Universidade Estadual de Campinas

Faculdade de Engenharia Mecânica

ES827 - Robótica Industrial

Projeto Final

Dinâmica e Cinemática do Robô Puma 560

Nome:	*RA*
Daniel Dello Russo Oliveira	101918
Marcelli Tiemi Kian	117892

28 de junho de 2015

1 Objetivos

O objetivo desse projeto é simular e analisar aspectos dinâmicos de um robô Puma 560, e cinemática de um robô criado pelo arquivo “ini_Rbt.m”[3] conforme sugerido pelo roteiro[1].

2 Dinâmica do Puma 560

Pelo roteiro foi sugerido para a análise dinâmica, utilizar o robô Puma 560 construído automaticamente pela “toolbox”de robótica[2], a função “accel”e a “Matlab Function”pelo Simulink. Entretanto, algumas modificações foram feitas. Retiramos o atrito seco (de Coulomb) nas juntas do robô, pois gerava longos tempos de simulação, e utilizamos um “script”do Matlab em vez de usar o Simulink.

2.1 Malha aberta

A seção 6.3 da tese[5] utilizada como base mostra simulações feitas com o robô em malha aberta. A partir de entradas desejadas para q_{des} e $\dot{q_{des}}$ calculamos o vetor de torque u necessário para manter a posição. Como podemos observar, fizemos as simulações abaixo utilizando o robô Puma 560 também em malha aberta para as mesmas condições iniciais informadas pela tese[5], mas com o vetor u calculado para o robô Puma 560.

$$q_{des} = \begin{bmatrix} 0 & \frac{\pi}{2} & -\frac{\pi}{2} & 0 & 0 & 0 \end{bmatrix}^T \tag{1}$$

$$\dot{q_{des}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{2}$$

$$u = \begin{bmatrix} 0 & -0.7752 & 0.2489 & 0 & 0 & 0 \end{bmatrix}^T \tag{3}$$

2.1.1 Simulação 1

Condições iniciais:

$$q_{init} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{4}$$

$$\dot{q_{init}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{5}$$

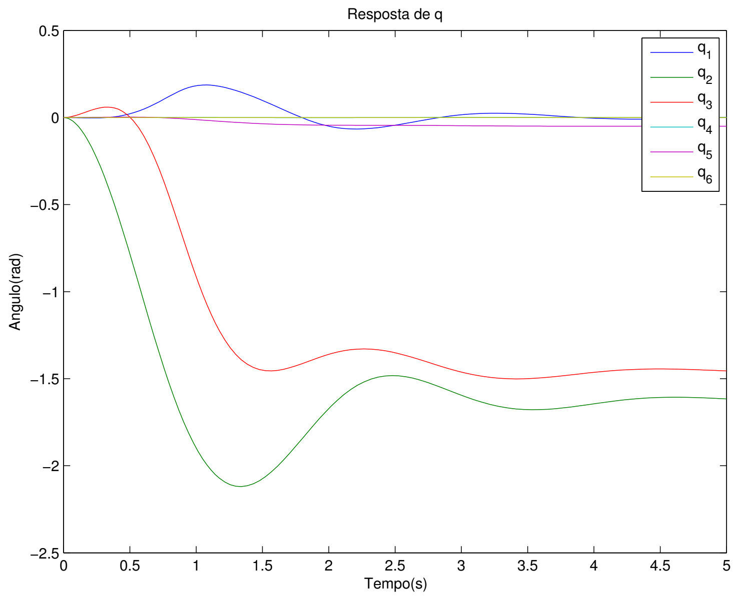

Figura 1: Simulação 1 para o robô Puma

2.1.2 Simulação 2

Condições iniciais:

$$q_{init} = \begin{bmatrix} 0 & \pi & -\frac{\pi}{2} & 0 & 0 & 0 \end{bmatrix}^T \tag{6}$$

