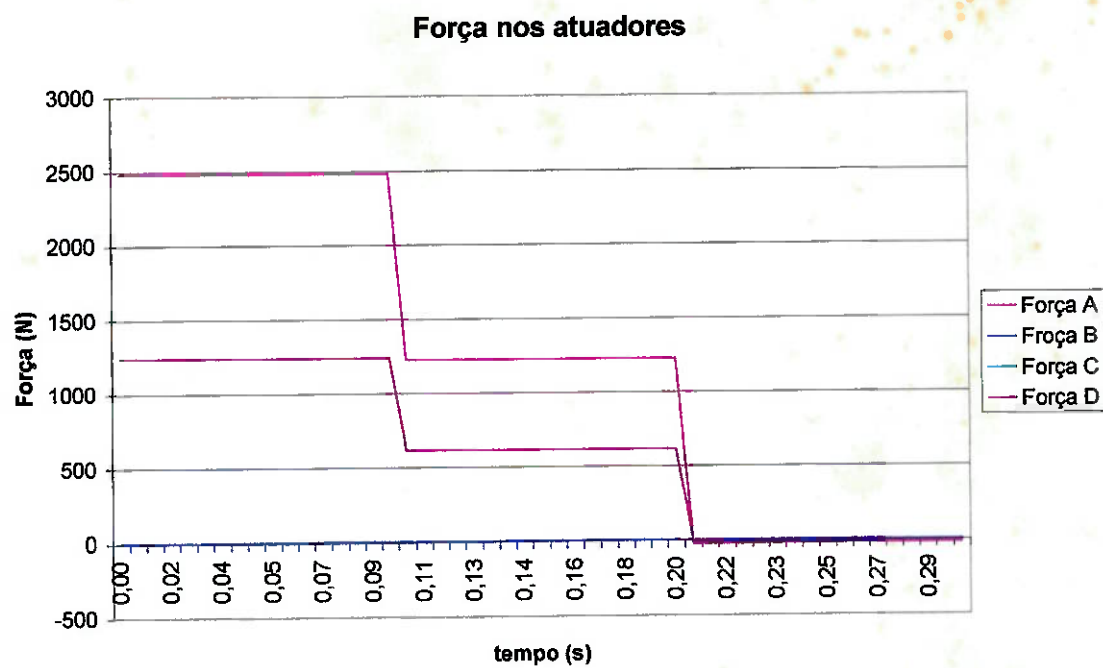


Fig. 8.4 : Gráfico das forças nos atuadores

Simulação 2 : Movimento genérico

Dados de entrada :

Plataforma

Coordenadas (m):

	Ao	Bo	Co	Do
X	0	-0.555	0.45	0.45
Y	0	0	0.225	-0.225
Z	0	0	0	0

Número de Interações : 50

Coordenadas Iniciais (m):

Xp 0.225 Yp 0 Zp 0.8 Theta(°) X 0 Y 0

Coordenadas Finais (m):

Xp 0.425 Yp 0 Zp 1.2 Theta(°) X 10 Y -25

Limites Máximos :

Aceleração: 10 m/s²

Aceleração Angular: X 6.981 rad/s² Y 6.981 rad/s²

Velocidade: 1 m/s

Velocidade Angular: X 0.524 rad/s Y 0.524 rad/s

Simular

Sair

Fig. 8.5 : Dados de entrada do movimento

Dados Relativos à Plataforma

Entrada

Dimensões da Plataforma : X 0.45 m. Y 0.45 m.

Massa : 250 kg

Momento de Inercia : Jx 4.23 kg.m² Jy 4.23 kg.m²
Jz 8.46 kg.m²

Forças Atuantes : Fx 0 N Fz 0 N

Simular

Fig. 8.6 : Dados da plataforma

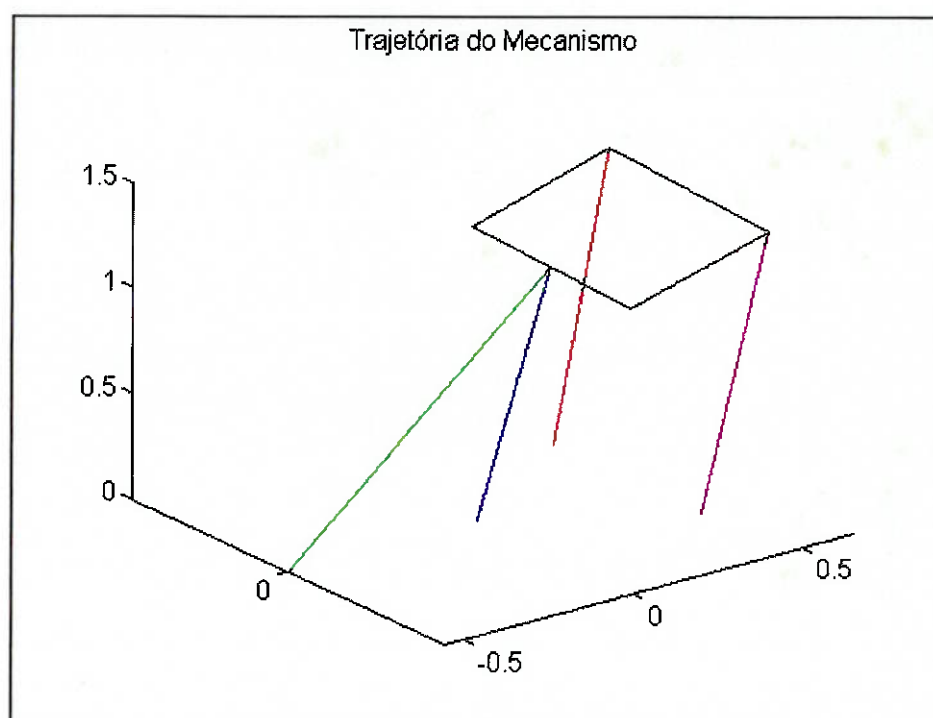


Fig. 8.7 : Posição final do movimento

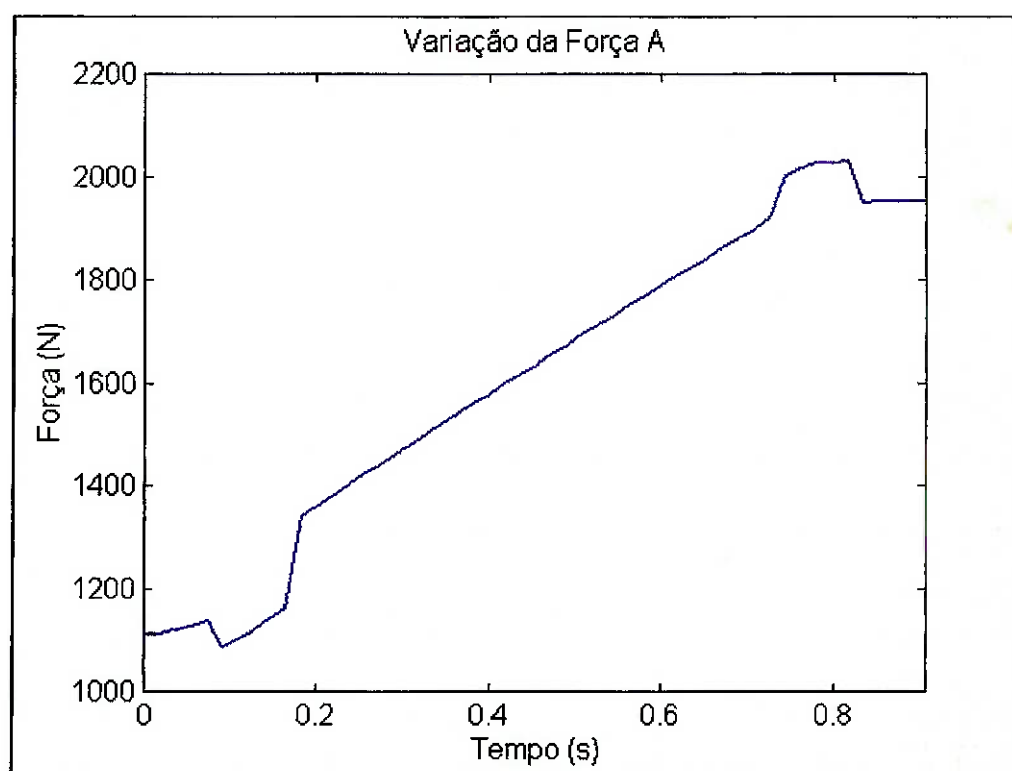


Fig. 8.8 : Gráfico da Força A

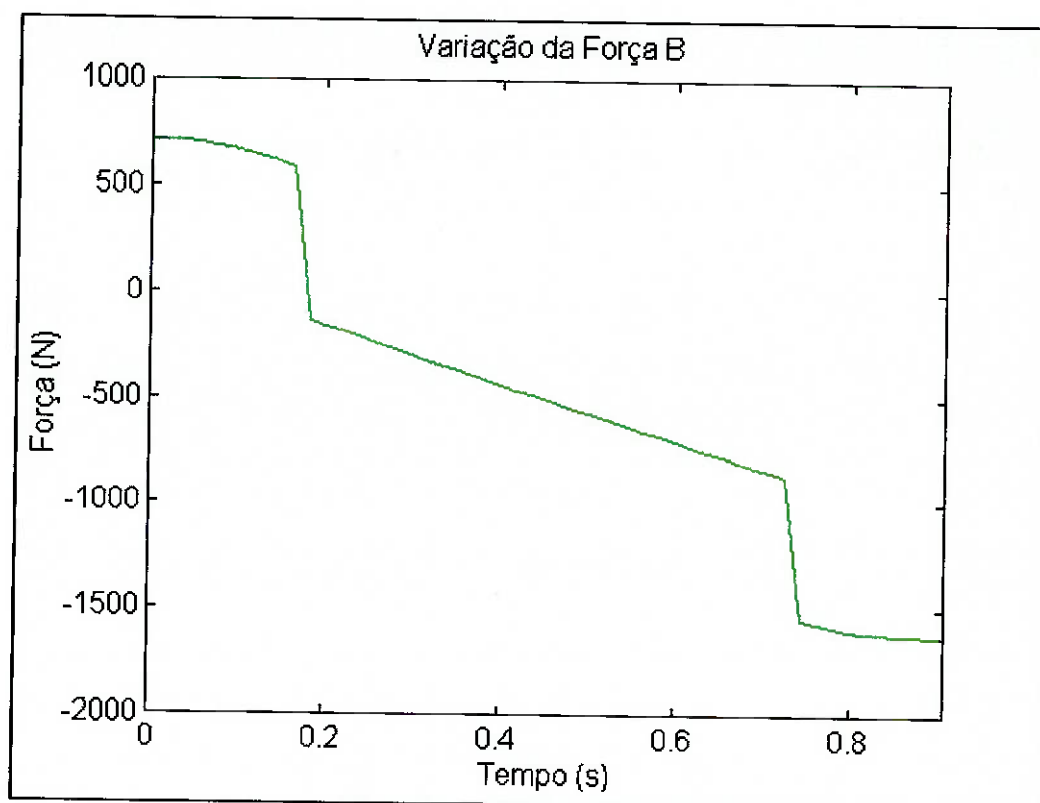


Fig. 8.9 : Gráfico da Força B

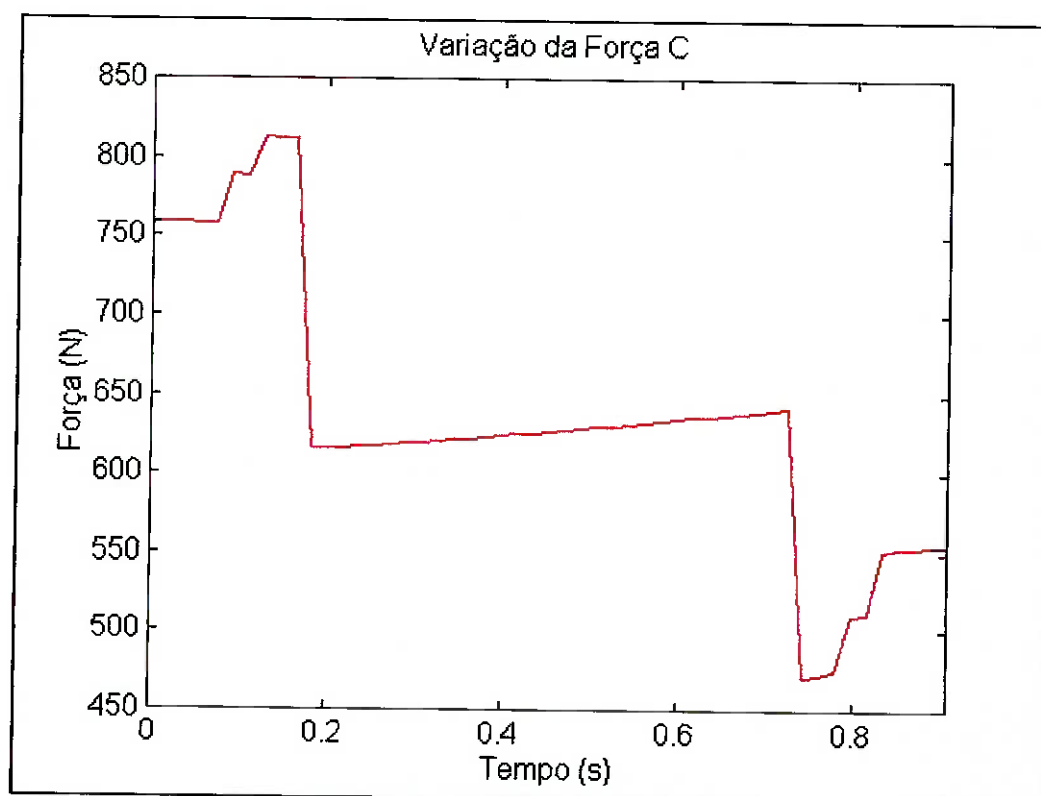


Fig. 8.10 : Gráfico da Força C

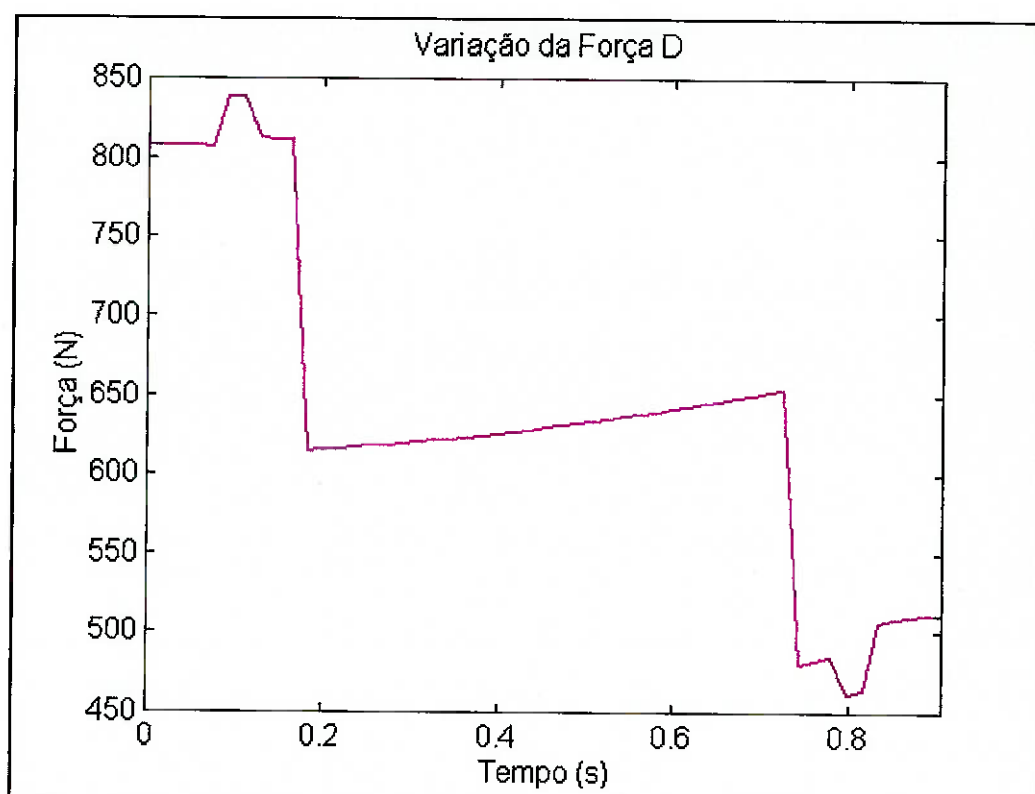


Fig. 8.11 : Gráfico da Força D

9. Programa Working Model

O programa comercial "Working Model 3D" é um simulador de movimento para projetos mecânicos usados na área educacional. Ele permite a simulação de modelos mecânicos, verificando a iteração de componentes, o estudo de posições, forças e torques.

