

Trabalho Computacional 02

Última Atualização: 29 de agosto de 2024

Objetivo: O propósito deste trabalho é realizar uma animação gráfica de um sistema mecânico, em particular, um pêndulo invertido posicionado sobre um carro. Para computar as variáveis do sistema, utiliza-se novamente os resolvidores de equações diferenciais disponíveis no Matlab/Octave.

Definição do Sistema: Considere o pêndulo invertido mostrado na Figura 1. Adotando as seguintes escolhas de variáveis

- x_1 : posição angular da haste θ ;
- x_2 : velocidade angular da haste $\dot{\theta}$;
- x_3 : posição do carro x ;
- x_4 : velocidade do carro \dot{x} ;

tem-se a seguinte representação de estados da dinâmica

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{u \cos(x_1) - (m + M)g \sin(x_1) + m\ell \cos(x_1) \sin(x_1)x_2^2}{m\ell \cos^2(x_1) - (m + M)\ell} \\ \dot{x}_3 &= x_4 \\ \dot{x}_4 &= \frac{u + m\ell \sin(x_1)x_2^2 - mg \cos(x_1) \sin(x_1)}{m + M - m \cos^2(x_1)}\end{aligned}\tag{1}$$

Tarefas: Considere o início do código do *script* `animacaoPendulo.m` (disponibilizado no Google Classroom) reproduzido na sequência:

```
1 x0 = [deg2rad(30) 2 0 0]';
2 tempoTotal = 3;
3 %parametros = [m M ell g];
4 parametros = [0.5 2 0.5 9.81];
5 [theta, dotTheta, x, dotX, t] = simulaPendulo(x0, tempoTotal, parametros)
```

A primeira tarefa é implementar a função `simulaPendulo`, que basicamente vai computar os valores $x_i(t)$, $i = 1, \dots, 4$, ao longo do tempo considerando os valores especificados para os parâmetros do pêndulo. A estrutura desta função é a mesma dos códigos disponibilizados no trabalho computacional 01. O trabalho (que deve ser realizado com bastante cuidado) é introduzir as equações dadas em (1) na função que é passada como parâmetro de entrada do *script* `ode45`. Como força u de entrada, deve-se utilizar o seguinte sinal (realimentação dos estados):

$$u = [137,78 \quad 25,97 \quad 22,36 \quad 27,57] x = 137,78x_1 + 25,97x_2 + 22,36x_3 + 27,57x_4$$

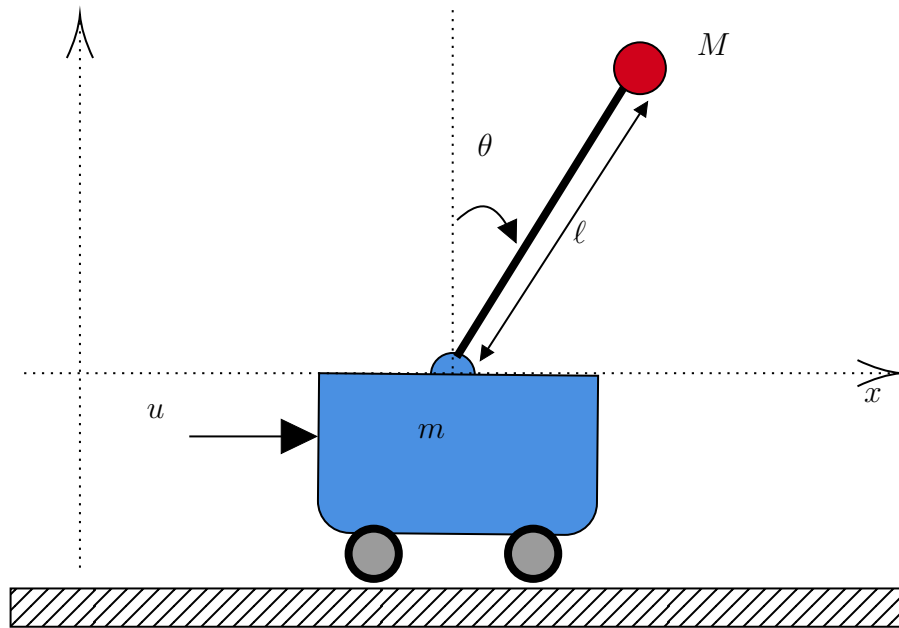


Figura 1: Pêndulo invertido sobre um carro.

A segunda tarefa consiste em produzir um vídeo (ou alternativamente um conjunto de imagens) que mostra o movimento do sistema ao longo do tempo. Para isso é utilizada a função `videoPendulo`, e a única implementação a ser realizada é a definição do primeiro parâmetro da função, que informa a origem da haste. Concretamente, defina o valor da variável `origemHaste`. O resultado deve mostrar o pêndulo sendo estabilizado na posição vertical. Como segundo exercício, vá aumentando gradativamente o tamanho da haste do pêndulo (ℓ) até obter um comportamento instável. Um segundo vídeo com o comportamento instável também deve ser entregue.

Apresentação dos resultados:

- Todos os códigos fontes utilizados, incluindo aqueles que foram disponibilizados;
- Um vídeo (ou um conjunto de imagens em um arquivo zip) com a animação do sistema estável. Nome do arquivo: `penduloEstavel.avi` (ou `penduloEstavel.zip`);
- Um vídeo (ou um conjunto de imagens em um arquivo zip) com a animação do sistema instável. Nome do arquivo: `penduloInstavel.avi` (ou `penduloInstavel.zip`);

Nota sobre o vídeo: Por ora, o código que gera o vídeo funciona apenas no Matlab. Aqueles que estiverem usando o Octave devem gerar um conjunto de imagens.

Formato de entrega: Arquivo PDF contendo a identificação da disciplina e dos alunos (nome e RA), e todos os códigos fontes utilizados. Dois arquivos referentes às animações.

Pontos Extras:

- Crie uma animação para evidenciar o movimento das rodas do carro de acordo com posição do carro, e ganhe 1,5 pontos extras (escala de 0 a 10);

- Faça a cor da massa m mudar de acordo com a velocidade angular. Se a velocidade for nula, então use a cor branca. Se a velocidade for máxima, use a cor vermelha. Para velocidades intermediárias, faça uma interpolação linear entre o branco e o vermelho. Valor da tarefa: 0,5 ponto.