$$\dot{q_{init}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{7}$$

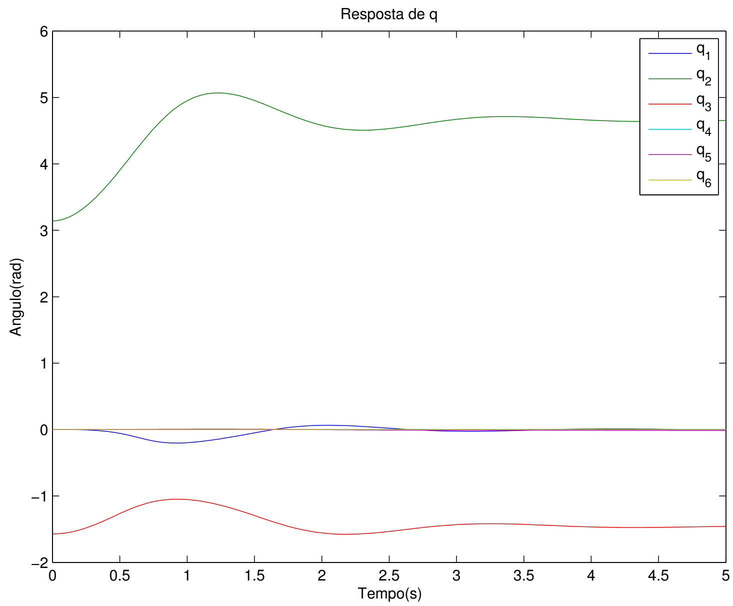

Figura 2: Simulação 2 para o robô Puma

2.1.3 Simulação 3

Condições iniciais:

$$q_{init} = \begin{bmatrix} 0 & \frac{\pi}{2} & -\frac{\pi}{2} & 0 & 0 & 0 \end{bmatrix}^T \tag{8}$$

$$\dot{q_{init}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{9}$$

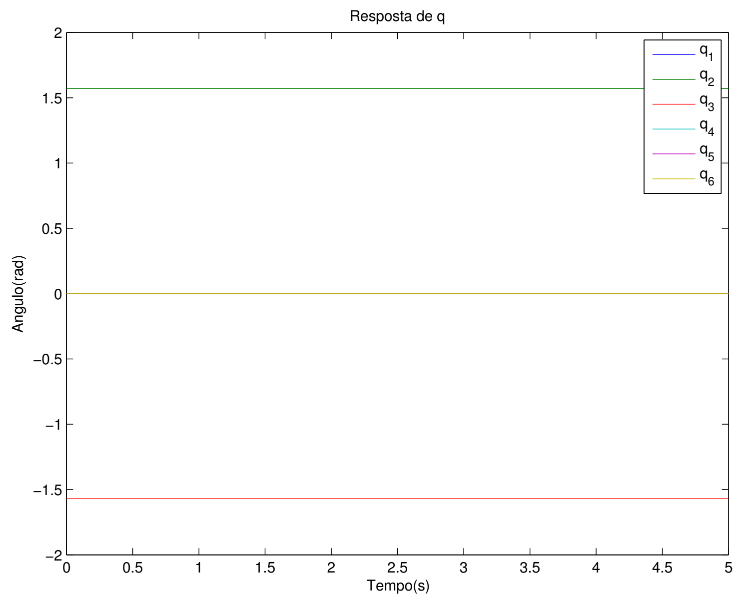

Figura 3: Simulação 3 para o robô Puma

2.1.4 Simulação 4

Condições iniciais:

$$q_{init} = \begin{bmatrix} 0 & \frac{\pi}{2} + 0.05 & -\frac{\pi}{2} & 0 & 0 & 0 \end{bmatrix}^T \tag{10}$$

$$\dot{q_{init}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{11}$$

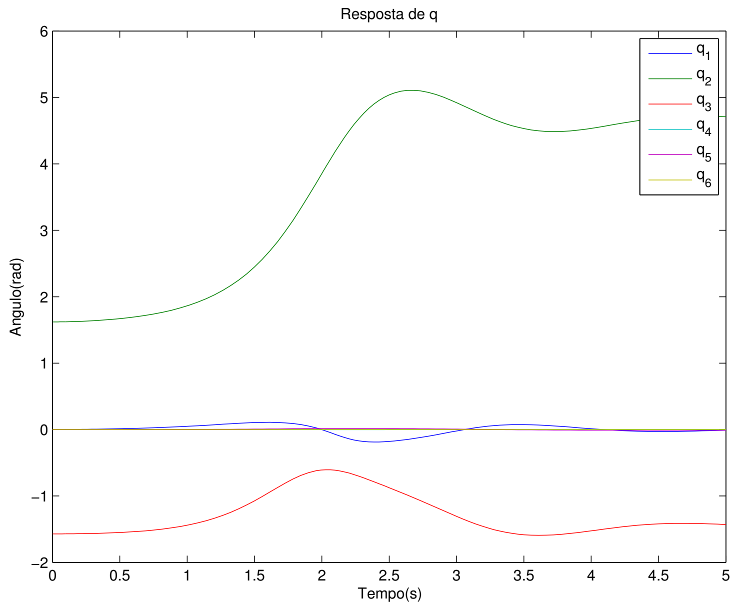

Figura 4: Simulação 4 para o robô Puma

2.2 Equilíbrio de energia

A seção 6.4 da tese[5] utilizada como base trata da análise de equilíbrio de energia do robô. No nosso estudo, vamos considerar nulos os esforços de compensação da força gravitacional, fazendo um estudo da energia cinética do robô Puma 560 (translação e rotação), conforme mostrado nas simulações a seguir, sendo as simulações 5 e 6 feitas com atrito viscoso (sem atrito seco), e sem atrito viscoso nem atrito seco.