Através do programa "AutoCAD" foi construído um modelo tridimensional da plataforma. Este modelo foi exportado ao "Working Model" e inserido atuadores lineares e outras características para que pudesse ser simulado.

A fim de que fosse possível uma comparação com os resultados obtidos no programa desenvolvido em "Matlab", os comprimentos dos atuadores resultantes da simulação no "Matlab" foram exportado para o WM. Essa integração foi possível através da aproximação dos comprimentos dos atuadores ao longo do tempo por polinômios de grau 6. Esse polinômios foram inseridos no Working Model como fórmulas relativas ao comprimentos dos atuadores em função do tempo. Os resultados obtidos são demonstrados, em conjunto com os do Matlab, a seguir.

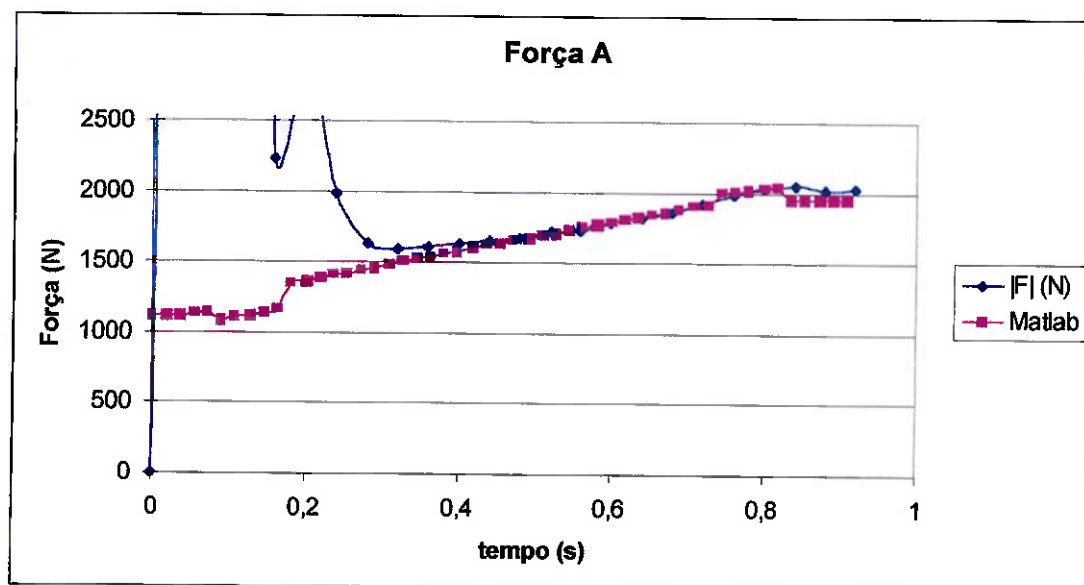


Fig. 9.1 : Gráfico de forças no atuador A

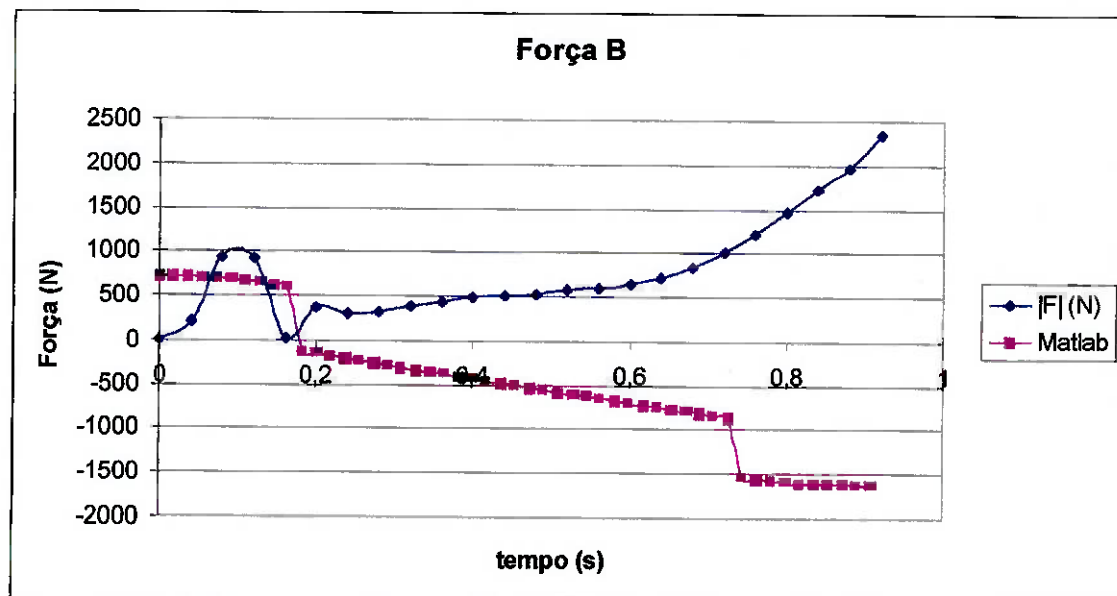


Fig. 9.2 : Gráfico de forças no atuador B

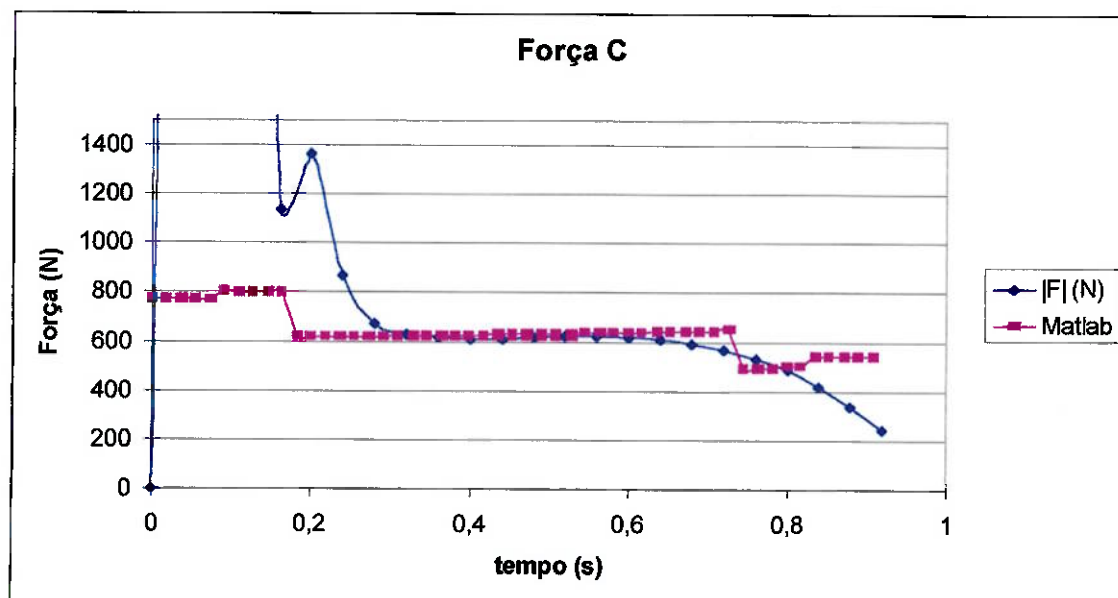


Fig. 9.3 : Gráfico de forças no atuador C

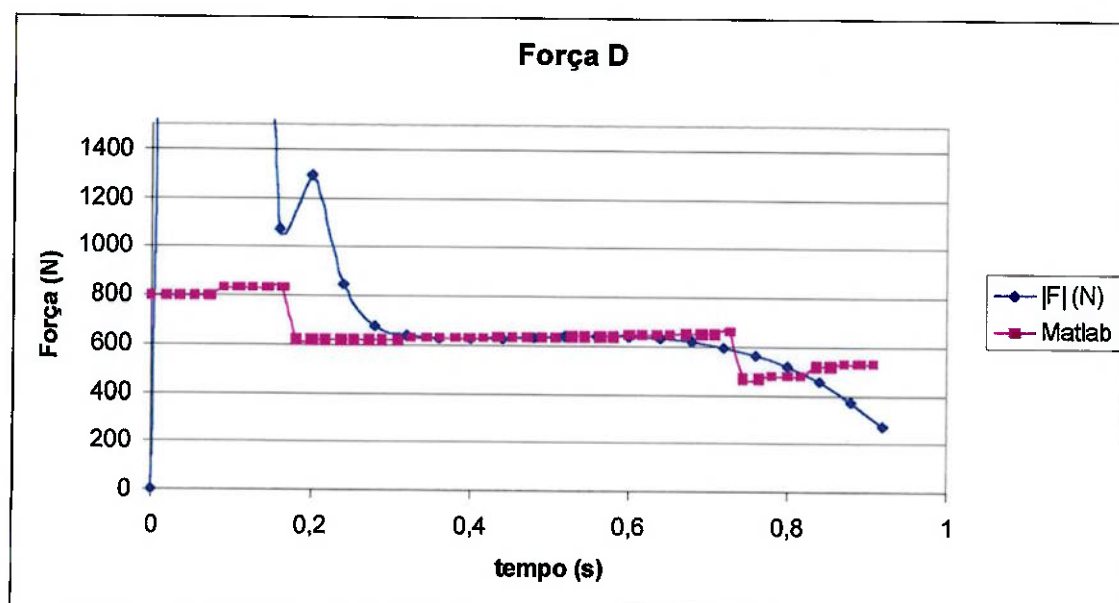


Fig. 9.4 : Gráfico de forças no atuador D

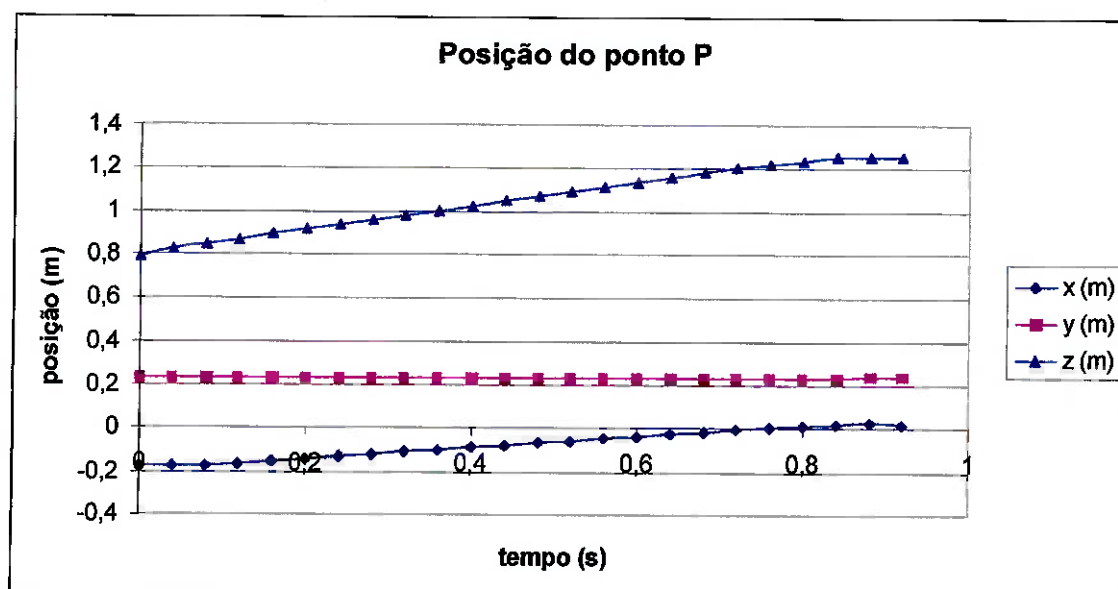


Fig. 9.5 : Gráfico da posição do ponto P

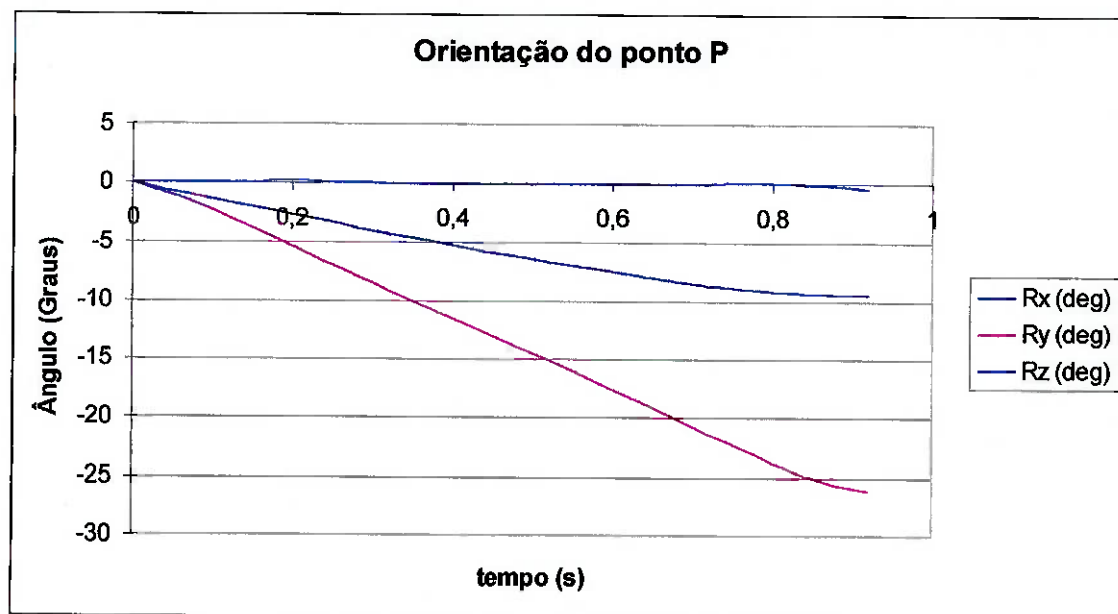


Fig. 9.6 : Orientação do ponto P

Como pode ser visto pelas figuras 9.5 e 9.6, a plataforma projetada no Working Model atingiu as posições desejadas, e através dos gráficos das forças percebe-se que depois de 0,3 segundo, os valores das forças do Working Model acompanha os valores do Matlab. Antes de 0,3 segundo, os valores das forças calculados pelo Working Model são completamente diferentes das do Matlab, e este fato pode ser explicado por dois motivos :

1. A plataforma projetado no Working Model, que está em uma posição diferente das condições iniciais do Matlab, ao iniciar o percurso da plataforma deve atingir a posição inicial, fornecida pelo Matlab, muito rápido, provocando um esforço muito grande nos atuadores e/ou
2. O Working Model considera que no tempo inicial na plataforma não possui nenhum esforço, isto pode ser visto porque todos os gráficos de força começam no valor nulo e isto não é verdade uma vez que a plataforma possui peso. Quando se tem o início do movimento, a ação do peso, em um curto espaço de tempo, age como se fosse um impacto e os atuadores devem suportá-lo.

10. Conclusões

10.1. O projeto

A indústria de entretenimento é um segmento de grande potencial econômico. A simulação de voo com a utilização de uma plataforma móvel é uma forma de entretenimento segura, ao mesmo tempo que consegue gerar emoção ao usuário. A viabilidade econômica e física do projeto são baseadas em equipamentos já existentes no mercado.

O estudo de plataformas móveis também pode ser utilizado para outras áreas, como os centros de usinagem. Trata-se de uma tecnologia inovadora, e pouco explorada cientificamente e economicamente.

10.2. O programa

O programa de computador desenvolvido admite 4 graus de liberdade, com os atuadores em disposição pré-determinada. Os pontos de fixação dos atuadores na base fixa, as dimensões da plataforma, e características de massa e momentos de inércia podem ser alterados pelo usuário do programa, conforme necessidade de aplicação.

O programa gera saídas gráficas que podem servir de poderosa ferramenta no projeto de uma plataforma. A trajetória, animada, permite uma visualização de como se comportará a plataforma ao longo do tempo. Os gráficos de forças permitem a seleção dos atuadores prismáticos do projeto.

A comparação dos resultados obtidos no programa "Matlab" com os obtidos na simulação do "Working Model" permite verificar a coerência dos mesmos.

O uso de valores de velocidade e aceleração de um projeto semelhante já implementado também gera saídas de acordo com o esperado.

10.3. Sugestões de continuidade do projeto

O trabalho dá todas as ferramentas necessárias para o dimensionamento dos componentes de uma plataforma. Torna-se viável a construção de um protótipo com mobilidade própria.

A continuidade no estudo do tema permite a evolução de uma plataforma com quatro graus de mobilidade para seis graus de mobilidade. Ficaria, assim, de acordo com plataformas de simulação para uso profissional.

11. Referências Bibliográficas

CALEFFI, T. Desenvolvimento de um programa para simuladores de vôo. São Carlos, 1995. 182 p. Dissertação (Mestrado) – Escola de Engenharia de São Carlos.

KOREN, Y. Robotics for engineers. New York, McGraw-Hill, 1985.

BEER, F. P.; JOHNSON, E. R. Mecânica vetorial para engenheiros - Dinâmica. 3.ed. Vol. II McGraw-Hill do Brasil, 1980.

COELHO, T. H. Otimização no balanceamento de mecanismo tri-dimensionais do tipo "RSSR". São Paulo, 1997. 197p. Tese (Doutorado) – Escola Politécnica, Universidade de São Paulo.

GIACAGLIA, G. E. O Mecânica geral. Rio de Janeiro : Campus, 1982.

Apêndice I

Programa feito no Matlab para a simulação da barra.

```
function simula2D(ato);

% Programa para a simulação da barra
% Objetivo : Calcular as forças nos atuadores ao longo da sua trajetória e
%            como saída tem-se :
%            - Trajetória do movimento
%            - Gráficos das forças nos atuadores
%            - Arquivo (resposta) com as forças nos atuadores ao longo do tempo

% Variáveis Globais das Figuras
global numerofig figbarra figtraj figforca4 figforca7 figforca2
global muda

if nargin < 1,      % Primeira vez que é executado o programa
    ato = 'inicio';
    muda = 0;
end;
if strcmp(ato,'inicio'),

    numerofig = figure( ...
        'Name','Dados de Entrada', ...
        'NumberTitle','off');

    %=====
    % MOLDULA DO CONSOLE

    frmPos = [0.08 0.08 0.85 0.85];
    h = uicontrol( ...
        'Style','frame', ...
        'Units','normalized', ...
        'Position',frmPos, ...
        'BackgroundColor',[0.5 0.5 0.5]);

% Definição das posições dos textos

% Coordenados dos Pontos fixos

% Ao =====

textpos = [0.127 0.8164 0.20 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz', 'left', ...
    'String','Coordenadas (m):', ...
    'Interruptible', 'no', ...
```

```

        'BackgroundColor',[0.5 0.5 0.5], ...
        'ForegroundColor','white');
textpos = [0.19 0.76 0.04 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','left', ...
    'String','Ao', ...
    'Interruptible','no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor','white');
textpos = [0.127 0.7 0.03 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','left', ...
    'String','X', ...
    'Interruptible','no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor','white');
textpos = [0.160 0.72 0.1 0.04];
callbackStr = 'simula2D("aox")';
Paox = uicontrol( ...
    'Style','edit', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','center', ...
    'Background','white', ...
    'Foreground','black', ...
    'String','10','Userdata',10, ...
    'Callback',callbackStr);
textpos = [0.127 0.65 0.03 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','left', ...
    'String','Y', ...
    'Interruptible','no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor','white');
textpos = [0.160 0.67 0.1 0.04];
callbackStr = 'simula2D("aoy")';
Paoy = uicontrol( ...
    'Style','edit', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','center', ...
    'Background','white', ...
    'Foreground','black', ...
    'String','0','Userdata',0, ...
    'Callback',callbackStr);

```

```
% Bo =====
```

```
textpos = [0.49 0.76 0.04 0.06];  
h = uicontrol( ...  
    'Style','text', ...  
    'Units','normalized', ...  
    'Position',textpos, ...  
    'Horiz','left', ...  
    'String','Bo', ...  
    'Interruptible','no', ...  
    'BackgroundColor',[0.5 0.5 0.5], ...  
    'ForegroundColor','white');  
textpos = [0.427 0.7 0.03 0.06];  
h = uicontrol( ...  
    'Style','text', ...  
    'Units','normalized', ...  
    'Position',textpos, ...  
    'Horiz','left', ...  
    'String','X', ...  
    'Interruptible','no', ...  
    'BackgroundColor',[0.5 0.5 0.5], ...  
    'ForegroundColor','white');  
textpos = [0.460 0.72 0.1 0.04];  
callbackStr = 'simula2D("box")';  
Pbox = uicontrol( ...  
    'Style','edit', ...  
    'Units','normalized', ...  
    'Position',textpos, ...  
    'Horiz','center', ...  
    'Background','white', ...  
    'Foreground','black', ...  
    'String','30','Userdata',30, ...  
    'Callback',callbackStr);
```

```
textpos = [0.427 0.65 0.03 0.06];  
h = uicontrol( ...  
    'Style','text', ...  
    'Units','normalized', ...  
    'Position',textpos, ...  
    'Horiz','left', ...  
    'String','Y', ...  
    'Interruptible','no', ...  
    'BackgroundColor',[0.5 0.5 0.5], ...  
    'ForegroundColor','white');  
textpos = [0.460 0.67 0.1 0.04];  
callbackStr = 'simula2D("boy")';  
Pboy = uicontrol( ...  
    'Style','edit', ...  
    'Units','normalized', ...  
    'Position',textpos, ...  
    'Horiz','center', ...  
    'Background','white', ...  
    'Foreground','black', ...
```

```

        'String','0','Userdata',0, ...
        'Callback',callbackStr);

% Co =====

textpos = [0.76 0.76 0.04 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','left', ...
    'String','Co', ...
    'Interruptible','no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor','white');
textpos = [0.7 0.7 0.03 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','left', ...
    'String','X', ...
    'Interruptible','no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor','white');
textpos = [0.740 0.72 0.1 0.04];
callbackStr = 'simula2D("cox")';
Pcox = uicontrol( ...
    'Style','edit', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','center', ...
    'Background','white', ...
    'Foreground','black', ...
    'String','0','Userdata',0, ...
    'Callback',callbackStr);

textpos = [0.7 0.65 0.03 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','left', ...
    'String','Y', ...
    'Interruptible','no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor','white');
textpos = [0.740 0.67 0.1 0.04];
callbackStr = 'simula2D("coy")';
Pcoy = uicontrol( ...
    'Style','edit', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','center', ...

```

```

        'Background','white', ...
        'Foreground','black', ...
        'String','10','Userdata',10, ...
        'Callback',callbackStr);

% Número de Interações =====

    textpos = [0.5 0.8164 0.27 0.06];
    h = uicontrol( ...
        'Style','text', ...
        'Units','normalized', ...
        'Position',textpos, ...
        'Horiz','left', ...
        'String','Número de Interações:', ...
        'Interruptible','no', ...
        'BackgroundColor',[0.5 0.5 0.5], ...
        'ForegroundColor','white');
    textpos = [0.77 0.83 0.05 0.04];
    callbackStr = 'simula2D("n")';
    Pn = uicontrol( ...
        'Style','edit', ...
        'Units','normalized', ...
        'Position',textpos, ...
        'Horiz','center', ...
        'Background','white', ...
        'Foreground','black', ...
        'String','30','Userdata',30, ...
        'Callback',callbackStr);

%=====
% COORDENADAS INICIAIS

% Xp =====

    textpos = [0.127 0.58 0.30 0.06];
    h = uicontrol( ...
        'Style','text', ...
        'Units','normalized', ...
        'Position',textpos, ...
        'Horiz','left', ...
        'String','Coordenadas Iniciais (m):', ...
        'Interruptible','no', ...
        'BackgroundColor',[0.5 0.5 0.5], ...
        'ForegroundColor','white');
    textpos = [0.127 0.53 0.04 0.06];
    h = uicontrol( ...
        'Style','text', ...
        'Units','normalized', ...
        'Position',textpos, ...
        'Horiz','left', ...
        'String','Xp', ...
        'Interruptible','no', ...
        'BackgroundColor',[0.5 0.5 0.5], ...

```

```

    'ForegroundColor', 'white');
textpos = [0.170 0.55 0.1 0.04];
callbackStr = 'simula2D("xpi")';
Pxpi = uicontrol( ...
    'Style','edit', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','center', ...
    'Background','white', ...
    'Foreground','black', ...
    'String','20','Userdata',20, ...
    'Callback',callbackStr);

```

% Yp =====

```

textpos = [0.427 0.53 0.04 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','left', ...
    'String','Yp', ...
    'Interruptible','no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor','white');
textpos = [0.47 0.55 0.1 0.04];
callbackStr = 'simula2D("ypi")';
Pypi = uicontrol( ...
    'Style','edit', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','center', ...
    'Background','white', ...
    'Foreground','black', ...
    'String','20','Userdata',20, ...
    'Callback',callbackStr);

```

% Theta =====

```

textpos = [0.7 0.53 0.07 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','left', ...
    'String','Theta', ...
    'Interruptible','no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor','white');
textpos = [0.775 0.55 0.07 0.04];
callbackStr = 'simula2D("theta")';
Ptheta = uicontrol( ...
    'Style','edit', ...

```

```

        'Units','normalized', ...
        'Position',textpos, ...
        'Horiz','center', ...
        'Background','white', ...
        'Foreground','black', ...
        'String','0','Userdata',0, ...
        'Callback',callbackStr);
textpos = [0.85 0.54 0.01 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','left', ...
    'String','o', ...
    'Interruptible','no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor','white');

%=====
% COORDENADAS FINAIS  Xp

textpos = [0.127 0.46 0.28 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','left', ...
    'String','Coordenadas Finais (m):', ...
    'Interruptible','no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor','white');
textpos = [0.127 0.41 0.04 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','left', ...
    'String','Xp', ...
    'Interruptible','no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor','white');
textpos = [0.170 0.43 0.1 0.04];
callbackStr = 'simula2D("xpf")';
Pxp = uicontrol( ...
    'Style','edit', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','center', ...
    'Background','white', ...
    'Foreground','black', ...
    'String','20','Userdata',20, ...
    'Callback',callbackStr);

```

```
% Yp =====
```

```
textpos = [0.427 0.41 0.04 0.06];  
h = uicontrol( ...  
    'Style','text', ...  
    'Units','normalized', ...  
    'Position',textpos, ...  
    'Horiz','left', ...  
    'String','Yp', ...  
    'Interruptible','no', ...  
    'BackgroundColor',[0.5 0.5 0.5], ...  
    'ForegroundColor','white');  
textpos = [0.47 0.43 0.1 0.04];  
callbackStr = 'simula2D("ypf")';  
Pypf = uicontrol( ...  
    'Style','edit', ...  
    'Units','normalized', ...  
    'Position',textpos, ...  
    'Horiz','center', ...  
    'Background','white', ...  
    'Foreground','black', ...  
    'String','40','Userdata',40, ...  
    'Callback',callbackStr);
```

```
% Theta =====
```

```
textpos = [0.7 0.41 0.07 0.06];  
h = uicontrol( ...  
    'Style','text', ...  
    'Units','normalized', ...  
    'Position',textpos, ...  
    'Horiz','left', ...  
    'String','Theta', ...  
    'Interruptible','no', ...  
    'BackgroundColor',[0.5 0.5 0.5], ...  
    'ForegroundColor','white');  
textpos = [0.85 0.42 0.01 0.06];  
h = uicontrol( ...  
    'Style','text', ...  
    'Units','normalized', ...  
    'Position',textpos, ...  
    'Horiz','left', ...  
    'String','°', ...  
    'Interruptible','no', ...  
    'BackgroundColor',[0.5 0.5 0.5], ...  
    'ForegroundColor','white');  
textpos = [0.775 0.43 0.07 0.04];  
callbackStr = 'simula2D("thetaf")';  
Pthetaf = uicontrol( ...  
    'Style','edit', ...  
    'Units','normalized', ...  
    'Position',textpos, ...  
    'Horiz','center', ...  
    'Background','white', ...
```

```

        'Foreground','black', ...
        'String','0','Userdata',0, ...
        'Callback',callbackStr);

%=====
% LIMITES MÁXIMOS

textpos = [0.127 0.33 0.23 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','left', ...
    'String','Limites Máximos :!', ...
    'Interruptible','no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor','white');

% Aceleração =====

textpos = [0.127 0.28 0.15 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','left', ...
    'String','Aceleração', ...
    'Interruptible','no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor','white');
textpos = [0.127 0.25 0.10 0.04];
callbackStr = 'simula2D("acel")';
Pacel = uicontrol( ...
    'Style','edit', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','center', ...
    'Background','white', ...
    'Foreground','black', ...
    'String','1','Userdata',1, ...
    'Callback',callbackStr);
textpos = [0.23 0.23 0.05 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','left', ...
    'String','m/s²', ...
    'Interruptible','no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor','white');

```

% Aceleração angular =====

```
textpos = [0.47 0.28 0.25 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','left', ...
    'String','Aceleração Angular', ...
    'Interruptible','no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor','white');
textpos = [0.47 0.25 0.1 0.04];
callbackStr = 'simula2D("alfa")';
Palfa = uicontrol( ...
    'Style','edit', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','center', ...
    'Background','white', ...
    'Foreground','black', ...
    'String','0.1','Userdata',.1, ...
    'Callback',callbackStr);
textpos = [0.55 0.23 0.07 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','left', ...
    'String','rad/s²', ...
    'Interruptible','no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor','white');
```

% Velocidade Angular =====

```
textpos = [0.7 0.28 0.2 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','left', ...
    'String','Velocidade Angular', ...
    'Interruptible','no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor','white');
textpos = [0.7 0.25 0.1 0.04];
callbackStr = 'simula2D("w")';
Pw = uicontrol( ...
    'Style','edit', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','center', ...
    'Background','white', ...
```

```

        'Foreground','black', ...
        'String','20','Userdata',20, ...
        'Callback',callbackStr);
textpos = [0.8 0.23 0.07 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','left', ...
    'String','rad/s²', ...
    'Interruptible','no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor','white');

% Velocidade =====

textpos = [0.31 0.28 0.15 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','left', ...
    'String','Velocidade', ...
    'Interruptible','no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor','white');
textpos = [0.31 0.25 0.1 0.04];
callbackStr = 'simula2D("vel")';
Pvel = uicontrol( ...
    'Style','edit', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','center', ...
    'Background','white', ...
    'Foreground','black', ...
    'String','0.1','Userdata',0.1, ...
    'Callback',callbackStr);
textpos = [0.41 0.23 0.05 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','left', ...
    'String','m/s', ...
    'Interruptible','no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor','white');

```

```
%=====
% FORÇAS ATUANTES
```

```
textpos = [0.127 0.17 0.21 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','left', ...
    'String','Forças Atuantes :', ...
    'Interruptible','no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor','white');
```

```
% Fx =====
```

```
textpos = [0.127 0.13 0.04 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','left', ...
    'String','Fx', ...
    'Interruptible','no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor','white');
textpos = [0.170 0.15 0.10 0.04];
callbackStr = 'simula2D("fx")';
Pfx = uicontrol( ...
    'Style','edit', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','center', ...
    'Background','white', ...
    'Foreground','black', ...
    'String','2','Userdata',2, ...
    'Callback',callbackStr);
textpos = [0.27 0.13 0.03 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','left', ...
    'String','N', ...
    'Interruptible','no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor','white');
```

```
% Fy =====
```

```
textpos = [0.427 0.13 0.04 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
```

```

        'Position',textpos, ...
        'Horiz', 'left', ...
        'String', 'fy', ...
        'Interruptible', 'no', ...
        'BackgroundColor',[0.5 0.5 0.5], ...
        'ForegroundColor', 'white');
textpos = [0.47 0.15 0.1 0.04];
callbackStr = 'simula2D("fy")';
Pfy = uicontrol( ...
        'Style','edit', ...
        'Units','normalized', ...
        'Position',textpos, ...
        'Horiz','center', ...
        'Background','white', ...
        'Foreground','black', ...
        'String','3','Userdata',3, ...
        'Callback',callbackStr);
textpos = [0.57 0.13 0.03 0.06];
h = uicontrol( ...
        'Style','text', ...
        'Units','normalized', ...
        'Position',textpos, ...
        'Horiz', 'left', ...
        'String','N', ...
        'Interruptible', 'no', ...
        'BackgroundColor',[0.5 0.5 0.5], ...
        'ForegroundColor', 'white');

```

```

%=====
% BOTÃO PARA INICIAR A SIMULAÇÃO

```

```

labelStr = 'Simular';
callbackStr = 'simula2D("simular")';
inicio = uicontrol(...
        'Style','pushbutton',...
        'Units','normalized',...
        'HorizontalAlignment','center',...
        'Position',[0.7 0.13 0.15 0.06],...
        'String',labelStr,...
        'Callback',callbackStr);