2.2.1 Simulação 5

Condições iniciais e torque:

$$q_{init} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{12}$$

$$\dot{q_{init}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{13}$$

$$\tau = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{14}$$

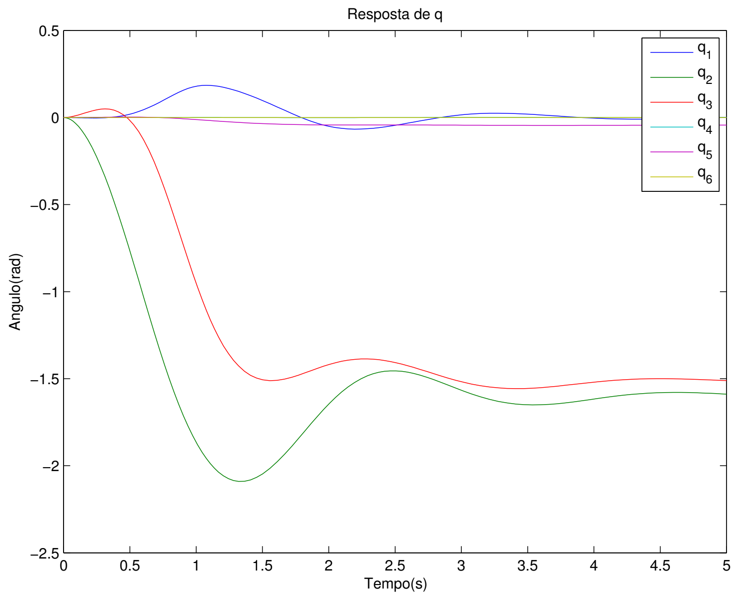

Figura 5: Simulação 5 para o robô Puma com atrito viscoso

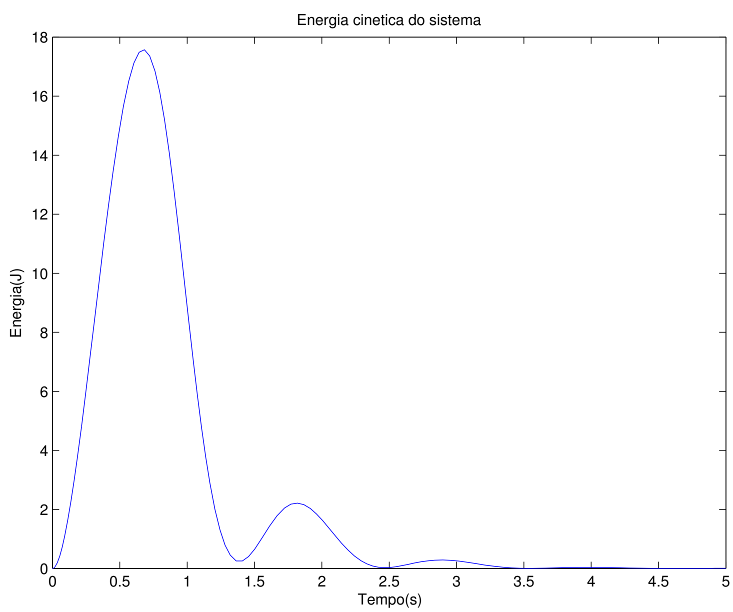

Figura 6: Simulação 5 para energia do robô Puma com atrito viscoso

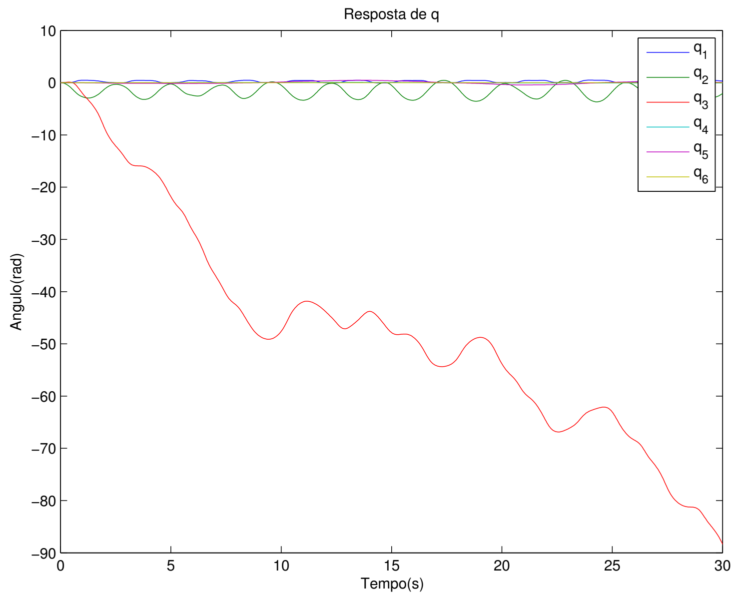

Figura 7: Simulação 5 para o robô Puma sem atrito viscoso

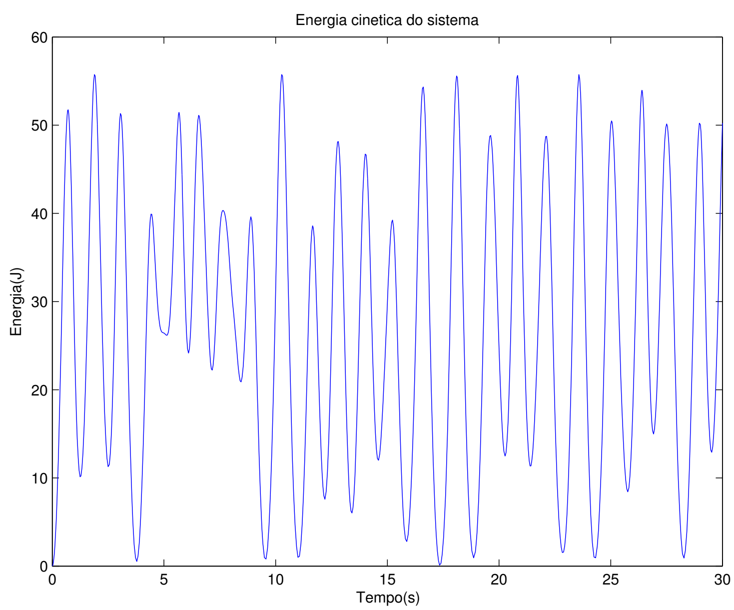

Figura 8: Simulação 5 para energia do robô Puma sem atrito viscoso

2.2.2 Simulação 6

Condições iniciais e torque:

$$q_{init} = \begin{bmatrix} 0 & 0.000001 & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{15}$$

$$\dot{q_{init}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{16}$$

$$\tau = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{17}$$

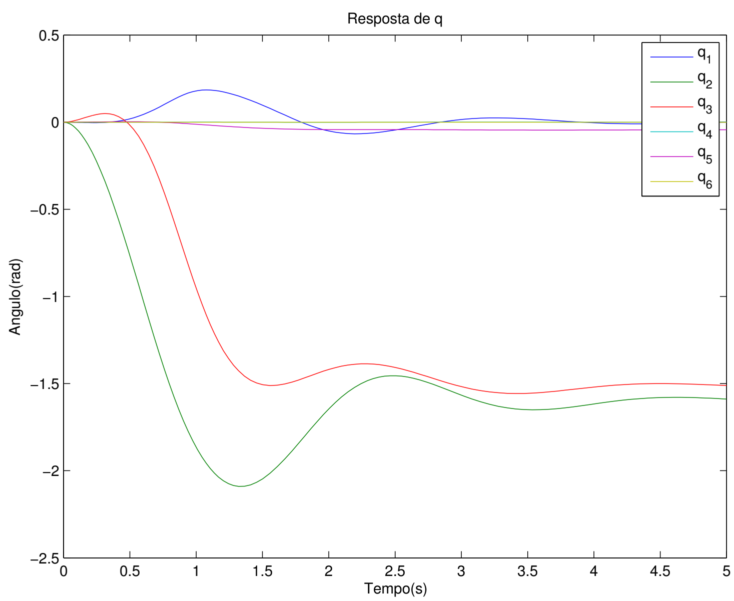

Figura 9: Simulação 6 para o robô Puma com atrito viscoso

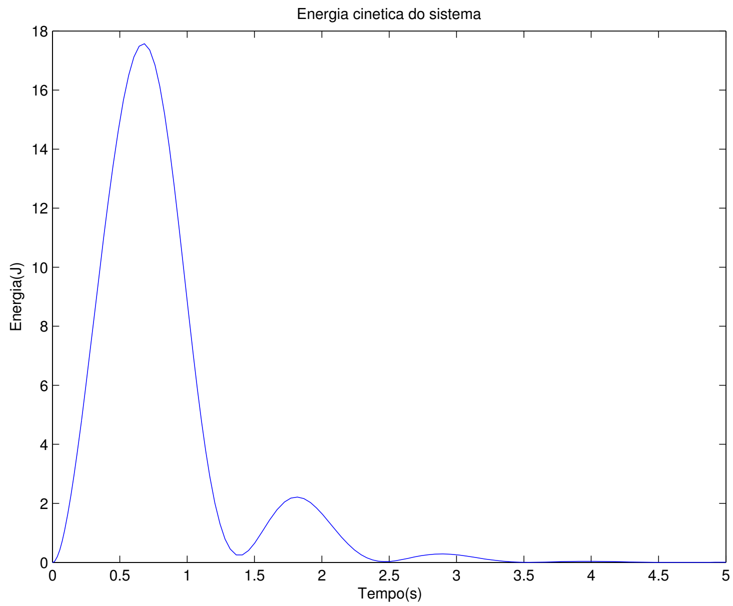

Figura 10: Simulação 6 para energia o robô Puma com atrito viscoso

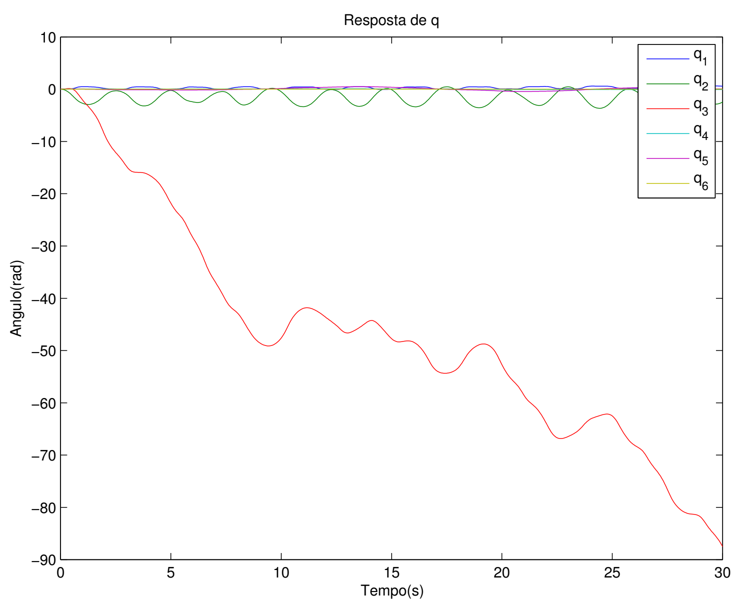

Figura 11: Simulação 6 para o robô Puma sem atrito viscoso

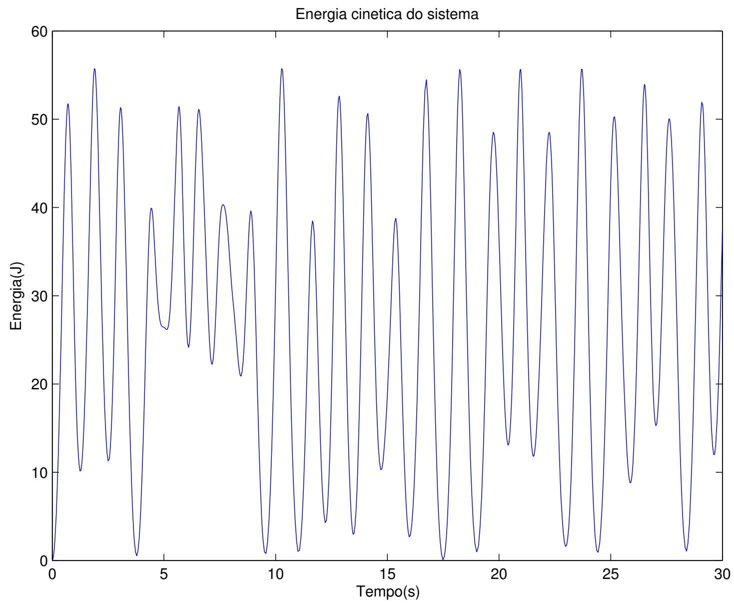

Figura 12: Simulação 6 para energia do robô Puma sem atrito viscoso

2.3 Controle em malha fechada

Para fazer um controle efetivo do robô, iremos agora considerar um controlador para o sistema em malha fechada. Ao fechar a malha, temos como entrada do controlador a diferença entre a referência e a saída, também chamada de erro. Conforme proposto na tese[5] que estamos utilizando como base, iremos implementar um controlador PD, considerando cada uma das juntas como um sistema SISO.

Após verificar que os torques utilizados no robô Puma 560 eram aproximadamente um terço dos torques do robô utilizado na tese[5], utilizando como base os ganhos K_p e K_d do controlador calculado para o robô da tese, aprimoramos seus valores dividindo os da tese por três e chegamos em K_p e K_d dados pelas equações 18 e 19.