```

```

%=====
% BOTÃO PARA FINALIZAR O PROGRAMA

```

```

labelStr = 'Sair';
callbackStr = 'simula2D("sair")';
inicio = uicontrol(...
        'Style','pushbutton',...
        'Units','normalized',...
        'HorizontalAlignment','center',...
        'Position',[0.77 0.01 0.15 0.06],...
        'String',labelStr,...
        'Callback',callbackStr);

```

```
%=====
% BOTÃO PARA ALTERAR DE TELA ENTRE OS DADOS GERAIS E O DA BARRA
```

```
labelStr = 'Barra';
callbackStr = 'simula2D("TelaBarra")';
inicio = uicontrol(...
    'Style','pushbutton',...
    'Units','normalized',...
    'HorizontalAlignment','center',...
    'Position',[0.77 0.932 0.15 0.06],...
    'String',labelStr,...
    'Callback',callbackStr);
```

```
%=====
% Janela Para a Entrada de dados da Barra
%
```

```
figbarra = figure( ...
    'Name','Propriedades da Barra', ...
    'NumberTitle','off');
```

```
figure(figbarra);
```

```
%=====
% CONSOLE frame
```

```
frmPos = [0.1 0.1 0.8 0.8];
h = uicontrol( ...
    'Style','frame', ...
    'Units','normalized', ...
    'Position',frmPos, ...
    'BackgroundColor',[0.5 0.5 0.5]);
axis('off');
textpos = [0.35 0.9 0.35 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','left', ...
    'String','Dados Relativos à Barra', ...
    'Interruptible','no', ...
    'BackgroundColor',[0 0 0], ...
    'ForegroundColor','white');
```

```
%=====
% Comprimento da Barra
```

```
textpos = [0.127 0.70 0.27 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','left', ...
```

```

    'String', 'Comprimento da Barra :', ...
    'Interruptible', 'no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor', 'white');
textpos = [0.41 0.72 0.07 0.04];
callbackStr = 'simula2D("r3")';
Pr3 = uicontrol( ...
    'Style','edit', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','center', ...
    'Background','white', ...
    'Foreground','black', ...
    'String','20','Userdata',20, ...
    'Callback',callbackStr);
textpos = [0.48 0.7 0.03 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz', 'left', ...
    'String','m.', ...
    'Interruptible', 'no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor', 'white');

%=====
% Massa

textpos = [0.127 0.6 0.11 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz', 'left', ...
    'String', 'Massa :', ...
    'Interruptible', 'no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor', 'white');
textpos = [0.24 0.62 0.07 0.04];
callbackStr = 'simula2D("mass")';
Pmass = uicontrol( ...
    'Style','edit', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','center', ...
    'Background','white', ...
    'Foreground','black', ...
    'String','5','Userdata',5, ...
    'Callback',callbackStr);
textpos = [0.32 0.6 0.03 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...

```

```

        'Position',textpos, ...
        'Horiz', 'left', ...
        'String','kg', ...
        'Interruptible', 'no', ...
        'BackgroundColor',[0.5 0.5 0.5], ...
        'ForegroundColor', 'white');

%=====
% Momento de Inércia

textpos = [0.127 0.5 0.27 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz', 'left', ...
    'String', 'Momento de Inercia : ', ...
    'Interruptible', 'no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor', 'white');
textpos = [0.41 0.52 0.07 0.04];
callbackStr = 'simula2D("Jz")';
Pjz = uicontrol( ...
    'Style','edit', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','center', ...
    'Background','white', ...
    'Foreground','black', ...
    'String','27','Userdata',27, ...
    'Callback',callbackStr);
textpos = [0.48 0.5 0.06 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz', 'left', ...
    'String','kg.m²', ...
    'Interruptible', 'no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor', 'white');

%=====
% BOTÃO PARA INICIAR A SIMULAÇÃO

labelStr = 'Simular';
callbackStr = 'simula2D("simular")';
inicio = uicontrol(...
    'Style','pushbutton',...
    'Units','normalized',...
    'Position',[0.7 0.13 0.15 0.06],...
    'String',labelStr,...
    'Callback',callbackStr);

```

```
%=====
% BOTÃO PARA ALTERAR DE TELA ENTRE OS DADOS GERAIS E O DA BARRA
```

```
labelStr = 'Entrada';
callbackStr = 'simula2D("Telaini")';
inicio = uicontrol(...
    'Style','pushbutton',...
    'Units','normalized',...
    'HorizontalAlignment','center',...
    'Position',[0.75 0.9 0.15 0.06],...
    'String',labelStr,...
    'Callback',callbackStr);
```

```
Lista = [Paox Paoy Pbox Pboy Pcox Pcoy Pypi Pypj ...
    Ptheta Pxpf Pypf Pthetaf Pacel Pvel ...
    Pfx Pfy Pn Palfa Pw];
```

```
set(figure,...
    'Visible','off',...
    'UserData',Lista);
figure(figbarra);
ListBarra = [ Pr3 Pmass Pjz ];
set(figbarra,...
    'Visible','off',...
    'UserData',ListBarra);
set(gcf, 'Pointer','arrow');
drawnow
```

```
simula2D('simular');
figure(figure);
set(gcf, 'Pointer','arrow');
return
```

```
%=====
```

```
elseif strcmp(ato,'simular'),
```

```
%=====
```

```
% Recolhimento dos Dados Gerais
```

```
figure(figure);
Lista = get(gcf,'UserData');
```

```
Paox = Lista(1);
aox = get(Paox,'UserData');
Paoy = Lista(2);
aoy = get(Paoy,'UserData');
Pbox = Lista(3);
box = get(Pbox,'UserData');
Pboy = Lista(4);
boy = get(Pboy,'UserData');
Pcox = Lista(5);
cox = get(Pcox,'UserData');
Pcoy = Lista(6);
```

```

coy = get(Pcoy,'UserData');
Pxpi = Lista(7);
xpi = get(Pxpi,'UserData');
Pypi = Lista(8);
ypi = get(Pypi,'UserData');
Ptheta = Lista(9);
theta3i = get(Ptheta,'UserData');
Pxpfi = Lista(10);
xpfi = get(Pxpfi,'UserData');
Pypfi = Lista(11);
ypfi = get(Pypfi,'UserData');
Pthetafi = Lista(12);
theta3f = get(Pthetafi,'UserData');
Pacel = Lista(13);
amax = get(Pacel,'UserData');
Pvel = Lista(14);
vmax = get(Pvel,'UserData');
Pfx = Lista(15);
fx = get(Pfx,'UserData');
Pfy = Lista(16);
fy = get(Pfy,'UserData');
Pn = Lista(17);
n = get(Pn,'UserData');
Palfa = Lista(18);
alfamax = get(Palfa,'UserData');
Pw = Lista(19);
wmax = get(Pw,'UserData');
set(gcf,'nextplot','add');

```

```

%=====
% Recolhimento dos Dados Referentes à barra

```

```

figure(figbarra);
ListBarra = get(gcf,'UserData');

Pr3 = ListBarra(1);
r3 = get(Pr3,'UserData');
Pmass = ListBarra(2);
massa = get(Pmass,'UserData');
Pjz = ListBarra(3);
Jz = get(Pjz,'UserData');

set(gcf,'nextplot','add');

```

```

% Mudança dos ângulos para radianos
theta3i = (theta3i*pi)/180;
theta3f = (theta3f*pi)/180;

```

```

%=====
% Nome das figuras geradas durante a simulação

```

```

if muda == 0,
    figtraj = figure('Name','Trajetória',...

```

```

        'NumberTitle','off');
    end
    if muda == 1,
        figure(figtraj);
        clf;
    end
    if muda == 0,
        figforca4 = figure('Name','Força - 4',...
            'NumberTitle','off');
    end
    if muda == 0,
        figforca7 = figure('Name','Força - 7',...
            'NumberTitle','off');
    end
    if muda == 0,
        figforca2 = figure('Name','Força - 2',...
            'NumberTitle','off');
    end
    if muda == 0,
        muda = 1;
    end

%=====

cont = 1; % Variável auxiliar

r=sqrt((xpf-xpi)^2+(ypf-ypi)^2);
if sqrt(amax*r) > vmax
    t1 = vmax/amax;
    s = r - amax*t1^2;
    tttotal = 2*t1 + s/vmax;
    t2 = tttotal - t1;
else
    vmax = sqrt(amax*r);
    t1 = sqrt(r/amax);
    t2 = sqrt(r/amax);
    tttotal = 2*t1;
end

%=====
%      Calculo para o movimento rotacional

deltang = theta3f - theta3i;
sinalw = 1;
if theta3f == theta3i,
    alfamax = 0;
    tfinal = tttotal;
else
    sinalw = deltang/abs(deltang);
    if wmax < sqrt(abs(alfamax)*abs(deltang)),
        tw1 = abs(wmax)/abs(alfamax);
        twcte = (abs(deltang)-(wmax*wmax)/asb(alfamax))/abs(wmax) ;
        twtotal = 2*tw1 + twcte;
        tw2 = twtotal - tw1;
    end
end

```

```

else
    wmax = sqrt(abs(alfamax)*abs(deltang));
    tw1 = sqrt(abs(deltang)/abs(alfamax));
    tw2 = tw1;
    twtotal = 2*tw1;
end
if ttotal > twtotal ,
    tfinal = ttotal;
    wmax = wmax*(twtotal/ttotal);
    tw1 = tw1*(ttotal/twtotal);
    tw2 = tw2*(ttotal/twtotal);
    alfaamax = alfaamax*(twtotal/ttotal)^2;

else % twtotal > ttotal
    tfinal = twtotal;
    vmax = vmax*(ttotal/twtotal);
    t1 = t1*(twtotal/ttotal);
    t2 = t2*(twtotal/ttotal);
    amax = amax*(ttotal/twtotal)^2;
end
wmax = wmax*sinalw;
alfaamax = alfaamax*sinalw;
end
for t = 0 : tfinal/n : tfinal,

    xp = xpi + ((amax*t^2)/2)*(xpf-xpi)/r;
    yp = ypi + ((amax*t^2)/2)*(ypf-ypi)/r;
    aux1 = (amax*t1^2)/2;
    aux2 = (aux1+vmax*(t2-t1));
    if t >= t1
        aux1 = (amax*t1^2)/2;
        xp = xpi + (aux1+vmax*(t-t1))*(xpf-xpi)/r;
        yp = ypi + (aux1+vmax*(t-t1))*(ypf-ypi)/r;
    end
    if t >= t2
        aux2 = (aux1+vmax*(t2-t1));
        xp = xpi + (aux2+vmax*(t-t2)-(amax*(t-t2)^2)/2)*(xpf-xpi)/r;
        yp = ypi + (aux2+vmax*(t-t2)-(amax*(t-t2)^2)/2)*(ypf-ypi)/r;
    end
    theta3 = theta3i + ((alfaamax*t^2)/2);
    auxw1 = (alfaamax*tw1^2)/2;
    auxw2 = (auxw1+wmax*(tw2-tw1));
    if deltang ~= 0,
        if t >= tw1
            auxw1 = (alfaamax*tw1^2)/2;
            theta3 = theta3i + (auxw1+wmax*(t-tw1));
        end
        if t >= tw2
            if tw2 ~= tw1
                auxw2 = (auxw1+wmax*(tw2-tw1));
            else
                auxw2 = auxw1;
            end
            theta3 = theta3i + (auxw2+wmax*(t-tw2)-(alfaamax*(t-tw2)^2)/2);
        end
    end
end

```

```

        end
    end
    if ~isempty(theta3),
        r2x = xp - r3*cos(theta3)/2-aox;
        r2y = yp - r3*sin(theta3)/2-aoy;
        theta2 = atan2(r2y,r2x);
        r2 = sqrt(r2x^2+r2y^2);

        r4x = xp + r3*cos(theta3)/2-box;
        r4y = yp + r3*sin(theta3)/2-boy;
        theta4 = atan2(r4y,r4x);
        r4 = sqrt(r4x^2+r4y^2);

        r7x = xp-r3*cos(theta3)/2-cox;
        r7y = yp-r3*sin(theta3)/2-coy;
        theta7 = atan2(r7y,r7x);
        r7 = sqrt(r7x^2+r7y^2);

        if t <= t1,
            acel = amax;
        elseif t < t2 & t > t1,
            acel = 0;
        else
            acel = -amax;
        end
        if theta3f == theta3i,
            alfa1 = 0;
        else
            if t <= tw1,
                alfa1 = alfamax;
            elseif t > tw1 & t < tw2,
                alfa1 = 0;
            else
                alfa1 = -alfamax;
            end
        end
    end

%
=====
%      Resolução da equação para o cálculo das forças

    ax = xp - r3*cos(theta3)/2;
    ay = yp - r3*sin(theta3)/2;
    bx = xp + r3*cos(theta3)/2;
    by = yp + r3*sin(theta3)/2;
    B = [massa*acel*((ypf-ypi)/r) - fy + massa*10; massa*acel*((xpf-xpi)/r) - fx;
    Jz*alfa1];
    A = zeros(3);
    d11 = sin(theta2);
    d12 = sin(theta7);
    d13 = sin(theta4);
    d21 = cos(theta2);
    d22 = cos(theta7);
    d23 = cos(theta4);

```

```

d31 = sin(theta2)*(ax-xp)+cos(theta2)*(yp-ay);
d32 = sin(theta7)*(ax-xp)+cos(theta7)*(yp-ay);
d33 = sin(theta4)*(bx-xp)+cos(theta4)*(yp-by);
A=[ d11, d12, d13; d21, d22, d23; d31, d32, d33];

```

```

F = A\B;

```

```

tempo(cont) = t;
fo4(cont) = F(3);
fo7(cont) = F(2);
fo2(cont) = F(1);
cont = cont + 1;

```

```

%desenha a trajetória das barras
figure(figtraj)

```

```

a = [ aox ax ];
b = [ aoy ay ];
line (a,b,'color','g');
a = [ bx ax ];
b = [ by ay ];
line (a,b,'color','w');
a = [ bx box ];
b = [ by boy ];
line (a,b,'color','c');
c = [ cox ax ];
d = [ coy ay ];
line(c,d,'color','y');
axis('equal')

```

```

end % if theta3 ~isempty
end

```

```

save resposta2D fo4 fo2 fo7 tempo -ascii -tabs;

```

```

title('Trajetória do Mecanismo');

```

```

figure(figforca4)
plot(tempo,fo4,'c')
title('Variação da Força 4');
xlabel('Tempo (s)');
ylabel('Força (N)');
set(gca,'Xlim',[0 ttotall]);

```

```

figure(figforca7)
plot(tempo,fo7,'y')
title('Variação da Força 7');
xlabel('Tempo (s)');
ylabel('Força (N)');
set(gca,'Xlim',[0 ttotall]);

```

```

figure(figforca2)
plot(tempo,fo2,'g')
title('Variação da Força 2');

```

```

xlabel('Tempo (s)');
ylabel('Força (N)');
set(gca,'Xlim',[0 ttotal]);

```

```

figure(figure);
set(gcf, 'Pointer','arrow');
return

```

```

%=====

```

```

elseif strcmp(ato,'n'),
    v = get(gcf,'Userdata');
    s = get(gcf,'String');
    vv = eval(s,num2str(v));
    vv = round(vv);
    set(gcf,'Userdata',vv,'String',num2str(vv))
    muda = 1;
    return

```

```

%=====

```

```

elseif strcmp(ato,'aox'),
    v = get(gcf,'Userdata');
    s = get(gcf,'String');
    vv = eval(s,num2str(v));
    st = round(vv*100000)/100000;
    set(gcf,'Userdata',vv,'String',num2str(st))
    muda = 1;
    return

```

```

%=====

```

```

elseif strcmp(ato,'aoy'),
    v = get(gcf,'Userdata');
    s = get(gcf,'String');
    vv = eval(s,num2str(v));
    st = round(vv*100000)/100000;
    set(gcf,'Userdata',vv,'String',num2str(st))
    muda = 1;
    return

```

```

%=====

```

```

elseif strcmp(ato,'box'),
    v = get(gcf,'Userdata');
    s = get(gcf,'String');
    vv = eval(s,num2str(v));
    st = round(vv*100000)/100000;
    set(gcf,'Userdata',vv,'String',num2str(st))
    muda = 1;
    return

```

```

%=====

```

```

elseif strcmp(ato,'boy'),
    v = get(gco,'Userdata');
    s = get(gco,'String');
    vv = eval(s,num2str(v));
    st = round(vv*100000)/100000;
    set(gco,'Userdata',vv,'String',num2str(st))
    muda = 1;
    return

```

%=====

```

elseif strcmp(ato,'cox'),
    v = get(gco,'Userdata');
    s = get(gco,'String');
    vv = eval(s,num2str(v));
    st = round(vv*100000)/100000;
    set(gco,'Userdata',vv,'String',num2str(st))
    muda = 1;
    return