$$K_p = \begin{bmatrix} 16.6667 & 0 & 0 & 0 & 0 & 0 \\ 0 & 16.6667 & 0 & 0 & 0 & 0 \\ 0 & 0 & 16.6667 & 0 & 0 & 0 \\ 0 & 0 & 0 & 16.6667 & 0 & 0 \\ 0 & 0 & 0 & 0 & 16.6667 & 0 \\ 0 & 0 & 0 & 0 & 0 & 20 \end{bmatrix} \tag{18}$$

$$K_d = \begin{bmatrix} 6.6667 & 0 & 0 & 0 & 0 & 0 \\ 0 & 6.6667 & 0 & 0 & 0 & 0 \\ 0 & 0 & 6.6667 & 0 & 0 & 0 \\ 0 & 0 & 0 & 6.6667 & 0 & 0 \\ 0 & 0 & 0 & 0 & 6.6667 & 0 \\ 0 & 0 & 0 & 0 & 0 & 7.3333 \end{bmatrix} \tag{19}$$

2.3.1 Simulação 7

Condições iniciais e referência:

$$q_{init} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{20}$$

$$\dot{q_{init}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{21}$$

$$q_{ref} = \begin{bmatrix} \frac{\pi}{2} & 0 & -\frac{\pi}{2} & \pi & \frac{\pi}{2} & -\pi \end{bmatrix}^T \tag{22}$$

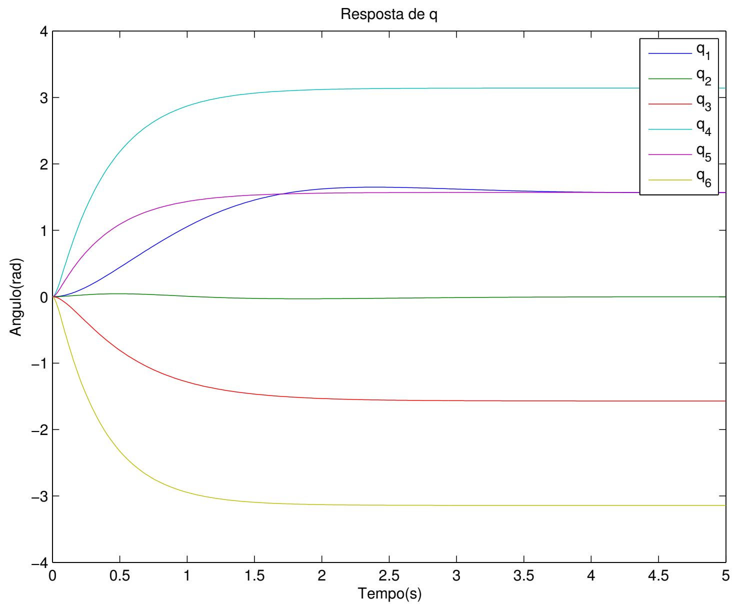

Figura 13: Simulação 7 para o robô Puma com controlador

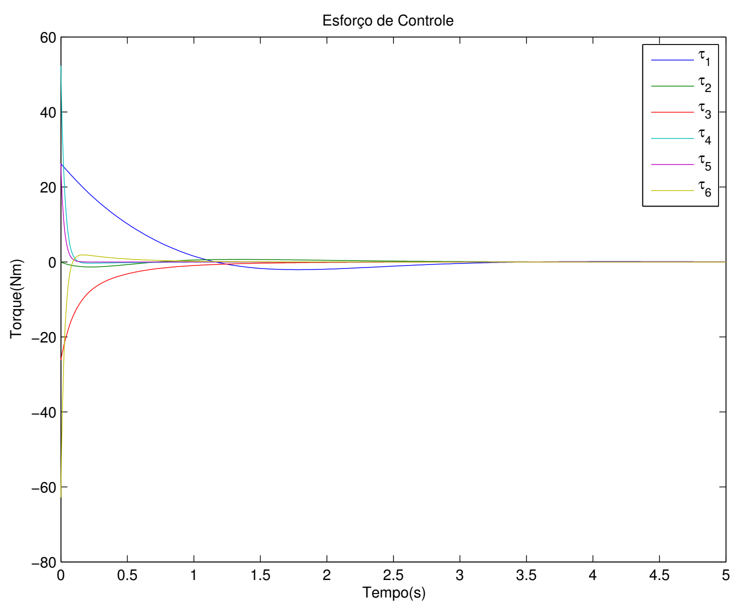

Figura 14: Esforços de controle para simulação 7 para o robô Puma com controlador

2.3.2 Simulação 8

Condições iniciais e referência:

$$q_{init} = \begin{bmatrix} 0 & \pi & -\frac{\pi}{2} & 0 & 0 & 0 \end{bmatrix}^T \tag{23}$$

$$\dot{q_{init}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{24}$$

$$q_{ref} = \begin{bmatrix} \pi & 0 & 0 & \pi & -\frac{\pi}{2} & 0 \end{bmatrix}^T \tag{25}$$

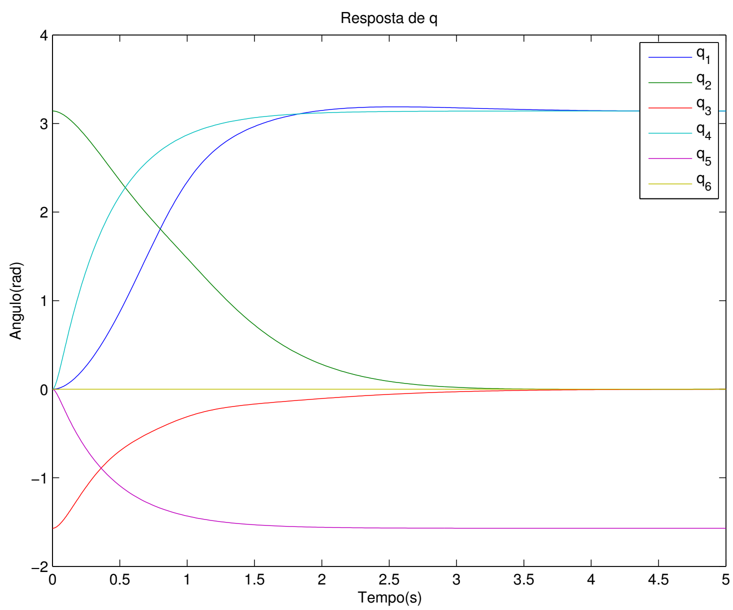

Figura 15: Simulação 8 para o robô Puma com controlador

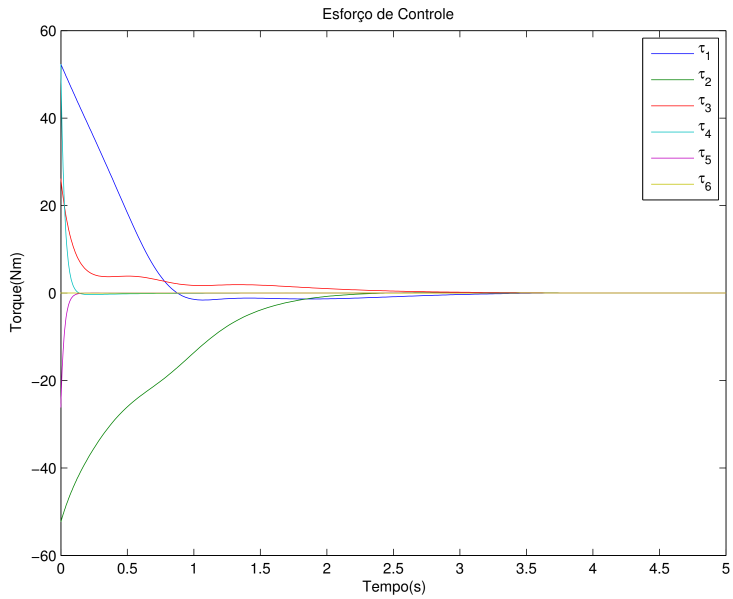

Figura 16: Esforços de controle para simulação 8 para o robô Puma com controlador

2.3.3 Simulação 9

Condições iniciais e referência:

$$q_{init} = \begin{bmatrix} 0 & \frac{\pi}{2} & -\frac{\pi}{2} & 0 & 0 & 0 \end{bmatrix}^T \tag{26}$$

$$\dot{q_{init}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{27}$$

$$q_{ref} = \begin{bmatrix} -\pi & \pi & -\pi & -\pi & -\frac{\pi}{2} & \pi \end{bmatrix}^T \tag{28}$$