```

%=====

```

elseif strcmp(ato,'coy'),
    v = get(gco,'Userdata');
    s = get(gco,'String');
    vv = eval(s,num2str(v));
    st = round(vv*100000)/100000;
    set(gco,'Userdata',vv,'String',num2str(st))
    muda = 1;
    return

```

%=====

```

elseif strcmp(ato,'xpi'),
    v = get(gco,'Userdata');
    s = get(gco,'String');
    vv = eval(s,num2str(v));
    st = round(vv*100000)/100000;
    set(gco,'Userdata',vv,'String',num2str(st))
    muda = 1;
    return

```

%=====

```

elseif strcmp(ato,'ypi'),
    v = get(gco,'Userdata');
    s = get(gco,'String');
    vv = eval(s,num2str(v));
    st = round(vv*100000)/100000;
    set(gco,'Userdata',vv,'String',num2str(st))
    muda = 1;
    return

```

%=====

```
elseif strcmp(ato,'theta'),
    v = get(gco,'Userdata');
    s = get(gco,'String');
    vv = eval(s,num2str(v));
    vv = round(vv*100)/100;
    set(gco,'Userdata',vv,'String',num2str(vv))
    muda = 1;
    return
```

%=====

```
elseif strcmp(ato,'xpf'),
    v = get(gco,'Userdata');
    s = get(gco,'String');
    vv = eval(s,num2str(v));
    st = round(vv*100000)/100000;
    set(gco,'Userdata',vv,'String',num2str(st))
    muda = 1;
    return
```

%=====

```
elseif strcmp(ato,'ypf'),
    v = get(gco,'Userdata');
    s = get(gco,'String');
    vv = eval(s,num2str(v));
    st = round(vv*100000)/100000;
    set(gco,'Userdata',vv,'String',num2str(st))
    muda = 1;
    return
```

%=====

```
elseif strcmp(ato,'thetaf'),
    v = get(gco,'Userdata');
    s = get(gco,'String');
    vv = eval(s,num2str(v));
    vv = round(vv*100)/100;
    set(gco,'Userdata',vv,'String',num2str(vv))
    muda = 1;
    return
```

%=====

```
elseif strcmp(ato,'acel'),
    v = get(gco,'Userdata');
    s = get(gco,'String');
    vv = eval(s,num2str(v));
    st = round(vv*100000)/100000;
    set(gco,'Userdata',vv,'String',num2str(st))
    muda = 1;
    return
```

%=====

```
elseif strcmp(ato,'desa'),
    v = get(gco,'Userdata');
    s = get(gco,'String');
    vv = eval(s,num2str(v));
    st = round(vv*100000)/100000;
    set(gco,'Userdata',vv,'String',num2str(st))
    muda = 1;
    return
```

%=====

```
elseif strcmp(ato,'vel'),
    v = get(gco,'Userdata');
    s = get(gco,'String');
    vv = eval(s,num2str(v));
    st = round(vv*100000)/100000;
    set(gco,'Userdata',vv,'String',num2str(st))
    muda = 1;
    return
```

%=====

```
elseif strcmp(ato,'w'),
    v = get(gco,'Userdata');
    s = get(gco,'String');
    vv = eval(s,num2str(v));
    st = round(vv*100000)/100000;
    set(gco,'Userdata',vv,'String',num2str(st))
    muda = 1;
    return
```

%=====

```
elseif strcmp(ato,'fx'),
    v = get(gco,'Userdata');
    s = get(gco,'String');
    vv = eval(s,num2str(v));
    st = round(vv*100000)/100000;
    set(gco,'Userdata',vv,'String',num2str(st))
    muda = 1;
    return
```

%=====

```
elseif strcmp(ato,'fy'),
    v = get(gco,'Userdata');
    s = get(gco,'String');
    vv = eval(s,num2str(v));
    st = round(vv*100000)/100000;
    set(gco,'Userdata',vv,'String',num2str(st))
    muda = 1;
    return
```

%=====

```
elseif strcmp(ato,'r3'),
    v = get(gco,'Userdata');
    s = get(gco,'String');
    vv = eval(s,num2str(v));
    st = round(vv*100000)/100000;
    set(gco,'Userdata',vv,'String',num2str(st))
    muda = 1;
    return
```

%=====

```
elseif strcmp(ato,'alfa'),
    v = get(gco,'Userdata');
    s = get(gco,'String');
    vv = eval(s,num2str(v));
    st = round(vv*100000)/100000;
    set(gco,'Userdata',vv,'String',num2str(st))
    muda = 1;
    return
```

%=====

```
elseif strcmp(ato,'mass'),
    v = get(gco,'Userdata');
    s = get(gco,'String');
    vv = eval(s,num2str(v));
    st = round(vv*100000)/100000;
    set(gco,'Userdata',vv,'String',num2str(st))
    muda = 1;
    return
```

%=====

```
elseif strcmp(ato,'Jz'),
    v = get(gco,'Userdata');
    s = get(gco,'String');
    vv = eval(s,num2str(v));
    st = round(vv*100000)/100000;
    set(gco,'Userdata',vv,'String',num2str(st))
    muda = 1;
    return
```

%=====

```
elseif strcmp(ato,'TelaBarra'),
    figure(figbarra);
    return
```

%=====

```
elseif strcmp(ato,'Telaini'),
```

```
    figure(nerofig);
    return

%=====

elseif strcmp(ato,'sair'),
    figure(figforca2);
    close(gcf);
    figure(figforca4);
    close(gcf);
    figure(figforca7);
    close(gcf);
    figure(figtraj);
    close(gcf);
    figure(figbarra);
    close(gcf);
    figure(nerofig);
    close(gcf);
    clear;

end
```

Apêndice II

Programa feito no Matlab para a simulação da plataforma

```
function simula3D(ato);

% Programa para a simulação da plataforma
% Objetivo : Calcular as forças nos atuadores ao longo da sua trajetória e
%            como saída tem-se :
%            - Trajetória do movimento
%            - Gráficos das forças nos atuadores
%            - Arquivo (compr) com os comprimentos de cada atuador
%            - Arquivo (forca) com as forças nos atuadores ao longo do tempo

% Variáveis Globais das Figuras
global numerofig figbarra figtraj figforcaa figforcab figforcac figforcad
global muda

if nargin < 1,      % Primeira vez que é executado o programa
    ato = 'inicio';
    muda = 0;
end;
if strcmp(ato,'inicio'),

    numerofig = figure( ...
        'Name','Dados de Entrada', ...
        'NumberTitle','off');

    %=====
    % MOLDULA DO CONSOLE

    frmPos = [0.08 0.08 0.85 0.85];
    h = uicontrol( ...
        'Style','frame', ...
        'Units','normalized', ...
        'Position',frmPos, ...
        'BackgroundColor',[0.5 0.5 0.5]);

    % Definição das posições dos textos

    % Coordenados dos Pontos fixos

    % Ao =====

    textpos = [0.127 0.85 0.20 0.06];
    h = uicontrol( ...
        'Style','text', ...
        'Units','normalized', ...
        'Position',textpos, ...
        'Horiz', 'left', ...
        'String','Coordenadas (m):', ...
```

```

        'Interruptible', 'no', ...
        'BackgroundColor',[0.5 0.5 0.5], ...
        'ForegroundColor', 'white');
textpos = [0.19 0.79 0.04 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz', 'left', ...
    'String', 'Ao', ...
    'Interruptible', 'no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor', 'white');
textpos = [0.127 0.75 0.03 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz', 'left', ...
    'String', 'X', ...
    'Interruptible', 'no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor', 'white');
textpos = [0.160 0.77 0.1 0.04];
callbackStr = 'simula3D("aox")';
Paox = uicontrol( ...
    'Style','edit', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','center', ...
    'Background','white', ...
    'Foreground','black', ...
    'String','0','Userdata',0, ...
    'Callback',callbackStr);

textpos = [0.127 0.70 0.03 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz', 'left', ...
    'String', 'Y', ...
    'Interruptible', 'no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor', 'white');
textpos = [0.160 0.72 0.1 0.04];
callbackStr = 'simula3D("aoy")';
Paoy = uicontrol( ...
    'Style','edit', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','center', ...
    'Background','white', ...
    'Foreground','black', ...

```

```

        'String','0','Userdata',0, ...
        'Callback',callbackStr);
textpos = [0.127 0.65 0.03 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','left', ...
    'String','Z', ...
    'Interruptible','no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor','white');
textpos = [0.160 0.67 0.1 0.04];
callbackStr = 'simula3D("aoz")';
Paoz = uicontrol( ...
    'Style','edit', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','center', ...
    'Background','white', ...
    'Foreground','black', ...
    'String','0','Userdata',0, ...
    'Callback',callbackStr);

```

% Bo =====

```

textpos = [0.38 0.79 0.04 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','left', ...
    'String','Bo', ...
    'Interruptible','no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor','white');

textpos = [0.35 0.77 0.1 0.04];
callbackStr = 'simula3D("box")';
Pbox = uicontrol( ...
    'Style','edit', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','center', ...
    'Background','white', ...
    'Foreground','black', ...
    'String','-0.555','Userdata',-0.555, ...
    'Callback',callbackStr);

```

```

textpos = [0.35 0.72 0.1 0.04];
callbackStr = 'simula3D("boy")';
Pboy = uicontrol( ...
    'Style','edit', ...

```

```

        'Units','normalized', ...
        'Position',textpos, ...
        'Horiz','center', ...
        'Background','white', ...
        'Foreground','black', ...
        'String','0','Userdata',0, ...
        'Callback',callbackStr);

textpos = [0.35 0.67 0.1 0.04];
callbackStr = 'simula3D("boz")';
Pboz = uicontrol( ...
    'Style','edit', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','center', ...
    'Background','white', ...
    'Foreground','black', ...
    'String','0','Userdata',0, ...
    'Callback',callbackStr);

% Co =====

textpos = [0.59 0.79 0.04 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','left', ...
    'String','Co', ...
    'Interruptible','no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor','white');

textpos = [0.55 0.77 0.1 0.04];
callbackStr = 'simula3D("cox")';
Pcox = uicontrol( ...
    'Style','edit', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','center', ...
    'Background','white', ...
    'Foreground','black', ...
    'String','0.45','Userdata',0.45, ...
    'Callback',callbackStr);

textpos = [0.55 0.72 0.1 0.04];
callbackStr = 'simula3D("coy")';
Pcoy = uicontrol( ...
    'Style','edit', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','center', ...
    'Background','white', ...

```

```

        'Foreground','black', ...
        'String','0.225','Userdata',0.225, ...
        'Callback',callbackStr);

textpos = [0.55 0.67 0.1 0.04];
callbackStr = 'simula3D("coz")';
Pcoz = uicontrol( ...
    'Style','edit', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','center', ...
    'Background','white', ...
    'Foreground','black', ...
    'String','0','Userdata',0, ...
    'Callback',callbackStr);

% Do =====

textpos = [0.79 0.79 0.04 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','left', ...
    'String','Do', ...
    'Interruptible','no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor','white');

textpos = [0.75 0.77 0.1 0.04];
callbackStr = 'simula3D("dox")';
Pdox = uicontrol( ...
    'Style','edit', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','center', ...
    'Background','white', ...
    'Foreground','black', ...
    'String','0.45','Userdata',0.45, ...
    'Callback',callbackStr);

textpos = [0.75 0.72 0.1 0.04];
callbackStr = 'simula3D("doy")';
Pdoy = uicontrol( ...
    'Style','edit', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','center', ...
    'Background','white', ...
    'Foreground','black', ...
    'String','-0.225','Userdata',-0.225, ...
    'Callback',callbackStr);

```

```

textpos = [0.75 0.67 0.1 0.04];
callbackStr = 'simula3D("doz")';
Pdoz = uicontrol( ...
    'Style','edit', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','center', ...
    'Background','white', ...
    'Foreground','black', ...
    'String','0','Userdata',0, ...
    'Callback',callbackStr);

```

% Número de Interações =====

```

textpos = [0.5 0.85 0.27 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','left', ...
    'String','Número de Interações :', ...
    'Interruptible','no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor','white');
textpos = [0.77 0.87 0.05 0.04];
callbackStr = 'simula3D("n")';
Pn = uicontrol( ...
    'Style','edit', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','center', ...
    'Background','white', ...
    'Foreground','black', ...
    'String','50','Userdata',50, ...
    'Callback',callbackStr);

```

%=====

% COORDENADAS INICIAIS

% Xp =====

```

textpos = [0.127 0.58 0.30 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','left', ...
    'String','Coordenadas Iniciais (m):', ...
    'Interruptible','no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor','white');
textpos = [0.127 0.53 0.04 0.06];

```

```

h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz', 'left', ...
    'String', 'Xp', ...
    'Interruptible', 'no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor', 'white');
textpos = [0.167 0.55 0.1 0.04];
callbackStr = 'simula3D("xpi")';
Pxpi = uicontrol( ...
    'Style','edit', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','center', ...
    'Background','white', ...
    'Foreground','black', ...
    'String','0.225','Userdata',0.225, ...
    'Callback',callbackStr);

% Yp =====

textpos = [0.28 0.53 0.04 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz', 'left', ...
    'String', 'Yp', ...
    'Interruptible', 'no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor', 'white');
textpos = [0.32 0.55 0.1 0.04];
callbackStr = 'simula3D("ypi")';
Pypi = uicontrol( ...
    'Style','edit', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','center', ...
    'Background','white', ...
    'Foreground','black', ...
    'String','0','Userdata',0, ...
    'Callback',callbackStr);

% Zp =====

textpos = [0.44 0.53 0.04 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz', 'left', ...
    'String', 'Zp', ...

```

```

    'Interruptible', 'no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor', 'white');
textpos = [0.48 0.55 0.1 0.04];
callbackStr = 'simula3D("zpi")';
Pzpi = uicontrol( ...
    'Style','edit', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','center', ...
    'Background','white', ...
    'Foreground','black', ...
    'String','0.8','Userdata',0.8, ...
    'Callback',callbackStr);

```

% Theta =====

```

textpos = [0.60 0.53 0.09 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz', 'left', ...
    'String', 'Theta(°)', ...
    'Interruptible', 'no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor', 'white');
textpos = [0.695 0.53 0.02 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz', 'left', ...
    'String', 'X', ...
    'Interruptible', 'no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor', 'white');

```

```

textpos = [0.715 0.55 0.07 0.04];
callbackStr = 'simula3D("thetax")';
Pthetax = uicontrol( ...
    'Style','edit', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','center', ...
    'Background','white', ...
    'Foreground','black', ...
    'String','0','Userdata',0, ...
    'Callback',callbackStr);

```

```

textpos = [0.80 0.53 0.02 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...

```

```

'Position',textpos, ...
'Horiz', 'left', ...
'String', 'Y', ...
'Interruptible', 'no', ...
'BackgroundColor',[0.5 0.5 0.5], ...
'ForegroundColor', 'white');

textpos = [0.825 0.55 0.07 0.04];
callbackStr = 'simula3D("thetay")';
Pthetay = uicontrol( ...
    'Style','edit', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','center', ...
    'Background','white', ...
    'Foreground','black', ...
    'String','0','Userdata',0, ...
    'Callback',callbackStr);

%=====
% COORDENADAS FINAIS  Xp

textpos = [0.127 0.46 0.28 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz', 'left', ...
    'String','Coordenadas Finais (m):', ...
    'Interruptible', 'no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor', 'white');
textpos = [0.127 0.41 0.04 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz', 'left', ...
    'String', 'Xp', ...
    'Interruptible', 'no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor', 'white');
textpos = [0.167 0.43 0.1 0.04];
callbackStr = 'simula3D("xpf")';
Pxpf = uicontrol( ...
    'Style','edit', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','center', ...
    'Background','white', ...
    'Foreground','black', ...
    'String','0.245','Userdata',0.245, ...
    'Callback',callbackStr);

```

% Yp =====

```
textpos = [0.28 0.41 0.04 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz', 'left', ...
    'String', 'Yp', ...
    'Interruptible', 'no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor', 'white');
textpos = [0.32 0.43 0.1 0.04];
callbackStr = 'simula3D("ypf")';
Pypf = uicontrol( ...
    'Style','edit', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','center', ...
    'Background','white', ...
    'Foreground','black', ...
    'String','0','Userdata',0, ...
    'Callback',callbackStr);
```

% Zp =====

```
textpos = [0.44 0.41 0.04 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz', 'left', ...
    'String', 'Zp', ...
    'Interruptible', 'no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor', 'white');
textpos = [0.48 0.43 0.1 0.04];
callbackStr = 'simula3D("zpf")';
Pzpf = uicontrol( ...
    'Style','edit', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','center', ...
    'Background','white', ...
    'Foreground','black', ...
    'String','0.9','Userdata',0.9, ...
    'Callback',callbackStr);
```

% Theta =====

```
textpos = [0.60 0.41 0.09 0.06];
h = uicontrol( ...
```

```

'Style','text', ...
'Units','normalized', ...
'Position',textpos, ...
'Horiz','left', ...
'String','Theta(°)', ...
'Interruptible','no', ...
'BackgroundColor',[0.5 0.5 0.5], ...
'ForegroundColor','white');
textpos = [0.695 0.41 0.02 0.06];
h = uicontrol( ...
'Style','text', ...
'Units','normalized', ...
'Position',textpos, ...
'Horiz','left', ...
'String','X', ...
'Interruptible','no', ...
'BackgroundColor',[0.5 0.5 0.5], ...
'ForegroundColor','white');
textpos = [0.715 0.43 0.07 0.04];
callbackStr = 'simula3D("thetafx")';
Pthetafx = uicontrol( ...
'Style','edit', ...
'Units','normalized', ...
'Position',textpos, ...
'Horiz','center', ...
'Background','white', ...
'Foreground','black', ...
'String','30','Userdata',30, ...
'Callback',callbackStr);
textpos = [0.80 0.41 0.02 0.06];
h = uicontrol( ...
'Style','text', ...
'Units','normalized', ...
'Position',textpos, ...
'Horiz','left', ...
'String','Y', ...
'Interruptible','no', ...
'BackgroundColor',[0.5 0.5 0.5], ...
'ForegroundColor','white');
textpos = [0.825 0.43 0.07 0.04];
callbackStr = 'simula3D("thetafy")';
Pthetafy = uicontrol( ...
'Style','edit', ...
'Units','normalized', ...
'Position',textpos, ...
'Horiz','center', ...
'Background','white', ...
'Foreground','black', ...
'String','20','Userdata',20, ...
'Callback',callbackStr);

```

```
%=====
```

```
% LIMITES MÁXIMOS
```

```
textpos = [0.127 0.33 0.23 0.06];  
h = uicontrol( ...  
    'Style','text', ...  
    'Units','normalized', ...  
    'Position',textpos, ...  
    'Horiz', 'left', ...  
    'String','Limites Máximos :', ...  
    'Interruptible', 'no', ...  
    'BackgroundColor',[0.5 0.5 0.5], ...  
    'ForegroundColor', 'white');
```

```
% Aceleração =====
```

```
textpos = [0.127 0.28 0.15 0.06];  
h = uicontrol( ...  
    'Style','text', ...  
    'Units','normalized', ...  
    'Position',textpos, ...  
    'Horiz', 'left', ...  
    'String', 'Aceleração', ...  
    'Interruptible', 'no', ...  
    'BackgroundColor',[0.5 0.5 0.5], ...  
    'ForegroundColor', 'white');  
textpos = [0.127 0.25 0.10 0.04];  
callbackStr = 'simula3D("acel")';  
Pacel = uicontrol( ...  
    'Style','edit', ...  
    'Units','normalized', ...  
    'Position',textpos, ...  
    'Horiz','center', ...  
    'Background','white', ...  
    'Foreground','black', ...  
    'String','10','Userdata',10, ...  
    'Callback',callbackStr);  
textpos = [0.23 0.23 0.05 0.06];  
h = uicontrol( ...  
    'Style','text', ...  
    'Units','normalized', ...  
    'Position',textpos, ...  
    'Horiz', 'left', ...  
    'String','m/s²', ...  
    'Interruptible', 'no', ...  
    'BackgroundColor',[0.5 0.5 0.5], ...  
    'ForegroundColor', 'white');
```

```
% Aceleração angular =====
```

```
textpos = [0.35 0.28 0.25 0.06];  
h = uicontrol( ...  
    'Style','text', ...  
    'Units','normalized', ...  
    'Position',textpos, ...  
    'Horiz', 'left', ...  
    'String', 'Aceleração Angular', ...  
    'Interruptible', 'no', ...  
    'BackgroundColor',[0.5 0.5 0.5], ...  
    'ForegroundColor', 'white');
```

```
textpos = [0.35 0.23 0.02 0.06];  
h = uicontrol( ...  
    'Style','text', ...  
    'Units','normalized', ...  
    'Position',textpos, ...  
    'Horiz', 'left', ...  
    'String', 'X', ...  
    'Interruptible', 'no', ...  
    'BackgroundColor',[0.5 0.5 0.5], ...  
    'ForegroundColor', 'white');
```

```
textpos = [0.38 0.25 0.1 0.04];  
callbackStr = 'simula3D("alfax")';  
Palfax = uicontrol( ...  
    'Style','edit', ...  
    'Units','normalized', ...  
    'Position',textpos, ...  
    'Horiz','center', ...  
    'Background','white', ...  
    'Foreground','black', ...  
    'String','6.981','Userdata',6.981, ...  
    'Callback',callbackStr);
```

```
textpos = [0.48 0.23 0.07 0.06];  
h = uicontrol( ...  
    'Style','text', ...  
    'Units','normalized', ...  
    'Position',textpos, ...  
    'Horiz', 'left', ...  
    'String','rad/s²', ...  
    'Interruptible', 'no', ...  
    'BackgroundColor',[0.5 0.5 0.5], ...  
    'ForegroundColor', 'white');
```

```
textpos = [0.57 0.23 0.02 0.06];  
h = uicontrol( ...  
    'Style','text', ...  
    'Units','normalized', ...  
    'Position',textpos, ...  
    'Horiz', 'left', ...  
    'String', 'Y', ...  
    'Interruptible', 'no', ...  
    'BackgroundColor',[0.5 0.5 0.5], ...
```

```

'ForegroundColor', 'white');
textpos = [0.59 0.25 0.1 0.04];
callbackStr = 'simula3D("alfay")';
Palfay = uicontrol( ...
    'Style','edit', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','center', ...
    'Background','white', ...
    'Foreground','black', ...
    'String','6.981','Userdata',6.981, ...
    'Callback',callbackStr);
textpos = [0.69 0.23 0.07 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','left', ...
    'String','rad/s²', ...
    'Interruptible','no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor','white');

```

% Velocidade Angular =====

```

textpos = [0.35 0.17 0.2 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','left', ...
    'String','Velocidade Angular', ...
    'Interruptible','no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor','white');
textpos = [0.35 0.11 0.02 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','left', ...
    'String','X', ...
    'Interruptible','no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor','white');

```

```

textpos = [0.38 0.13 0.1 0.04];
callbackStr = 'simula3D("wx")';
Pwx = uicontrol( ...
    'Style','edit', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','center', ...

```

```

        'Background','white', ...
        'Foreground','black', ...
        'String','0.524','Userdata',0.524, ...
        'Callback',callbackStr);
textpos = [0.48 0.11 0.07 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz', 'left', ...
    'String','rad/s', ...
    'Interruptible', 'no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor', 'white');
textpos = [0.57 0.11 0.02 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz', 'left', ...
    'String', 'Y', ...
    'Interruptible', 'no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor', 'white');

textpos = [0.59 0.13 0.1 0.04];
callbackStr = 'simula3D("wy")';
Pwy = uicontrol( ...
    'Style','edit', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','center', ...
    'Background','white', ...
    'Foreground','black', ...
    'String','0.524','Userdata',0.524, ...
    'Callback',callbackStr);
textpos = [0.69 0.11 0.07 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz', 'left', ...
    'String','rad/s', ...
    'Interruptible', 'no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor', 'white');

```

```
% Velocidade =====
```

```
textpos = [0.127 0.17 0.15 0.06];  
h = uicontrol( ...  
    'Style','text', ...  
    'Units','normalized', ...  
    'Position',textpos, ...  
    'Horiz','left', ...  
    'String','Velocidade', ...  
    'Interruptible','no', ...  
    'BackgroundColor',[0.5 0.5 0.5], ...  
    'ForegroundColor','white');  
textpos = [0.127 0.13 0.1 0.04];  
callbackStr = 'simula3D("vel")';  
Pvel = uicontrol( ...  
    'Style','edit', ...  
    'Units','normalized', ...  
    'Position',textpos, ...  
    'Horiz','center', ...  
    'Background','white', ...  
    'Foreground','black', ...  
    'String','1','Userdata',1, ...  
    'Callback',callbackStr);  
textpos = [0.23 0.11 0.05 0.06];  
h = uicontrol( ...  
    'Style','text', ...  
    'Units','normalized', ...  
    'Position',textpos, ...  
    'Horiz','left', ...  
    'String','m/s', ...  
    'Interruptible','no', ...  
    'BackgroundColor',[0.5 0.5 0.5], ...  
    'ForegroundColor','white');
```

```
%=====
```

```
% BOTÃO PARA INICIAR A SIMULAÇÃO
```

```
labelStr = 'Simular';  
callbackStr = 'simula3D("simular")';  
inicio = uicontrol(...  
    'Style','pushbutton',...  
    'Units','normalized',...  
    'HorizontalAlignment','center',...  
    'Position',[0.77 0.135 0.15 0.06],...  
    'String',labelStr,...  
    'Callback',callbackStr);
```

```
%=====
```

```
% BOTÃO PARA FINALIZAR O PROGRAMA
```

```
labelStr = 'Sair';  
callbackStr = 'simula3D("sair")';  
inicio = uicontrol(...
```

```

'Style','pushbutton',...
'Units','normalized',...
'HorizontalAlignment','center',...
'Position',[0.77 0.007 0.15 0.06],...
'String',labelStr,...
'Callback',callbackStr);

```

```

%=====
% BOTÃO PARA ALTERAR DE TELA ENTRE OS DADOS GERAIS E O DA BARRA

```

```

labelStr = 'Plataforma';
callbackStr = 'simula3D("TelaBarra")';
inicio = uicontrol(...
    'Style','pushbutton',...
    'Units','normalized',...
    'HorizontalAlignment','center',...
    'Position',[0.77 0.932 0.15 0.06],...
    'String',labelStr,...
    'Callback',callbackStr);

```

```

%=====
% Janela Para a Entrada de dados da Barra
%

```

```

figbarra = figure( ...
    'Name','Propriedades da Plataforma', ...
    'NumberTitle','off');

figure(figbarra);

```

```

%=====
% CONSOLE frame

```

```

frmPos = [0.1 0.1 0.8 0.8];
h = uicontrol( ...
    'Style','frame', ...
    'Units','normalized', ...
    'Position',frmPos, ...
    'BackgroundColor',[0.5 0.5 0.5]);
axis('off');
textpos = [0.35 0.9 0.35 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz', 'left', ...
    'String', 'Dados Relativos à Plataforma', ...
    'Interruptible', 'no', ...
    'BackgroundColor',[0 0 0], ...
    'ForegroundColor', 'white');

```

```
%=====
```

```
% Dimensões da Plataforma
```

```
textpos = [0.127 0.70 0.3 0.06];  
h = uicontrol( ...  
    'Style','text', ...  
    'Units','normalized', ...  
    'Position',textpos, ...  
    'Horiz','left', ...  
    'String','Dimensões da Plataforma :', ...  
    'Interruptible','no', ...  
    'BackgroundColor',[0.5 0.5 0.5], ...  
    'ForegroundColor','white');
```

```
textpos = [0.42 0.7 0.03 0.06];  
h = uicontrol( ...  
    'Style','text', ...  
    'Units','normalized', ...  
    'Position',textpos, ...  
    'Horiz','left', ...  
    'String','X', ...  
    'Interruptible','no', ...  
    'BackgroundColor',[0.5 0.5 0.5], ...  
    'ForegroundColor','white');
```

```
textpos = [0.45 0.72 0.07 0.04];  
callbackStr = 'simula3D("r3x")';  
Pr3x = uicontrol( ...  
    'Style','edit', ...  
    'Units','normalized', ...  
    'Position',textpos, ...  
    'Horiz','center', ...  
    'Background','white', ...  
    'Foreground','black', ...  
    'String','0.45','Userdata',0.45, ...  
    'Callback',callbackStr);
```

```
textpos = [0.52 0.7 0.03 0.06];  
h = uicontrol( ...  
    'Style','text', ...  
    'Units','normalized', ...  
    'Position',textpos, ...  
    'Horiz','left', ...  
    'String','m.', ...  
    'Interruptible','no', ...  
    'BackgroundColor',[0.5 0.5 0.5], ...  
    'ForegroundColor','white');
```

```
textpos = [0.62 0.7 0.03 0.06];  
h = uicontrol( ...  
    'Style','text', ...  
    'Units','normalized', ...  
    'Position',textpos, ...
```

```

'Horiz', 'left', ...
'String', 'Y', ...
'Interruptible', 'no', ...
'BackgroundColor', [0.5 0.5 0.5], ...
'ForegroundColor', 'white');

textpos = [0.65 0.72 0.07 0.04];
callbackStr = 'simula3D("r3y")';
Pr3y = uicontrol( ...
    'Style', 'edit', ...
    'Units', 'normalized', ...
    'Position', textpos, ...
    'Horiz', 'center', ...
    'Background', 'white', ...
    'Foreground', 'black', ...
    'String', '0.45', 'Userdata', 0.45, ...
    'Callback', callbackStr);

```

```

textpos = [0.72 0.7 0.03 0.06];
h = uicontrol( ...
    'Style', 'text', ...
    'Units', 'normalized', ...
    'Position', textpos, ...
    'Horiz', 'left', ...
    'String', 'm.', ...
    'Interruptible', 'no', ...
    'BackgroundColor', [0.5 0.5 0.5], ...
    'ForegroundColor', 'white');

```

```

%=====

```

```

% Massa

```

```

textpos = [0.127 0.6 0.11 0.06];
h = uicontrol( ...
    'Style', 'text', ...
    'Units', 'normalized', ...
    'Position', textpos, ...
    'Horiz', 'left', ...
    'String', 'Massa :', ...
    'Interruptible', 'no', ...
    'BackgroundColor', [0.5 0.5 0.5], ...
    'ForegroundColor', 'white');
textpos = [0.24 0.62 0.07 0.04];
callbackStr = 'simula3D("mass")';
Pmass = uicontrol( ...
    'Style', 'edit', ...
    'Units', 'normalized', ...
    'Position', textpos, ...
    'Horiz', 'center', ...
    'Background', 'white', ...
    'Foreground', 'black', ...
    'String', '250', 'Userdata', 250, ...
    'Callback', callbackStr);

```

```

textpos = [0.31 0.6 0.03 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz', 'left', ...
    'String','kg', ...
    'Interruptible', 'no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor', 'white');

%=====
% Momento de Inércia

textpos = [0.127 0.5 0.27 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz', 'left', ...
    'String', 'Momento de Inercia :', ...
    'Interruptible', 'no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor', 'white');

textpos = [0.42 0.5 0.03 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz', 'left', ...
    'String','Jx', ...
    'Interruptible', 'no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor', 'white');

textpos = [0.45 0.52 0.07 0.04];
callbackStr = 'simula3D("Jx")';
Pjx = uicontrol( ...
    'Style','edit', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','center', ...
    'Background','white', ...
    'Foreground','black', ...
    'String','4.23','Userdata',4.23, ...
    'Callback',callbackStr);
textpos = [0.52 0.5 0.06 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz', 'left', ...
    'String','kg.m²', ...

```

```

        'Interruptible', 'no', ...
        'BackgroundColor',[0.5 0.5 0.5], ...
        'ForegroundColor', 'white');

textpos = [0.42 0.45 0.03 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz', 'left', ...
    'String','Jz', ...
    'Interruptible', 'no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor', 'white');

textpos = [0.45 0.47 0.07 0.04];
callbackStr = 'simula3D("Jz")';
Pjz = uicontrol( ...
    'Style','edit', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','center', ...
    'Background','white', ...
    'Foreground','black', ...
    'String','8.46','Userdata',8.46, ...
    'Callback',callbackStr);
textpos = [0.52 0.45 0.06 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz', 'left', ...
    'String','kg.m²', ...
    'Interruptible', 'no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor', 'white');

textpos = [0.62 0.5 0.03 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz', 'left', ...
    'String','Jy', ...
    'Interruptible', 'no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor', 'white');

textpos = [0.65 0.52 0.07 0.04];
callbackStr = 'simula3D("Jy")';
Pjy = uicontrol( ...
    'Style','edit', ...
    'Units','normalized', ...
    'Position',textpos, ...

```

```

        'Horiz','center', ...
        'Background','white', ...
        'Foreground','black', ...
        'String','4.23','Userdata',4.23, ...
        'Callback',callbackStr);
textpos = [0.72 0.5 0.06 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz', 'left', ...
    'String','kg.m²', ...
    'Interruptible','no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor','white');

%=====
% FORÇAS ATUANTES

textpos = [0.127 0.38 0.21 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz', 'left', ...
    'String','Forças Atuantes :', ...
    'Interruptible','no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor','white');

% Fx =====

textpos = [0.127 0.34 0.04 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz', 'left', ...
    'String','Fx', ...
    'Interruptible','no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor','white');
textpos = [0.170 0.36 0.10 0.04];
callbackStr = 'simula3D("fx")';
Pfx = uicontrol( ...
    'Style','edit', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','center', ...
    'Background','white', ...
    'Foreground','black', ...
    'String','0','Userdata',0, ...
    'Callback',callbackStr);
textpos = [0.27 0.34 0.03 0.06];

```

```

h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz', 'left', ...
    'String','N', ...
    'Interruptible', 'no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor', 'white');

% Fz =====

textpos = [0.35 0.34 0.04 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz', 'left', ...
    'String', 'Fz', ...
    'Interruptible', 'no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor', 'white');
textpos = [0.39 0.36 0.1 0.04];
callbackStr = 'simula3D("fz")';
Pfz = uicontrol( ...
    'Style','edit', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz','center', ...
    'Background','white', ...
    'Foreground','black', ...
    'String','0','Userdata',0, ...
    'Callback',callbackStr);
textpos = [0.49 0.34 0.03 0.06];
h = uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position',textpos, ...
    'Horiz', 'left', ...
    'String','N', ...
    'Interruptible', 'no', ...
    'BackgroundColor',[0.5 0.5 0.5], ...
    'ForegroundColor', 'white');

%=====
% BOTÃO PARA INICIAR A SIMULAÇÃO

labelStr = 'Simular';
callbackStr = 'simula3D("simular")';
inicio = uicontrol(...
    'Style','pushbutton',...
    'Units','normalized',...
    'Position',[0.7 0.13 0.15 0.06],...
    'String',labelStr,...

```

```

        'Callback',callbackStr);

%=====
% BOTÃO PARA ALTERAR DE TELA ENTRE OS DADOS GERAIS E O DA BARRA

    labelStr = 'Entrada';
    callbackStr = 'simula3D("Telaini")';
    inicio = uicontrol(...
        'Style','pushbutton',...
        'Units','normalized',...
        'HorizontalAlignment','center',...
        'Position',[0.75 0.9 0.15 0.06],...
        'String',labelStr,...
        'Callback',callbackStr);

    Lista = [Paox Paoy Paoz Pbox Pboy Pboz Pcox Pcoy Pcoz Pdox Pdoz ...
        Pn Pxp Pyp Pzpi Pthetax Pthetay Pxpfp Pypf Pzpf Pthetafx Pthetafy ...
        Pacel Palfax Palfay Pwx Pwy Pvel];
    set(figure,...
        'Visible','off',...
        'UserData',Lista);
    figure(figbarra);
    ListBarra = [ Pr3x Pr3y Pmass Pjx Pjy Pfx Pfz Pjz];
    set(figbarra,...
        'Visible','off',...
        'UserData',ListBarra);
    set(gcf, 'Pointer','arrow');
    drawnow;

    simula3D('similar');
    figure(figure);
    set(gcf, 'Pointer','arrow');
    return

%=====

elseif strcmp(ato,'similar'),

%=====
% Recolhimento dos Dados Gerais

    figure(figure);
    Lista = get(gcf,'UserData');

    Paox = Lista(1);
    aox = get(Paox,'UserData');
    Paoy = Lista(2);
    aoy = get(Paoy,'UserData');
    Paoz = Lista(3);
    aoz = get(Paoz,'UserData');
    Pbox = Lista(4);
    box = get(Pbox,'UserData');
    Pboy = Lista(5);

```

```
boy = get(Pboy,'UserData');
Pboz = Lista(6);
boz = get(Pboz,'UserData');
Pcox = Lista(7);
cox = get(Pcox,'UserData');
Pcoy = Lista(8);
coy = get(Pcoy,'UserData');
Pcoz = Lista(9);
coz = get(Pcoz,'UserData');
Pdox = Lista(10);
dox = get(Pdox,'UserData');
Pdoy = Lista(11);
doy = get(Pdoy,'UserData');
Pdoz = Lista(12);
doz = get(Pdoz,'UserData');
Pn = Lista(13);
n = get(Pn,'UserData');
Pxpi = Lista(14);
xpi = get(Pxpi,'UserData');
Pypi = Lista(15);
ypi = get(Pypi,'UserData');
Pzpi = Lista(16);
zpi = get(Pzpi,'UserData');
Pthetax = Lista(17);
thetaxi = get(Pthetax,'UserData');
Pthetay = Lista(18);
thetayi = get(Pthetay,'UserData');
Pxpf = Lista(19);
xpf = get(Pxpf,'UserData');
Pypf = Lista(20);
ypf = get(Pypf,'UserData');
Pzpf = Lista(21);
zpf = get(Pzpf,'UserData');
Pthetafx = Lista(22);
thetafx = get(Pthetafx,'UserData');
Pthetafy = Lista(23);
thetafy = get(Pthetafy,'UserData');
Pacel = Lista(24);
amax = get(Pacel,'UserData');
Palfax = Lista(25);
alfaxmax = get(Palfax,'UserData');
Palfay = Lista(26);
alfamax = get(Palfay,'UserData');
Pwx = Lista(27);
wxmax = get(Pwx,'UserData');
Pwy = Lista(28);
wmax = get(Pwy,'UserData');
Pvel = Lista(29);
vmax = get(Pvel,'UserData');
set(gcf,'nextplot','add');
```

```

%=====
% Recolhimento dos Dados Referentes à plataforma

figure(figbarra);
ListBarra = get(gcf,'UserData');

Pr3x = ListBarra(1);
r3 = get(Pr3x,'UserData');
Pr3y = ListBarra(2);
ry = get(Pr3y,'UserData');
Pmass = ListBarra(3);
massa = get(Pmass,'UserData');
Pjx = ListBarra(4);
Jx = get(Pjx,'UserData');
Pjy = ListBarra(5);
Jy = get(Pjy,'UserData');
Pfx = ListBarra(6);
fx = get(Pfx,'UserData');
Pfy = ListBarra(7);
fy = get(Pfy,'UserData');
set(gcf,'nextplot','add');
Pjz = ListBarra(8);
Jz = get(Pjz,'UserData');
set(gcf,'nextplot','add');

yp = aoy;

% Mudança dos ângulos para radianos
% Os valores são negativos pq nos calculos das fórmulas
% foram considerados os eixos opostos ao real.

thetaxi = -(thetaxi*pi)/180;
thetafx = -(thetafx*pi)/180;
thetayi = -(thetayi*pi)/180;
thetafy = -(thetafy*pi)/180;

%=====
% Nome das figuras geradas durante a simulação

if muda == 0,
    figtraj = figure('Name','Trajetória',...
        'NumberTitle','off');
end
if muda == 1,
    figure(figtraj);
    clf;
end
if muda == 0,
    figforcaa = figure('Name','Força - A',...
        'NumberTitle','off');
end
if muda == 0,

```

```

        figforcab = figure('Name','Força - B',...
                           'NumberTitle','off');
    end
    if muda == 0,
        figforcac = figure('Name','Força - C',...
                           'NumberTitle','off');
    end
    if muda == 0,
        figforcad = figure('Name','Força - D',...
                           'NumberTitle','off');
    end
    if muda == 0, % indica que não é a primeira vez
        muda = 1;
    end
    %=====

    cont = 1; % Variável auxiliar

    % por hipótese o valor de y não muda
    r=sqrt((xpf-xpi)^2+(zpf-zpi)^2);
    if r == 0,
        r = 1;
    end
    if sqrt(amax*r) > vmax
        t1 = vmax/amax;
        s = r - amax*t1^2;
        tttotal = 2*t1 + s/vmax;
        t2 = tttotal - t1;
    else
        vmax = sqrt(amax*r);
        t1 = sqrt(r/amax);
        t2 = sqrt(r/amax);
        tttotal = 2*t1;
    end

    %=====
    %      Calculo para o movimento rotacional
    % Para Y

    deltang = thetapy - thetai;
    sinalw = 1;
    if thetapy == thetai,
        alfamax = 0;
        tfinal = tttotal;
        tw1 = 0;
        tw2 = 0;
        twtotal = 0;
    else
        sinalw = deltang/abs(deltang);
        if wmax < sqrt(abs(alfamax)*abs(deltang)),
            tw1 = abs(wmax)/abs(alfamax);
            twcte = (abs(deltang)-(wmax*wmax)/abs(alfamax))/abs(wmax) ;
            twtotal = 2*tw1 + twcte;
            tw2 = twtotal - tw1;
        end
    end

```

```

else
    wmax = sqrt(abs(alfamax)*abs(deltang));
    tw1 = sqrt(abs(deltang)/abs(alfamax));
    tw2 = tw1;
    twtotal = 2*tw1;
end
end % thetai <> thetafi

% Para X
deltangx = thetafx - thetaxi;
sinalwx = 1;
if thetafx == thetaxi,
    alfaxmax = 0;
    tfinal = tttotal;
    twx1 = 0;
    twx2 = 0;
    twxtotal = 0;
else
    sinalwx = deltangx/abs(deltangx);
    if wxmax < sqrt(abs(alfamax)*abs(deltangx)),
        twx1 = abs(wxmax)/abs(alfamax);
        twxcte = (abs(deltangx)-(wxmax*wxmax)/abs(alfamax))/abs(wxmax) ;
        twxtotal = 2*twx1 + twxcte;
        twx2 = twxtotal - twx1;
    else
        wxmax = sqrt(abs(alfamax)*abs(deltangx));
        twx1 = sqrt(abs(deltangx)/abs(alfamax));
        twx2 = twx1;
        twxtotal = 2*twx1;
    end
end % thetaxi <> thetafx

%=====
% Calculo dos novos valores

tempocomp = [tttotal twtotal twxtotal];
tfinal = max(tempocomp);
vmax = vmax*(tttotal/tfinal);
wmax = wmax*(twtotal/tfinal);
wxmax = wxmax*(twxtotal/tfinal);
t1 = t1*(tfinal/tttotal);
if thetafi == thetai,
    tw1 = 0;
else
    tw1 = tw1*(tfinal/twtotal);
end
if thetafx == thetaxi,
    twx1 = 0;
else
    twx1 = twx1*(tfinal/twxtotal);
end
t2 = tfinal - t1;
if thetafi == thetai,
    tw2 = 0;

```

```

else
    tw2 = tw2*(tfinal/twtotal);
end
if thetax == thetaxi,
    twx2 = 0;
else
    twx2 = twx2*(tfinal/twxtotal);
end

amax = amax*(ttotal/tfinal)^2;
alfamax = alfamax*(twtotal/tfinal)^2;
alfaxmax = alfaxmax*(twxtotal/tfinal)^2;

%
=====
wmax = wmax*sinalw;
alfamax = alfamax*sinalw;
wxmax = wxmax*sinalwx;
alfaxmax = alfaxmax*sinalwx;

for t = 0 : tfinal/n : tfinal,
    xp = xpi + ((amax*t^2)/2)*(xpf-xpi)/r;
    zp = zpi + ((amax*t^2)/2)*(zpf-zpi)/r;
    if t >= t1
        aux1 = (amax*t1^2)/2;
        xp = xpi + (aux1+vmax*(t-t1))*(xpf-xpi)/r;
        zp = zpi + (aux1+vmax*(t-t1))*(zpf-zpi)/r;
    end
    if t >= t2
        aux2 = (aux1+vmax*(t2-t1));
        xp = xpi + (aux2+vmax*(t-t2)-(amax*(t-t2)^2)/2)*(xpf-xpi)/r;
        zp = zpi + (aux2+vmax*(t-t2)-(amax*(t-t2)^2)/2)*(zpf-zpi)/r;
    end
    thetay = thetai + ((alfamax*t^2)/2);
    if thetaxy == thetai,
        tw1 = 0;
    else
        if t >= tw1
            auxw1 = (alfamax*tw1^2)/2;
            thetay = thetai + (auxw1+wmax*(t-tw1));
        end
        if t >= tw2
            if tw2 ~= tw1
                auxw2 = (auxw1+wmax*(tw2-tw1));
            else
                auxw2 = auxw1;
            end
            thetay = thetai + (auxw2+wmax*(t-tw2)-(alfamax*(t-tw2)^2)/2);
        end
    end
    thetax = thetaxi + ((alfaxmax*t^2)/2);
    auxwx1 = (alfaxmax*twx1^2)/2;
    auxwx2 = (auxwx1+wxmax*(twx2-twx1));

```

```

    if thetafx == thetaxi,
        twx1 = 0;
    else
        if t >= twx1
            auxwx1 = (alfamax*twx1^2)/2;
            thetax = thetaxi + (auxwx1+wxmax*(t-twx1));
        end
        if t >= twx2
            if twx2 == twx1
                auxwx2 = auxwx1;
            else
                auxwx2 = (auxwx1+wxmax*(twx2-twx1));
            end
            thetax = thetaxi + (auxwx2+wxmax*(t-twx2)-(alfamax*(t-twx2)^2)/2);
        end
    end
end
%=====
if ~isempty(thetay),

    rax = xp - r3*cos(thetay)/2-aox;
    ax = rax + aox;
    ray = yp - aoy;
    ay = ray + aoy;
    raz = zp - r3*sin(thetay)/2-aoz;
    az = raz + aoz;
    thetaa = atan2(raz,rax);
    ra = sqrt(rax^2+ray^2+raz^2);

    rbx = xp-r3*cos(thetay)/2-box;
    bx = rbx + box;
    rby = yp - boy;
    by = rby + boy;
    rbz = zp-r3*sin(thetay)/2-boz;
    bz = rbz + boz;
    thetab = atan2(rbz,rbx);
    rb = sqrt(rbx^2+rby^2+rbz^2);

    ex = xp + r3*cos(thetay)/2;
    ey = yp;
    ez = zp + r3*sin(thetay)/2;

    hx = xp - r3*cos(thetay)/2;
    hy = yp;
    hz = zp - r3*sin(thetay)/2;

    dx = ex - ry*sin(thetax)*sin(thetay)/2;
    dy = ey - ry*cos(thetax)/2;
    dz = ez + ry*sin(thetax)*cos(thetay)/2;

    cx = ex + ry*sin(thetax)*sin(thetay)/2;
    cy = ey + ry*cos(thetax)/2;
    cz = ez - ry*sin(thetax)*cos(thetay)/2;

    fx = hx - ry*sin(thetax)*sin(thetay)/2;

```

```
fy = hy - ry*cos(thetax)/2;  
fz = hz + ry*sin(thetax)*cos(thetay)/2;
```

```
gx = hx + ry*sin(thetax)*sin(thetay)/2;  
gy = hy + ry*cos(thetax)/2;  
gz = hz - ry*sin(thetax)*cos(thetay)/2;
```

```
rcx = cx - cox;  
rcy = cy - coy;  
rcz = cz - coz;  
thetacy = atan2(rcz,rcx);  
thetacz = atan2(rcx,rcy);  
rc = sqrt(rcx^2+rcy^2+rcz^2);
```

```
rdx = dx - dox;  
rdy = dy - doy;  
rdz = dz - doz;  
thetady = atan2(rdz,rdx);  
thetadz = atan2(rdx,rdy);  
rd = sqrt(rdx^2+rdy^2+rdz^2);
```

```
%=====
```

```
% Versores do sistema fixo na plataforma
```

```
lxx = cos(thetay);  
lyy = 0;  
lzz = sin(thetay);
```

```
Jxx = -sin(thetay)*sin(thetax);  
Jyy = cos(thetax);  
Jzz = cos(thetay)*sin(thetax);
```

```
Kxx = -sin(thetay)*cos(thetax);  
Kyy = -sin(thetax);  
Kzz = cos(thetay)*cos(thetax);
```

```
if t <= t1,  
    acel = amax;  
elseif t < t2 & t > t1,  
    acel = 0;  
else  
    acel = -amax;  
end  
wgy = 0;  
if thetapy == thetayi,  
    alfa1 = 0;  
else  
    if t <= tw1,  
        alfa1 = alfamax;  
    elseif t > tw1 & t < tw2,  
        alfa1 = 0;  
    else
```

```

        alfa1 = -alfamax;
    end
end
if thetapy == thetayi,
    alfa1 = 0;
else
    if t <= tw1,
        wgy = alfa1*t;
    elseif t > tw1 & t < tw2,
        wgy = wmax;
    else
        wgy = wmax + alfa1*(t-tw2);
    end
end
End
    alfax1 = 0;
if thetax == thetaxi,
    alfax1 = 0;
else
    if t <= twx1,
        alfax1 = alfaxmax;
    elseif t > twx1 & t < twx2,
        alfax1 = 0;
    else
        alfax1 = -alfamax;
    end
end
    wxd = 0;
if thetax == thetaxi,
    alfax1 = 0;
else
    if t <= twx1,
        wxd = alfax1*t;
    elseif t > twx1 & t < twx2,
        wxd = wxmax;
    else
        wxd = wxmax + alfax1*(t-twx2);
    end
end

% =====
% Resolução das equações para o cálculo das forças

wix = wxd + wgy*Iyy;
wiy = wgy*Iyy;
wiz = wgy*Kyy;
alfax = alfax1 + alfa1*Iyy;
alfay = alfa1*Iyy;
alfaz = alfa1*Kyy;
B = [massa*acel*((zpf-zpi)/r)-fy+massa*9.81;massa*acel*((xpf-xpi)/r)-
fx; -massa*alfax+(Jz-Jy)*wiy*wiz;Jy*alfay+(Jx-Jz)*wix*wiz];
A = zeros(4);
d11 = raz/ra;
d12 = rbz/rb;
d13 = rcz/rc;

```

```

d14 = rdz/rd;
d21 = rax/ra;
d22 = rbx/rb;
d23 = rcx/rc;
d24 = rdx/rd;
d31 = det([lxx,lyy,lzz;ax-xp,ay-yp,az-zp;rax/ra,ray/ra,raz/ra]);
d32 = det([lxx,lyy,lzz;bx-xp,by-yp,bz-zp;rbx/rb,rby/rb,rbz/rb]);

d33 = det([lxx,lyy,lzz;cx-xp,cy-yp,cz-zp;rcx/rc,rcy/rc,rcz/rc]);
d34 = det([lxx,lyy,lzz;dx-xp,dy-yp,dz-zp;rdx/rd,rdy/rd,rdz/rd]);
d41 = det([Jxx,Jyy,Jzz;ax-xp,ay-yp,az-zp;rax/ra,ray/ra,raz/ra]);
d42 = det([Jxx,Jyy,Jzz;bx-xp,by-yp,bz-zp;rbx/rb,rby/rb,rbz/rb]);
d43 = det([Jxx,Jyy,Jzz;cx-xp,cy-yp,cz-zp;rcx/rc,rcy/rc,rcz/rc]);
d44 = det([Jxx,Jyy,Jzz;dx-xp,dy-yp,dz-zp;rdx/rd,rdy/rd,rdz/rd]);
A = [ d11, d12, d13, d14; d21, d22, d23, d24; d31, d32, d33, d34; d41, d42, d43,
d44];

```

```

F = A\B;
tempo(cont) = t;
fod(cont) = F(4);
foc(cont) = F(3);
fob(cont) = F(2);
foa(cont) = F(1);

```

```

cpa(cont) = ra;
cpb(cont) = rb;
cpc(cont) = rc;
cpd(cont) = rd;

```

```

cont = cont + 1;

```

```

%desenha a trajetória das barras

```

```

pause(0.3);      % espera 0,3 segundos antes de continuar
figure(figtraj);
a = [ aox ax ];
b = [ aoy ay ];
c = [ aoz az ];
d = [ box ax ];
e = [ boy ay ];
f = [ boz az ];
g = [ cox cx ];
h = [ coy cy ];
i = [ coz cz ];
j = [ dox dx ];
k = [ doy dy ];
l = [ doz dz ];
a1 = [ dx cx ];
b1 = [ dy cy ];
c1 = [ dz cz ];
d1 = [ gx cx ];
e1 = [ gy cy ];
f1 = [ gz cz ];
g1 = [ gx fx ];

```

```

h1 = [ gy fy ];
i1 = [ gz fz ];
j1 = [ fx dx ];
k1 = [ fy dy ];
l1 = [ fz dz ];

plot3(a,b,c,'y',d,e,f,'g',g,h,i,'r',j,k,l,'m',a1,b1,c1,'w',d1,e1,f1,'w',g1,h1,i1,'w',j1,k1,l1,'w');
axis('equal');
end % if theta3 ~isempty
end

save compr cpa cpb cpc cpd tempo tfinal angulox anguloy aceangx aceangy acele -
ascii ;
save forca foa fob foc fod tempo -ascii ;

title('Trajetória do Mecanismo');
axis('equal');
figure(figforcaa);
plot(tempo,foa,'y');
title('Variação da Força A');
xlabel('Tempo (s)');
ylabel('Força (N)');
set(gca,'Xlim',[0 tfinal]);

figure(figforcab);
plot(tempo,fob,'g');
title('Variação da Força B');
xlabel('Tempo (s)');
ylabel('Força (N)');
set(gca,'Xlim',[0 tfinal]);

figure(figforcac);
plot(tempo,foc,'r');
title('Variação da Força C');
xlabel('Tempo (s)');
ylabel('Força (N)');
set(gca,'Xlim',[0 tfinal]);

figure(figforcad);
plot(tempo,fod,'m');
title('Variação da Força D');
xlabel('Tempo (s)');
ylabel('Força (N)');
set(gca,'Xlim',[0 tfinal]);

figure(figure);
set(gcf, 'Pointer','arrow');
return

%=====

elseif strcmp(ato,'n'),
v = get(gcf,'UserData');
s = get(gcf,'String');

```

```

vv = eval(s,num2str(v));
vv = round(vv);
set(gco,'Userdata',vv,'String',num2str(vv))
muda = 1;
return

%=====
elseif strcmp(ato,'aox'),
    v = get(gco,'Userdata');
    s = get(gco,'String');
    vv = eval(s,num2str(v));
    st = round(vv*100000)/100000;
    set(gco,'Userdata',vv,'String',num2str(st))
    muda = 1;
    return

%=====
elseif strcmp(ato,'aoy'),
    v = get(gco,'Userdata');
    s = get(gco,'String');
    vv = eval(s,num2str(v));
    st = round(vv*100000)/100000;
    set(gco,'Userdata',vv,'String',num2str(st))
    muda = 1;
    return

%=====
elseif strcmp(ato,'aoz'),
    v = get(gco,'Userdata');
    s = get(gco,'String');
    vv = eval(s,num2str(v));
    st = round(vv*100000)/100000;
    set(gco,'Userdata',vv,'String',num2str(st))
    muda = 1;
    return

%=====
elseif strcmp(ato,'box'),
    v = get(gco,'Userdata');
    s = get(gco,'String');
    vv = eval(s,num2str(v));
    st = round(vv*100000)/100000;
    set(gco,'Userdata',vv,'String',num2str(st))
    muda = 1;
    return

%=====
elseif strcmp(ato,'boy'),
    v = get(gco,'Userdata');
    s = get(gco,'String');
    vv = eval(s,num2str(v));
    st = round(vv*100000)/100000;
    set(gco,'Userdata',vv,'String',num2str(st))
    muda = 1;

```

```

return

%=====
elseif strcmp(ato,'boz'),
    v = get(gco,'Userdata');
    s = get(gco,'String');
    vv = eval(s,num2str(v));
    st = round(vv*100000)/100000;
    set(gco,'Userdata',vv,'String',num2str(st))
    muda = 1;
    return

%=====
elseif strcmp(ato,'cox'),
    v = get(gco,'Userdata');
    s = get(gco,'String');
    vv = eval(s,num2str(v));
    st = round(vv*100000)/100000;
    set(gco,'Userdata',vv,'String',num2str(st))
    muda = 1;
    return

%=====
elseif strcmp(ato,'coy'),
    v = get(gco,'Userdata');
    s = get(gco,'String');
    vv = eval(s,num2str(v));
    st = round(vv*100000)/100000;
    set(gco,'Userdata',vv,'String',num2str(st))
    muda = 1;
    return

%=====
elseif strcmp(ato,'coz'),
    v = get(gco,'Userdata');
    s = get(gco,'String');
    vv = eval(s,num2str(v));
    st = round(vv*100000)/100000;
    set(gco,'Userdata',vv,'String',num2str(st))
    muda = 1;
    return

%=====
elseif strcmp(ato,'dox'),
    v = get(gco,'Userdata');
    s = get(gco,'String');
    vv = eval(s,num2str(v));
    st = round(vv*100000)/100000;
    set(gco,'Userdata',vv,'String',num2str(st))
    muda = 1;
    return

%=====
elseif strcmp(ato,'doy'),

```

```

v = get(gco,'Userdata');
s = get(gco,'String');
vv = eval(s,num2str(v));
st = round(vv*100000)/100000;
set(gco,'Userdata',vv,'String',num2str(st))
muda = 1;
return

%=====
elseif strcmp(ato,'doz'),
v = get(gco,'Userdata');
s = get(gco,'String');
vv = eval(s,num2str(v));
st = round(vv*100000)/100000;
set(gco,'Userdata',vv,'String',num2str(st))
muda = 1;
return

%=====
elseif strcmp(ato,'xpi'),
v = get(gco,'Userdata');
s = get(gco,'String');
vv = eval(s,num2str(v));
st = round(vv*100000)/100000;
set(gco,'Userdata',vv,'String',num2str(st))
muda = 1;
return

%=====
elseif strcmp(ato,'ypi'),
v = get(gco,'Userdata');
s = get(gco,'String');
vv = eval(s,num2str(v));
st = round(vv*100000)/100000;
set(gco,'Userdata',vv,'String',num2str(st))
muda = 1;
return

%=====
elseif strcmp(ato,'zpi'),
v = get(gco,'Userdata');
s = get(gco,'String');
vv = eval(s,num2str(v));
st = round(vv*100000)/100000;
set(gco,'Userdata',vv,'String',num2str(st))
muda = 1;
return

%=====
elseif strcmp(ato,'thetax'),
v = get(gco,'Userdata');
s = get(gco,'String');
vv = eval(s,num2str(v));
vv = round(vv*100)/100;

```

```

        set(gco,'Userdata',vv,'String',num2str(vv))
        muda = 1;
        return

%=====
elseif strcmp(ato,'thetay'),
    v = get(gco,'Userdata');
    s = get(gco,'String');
    vv = eval(s,num2str(v));
    vv = round(vv*100)/100;
    set(gco,'Userdata',vv,'String',num2str(vv))
    muda = 1;
    return

%=====
elseif strcmp(ato,'xpf'),
    v = get(gco,'Userdata');
    s = get(gco,'String');
    vv = eval(s,num2str(v));
    st = round(vv*100000)/100000;
    set(gco,'Userdata',vv,'String',num2str(st))
    muda = 1;
    return

%=====
elseif strcmp(ato,'ypf'),
    v = get(gco,'Userdata');
    s = get(gco,'String');
    vv = eval(s,num2str(v));
    st = round(vv*100000)/100000;
    set(gco,'Userdata',vv,'String',num2str(st))
    muda = 1;
    return

%=====
elseif strcmp(ato,'zpf'),
    v = get(gco,'Userdata');
    s = get(gco,'String');
    vv = eval(s,num2str(v));
    st = round(vv*100000)/100000;
    set(gco,'Userdata',vv,'String',num2str(st))
    muda = 1;
    return

%=====
elseif strcmp(ato,'thetafx'),
    v = get(gco,'Userdata');
    s = get(gco,'String');
    vv = eval(s,num2str(v));
    vv = round(vv*100)/100;
    set(gco,'Userdata',vv,'String',num2str(vv))
    muda = 1;
    return

```

```
%=====
elseif strcmp(ato,'thetafy'),
    v = get(gco,'Userdata');
    s = get(gco,'String');
    vv = eval(s,num2str(v));
    vv = round(vv*100)/100;
    set(gco,'Userdata',vv,'String',num2str(vv))
    muda = 1;
    return
```

```
%=====
elseif strcmp(ato,'acel'),
    v = get(gco,'Userdata');
    s = get(gco,'String');
    vv = eval(s,num2str(v));
    st = round(vv*100000)/100000;
    set(gco,'Userdata',vv,'String',num2str(st))
    muda = 1;
    return
```

```
%=====
elseif strcmp(ato,'vel'),
    v = get(gco,'Userdata');
    s = get(gco,'String');
    vv = eval(s,num2str(v));
    st = round(vv*100000)/100000;
    set(gco,'Userdata',vv,'String',num2str(st))
    muda = 1;
    return
```

```
%=====
elseif strcmp(ato,'wx'),
    v = get(gco,'Userdata');
    s = get(gco,'String');
    vv = eval(s,num2str(v));
    st = round(vv*100000)/100000;
    set(gco,'Userdata',vv,'String',num2str(st))
    muda = 1;
    return
```

```
%=====
elseif strcmp(ato,'wy'),
    v = get(gco,'Userdata');
    s = get(gco,'String');
    vv = eval(s,num2str(v));
    st = round(vv*100000)/100000;
    set(gco,'Userdata',vv,'String',num2str(st))
    muda = 1;
    return
```

```
%=====
elseif strcmp(ato,'fx'),
    v = get(gco,'Userdata');
```

```

s = get(gco,'String');
vv = eval(s,num2str(v));
st = round(vv*100000)/100000;
set(gco,'Userdata',vv,'String',num2str(st))
muda = 1;
return

```

```

%=====

```

```

elseif strcmp(ato,'fz'),
    v = get(gco,'Userdata');
    s = get(gco,'String');
    vv = eval(s,num2str(v));
    st = round(vv*100000)/100000;
    set(gco,'Userdata',vv,'String',num2str(st))
    muda = 1;
    return

```

```

%=====

```

```

elseif strcmp(ato,'r3x'),
    v = get(gco,'Userdata');
    s = get(gco,'String');
    vv = eval(s,num2str(v));
    st = round(vv*100000)/100000;
    set(gco,'Userdata',vv,'String',num2str(st))
    muda = 1;
    return

```

```

%=====

```

```

elseif strcmp(ato,'r3y'),
    v = get(gco,'Userdata');
    s = get(gco,'String');
    vv = eval(s,num2str(v));
    st = round(vv*100000)/100000;
    set(gco,'Userdata',vv,'String',num2str(st))
    muda = 1;
    return

```

```

%=====

```

```

elseif strcmp(ato,'alfax'),
    v = get(gco,'Userdata');
    s = get(gco,'String');
    vv = eval(s,num2str(v));
    st = round(vv*100000)/100000;
    set(gco,'Userdata',vv,'String',num2str(st))
    muda = 1;
    return

```

```

%=====

```

```

elseif strcmp(ato,'alfay'),
    v = get(gco,'Userdata');
    s = get(gco,'String');
    vv = eval(s,num2str(v));
    st = round(vv*100000)/100000;
    set(gco,'Userdata',vv,'String',num2str(st))

```

```
muda = 1;  
return
```

```
%=====
```

```
elseif strcmp(ato,'mass'),  
    v = get(gco,'Userdata');  
    s = get(gco,'String');  
    vv = eval(s,num2str(v));  
    st = round(vv*100000)/100000;  
    set(gco,'Userdata',vv,'String',num2str(st))  
    muda = 1;  
    return
```

```
%=====
```

```
elseif strcmp(ato,'Jx'),  
    v = get(gco,'Userdata');  
    s = get(gco,'String');  
    vv = eval(s,num2str(v));  
    st = round(vv*100000)/100000;  
    set(gco,'Userdata',vv,'String',num2str(st))  
    muda = 1;  
    return
```

```
%=====
```

```
elseif strcmp(ato,'Jy'),  
    v = get(gco,'Userdata');  
    s = get(gco,'String');  
    vv = eval(s,num2str(v));  
    st = round(vv*100000)/100000;  
    set(gco,'Userdata',vv,'String',num2str(st))  
    muda = 1;  
    return
```

```
%=====
```

```
elseif strcmp(ato,'Jz'),  
    v = get(gco,'Userdata');  
    s = get(gco,'String');  
    vv = eval(s,num2str(v));  
    st = round(vv*100000)/100000;  
    set(gco,'Userdata',vv,'String',num2str(st))  
    muda = 1;  
    return
```

```
%=====
```

```
elseif strcmp(ato,'TelaBarra'),  
    figure(figbarra);  
    return
```

```
%=====
```

```
elseif strcmp(ato,'Telaini'),  
    figure(figurefig);  
    return
```

```
%=====
```

```
elseif strcmp(ato,'sair'),  
    figure(figforcaa);  
    close(gcf);  
    figure(figforcab);  
    close(gcf);  
    figure(figforcac);  
    close(gcf);  
    figure(figforcad);  
    close(gcf);  
    figure(figtraj);  
    close(gcf);  
    figure(figbarra);  
    close(gcf);  
    figure(nerofig);  
    close(gcf);  
    clear;
```

```
end
```