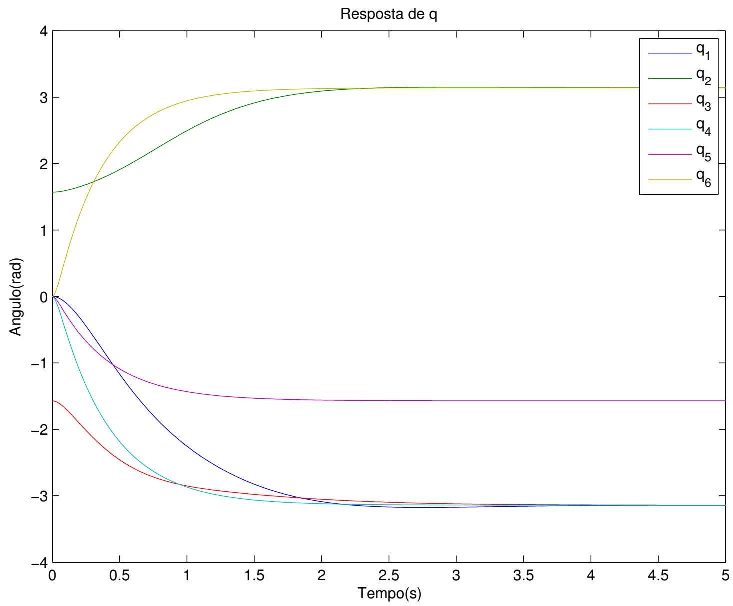

Figura 17: Simulação 9 para o robô Puma com controlador

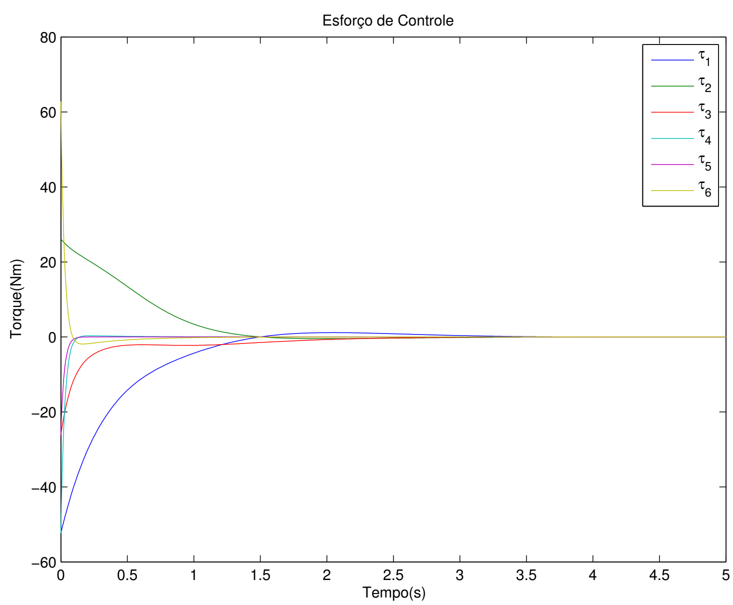

Figura 18: Esforços de controle para simulação 9 para o robô Puma com controlador

3 Cinemática

Para o estudo da cinemática, iremos usar o robô criado por "ini_Rbt"[3], e não o Puma 560 com que estávamos trabalhando até o momento. Foram feitos utilizando "script"do Matlab em que descrevemos a trajetória espacial desejada, e o robô calcula os ângulos de suas juntas para cada ponto da trajetória por meio da cinemática inversa usando a toolbox[2]. Estes valores são armazenados para criar a animação do movimento do robô, em que colocamos o robô em cada posição da sequência.

3.1 Elipse

Uma das trajetórias que testamos foi uma elipse no chão, conforme figura 19, e também uma elipse na parede, conforme **??**. Notamos que os raios da elipse e sua distância em relação à base do robô possuem um limite, já que existe um alcance restrito ao robô dado pelo comprimento de seus elos.

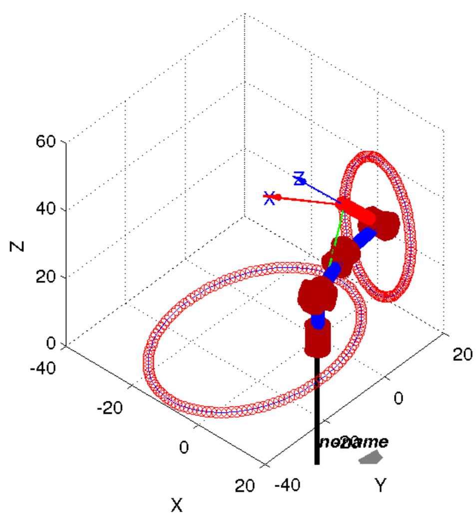

Figura 19: Execução da trajetória em elipse no chão e na parede

4 Questões

- **Por que na seção "Controle em malha fechada"os torques finais obtidos são nulos? O torque final não deveria ser igual ao torque**

de compensação da força da gravidade?

Conforme apontado por Herman Høifødt[5] em sua tese, na seção 6.5.1, ao somar a carga gravitacional ao esforço de controle do controlador PD podemos controlar o sistema. Fizemos então a mesma simplificação que foi feita por ele na seção 6.5.2 de remover o termo gravitacional diretamente do modelo do robô ao invés de somar esse termo na resposta do controlador, por isso obtemos os torques finais como nulos. Se não nos utilizássemos dessa simplificação, a equação de nosso controlador seria da seguinte forma e o esforço final seria não nulo:

$$u = -K_p \cdot (q - q_r) - K_d \cdot \dot{q} + gravload(q); \tag{29}$$

- **Na seção "Equilíbrio de energia cinética"foi indicado um comando do Matlab para se obter a matriz de massa/inércia no espaço das juntas → $M(q)$. Como você faria para obter essa matriz de massa utilizando o método de Newton-Euler além de manipulação simbólica? Veja os apêndices B e C da tese[5]. Essa matriz corresponde à matriz $M(q)$ da equação geral da dinâmica de um robô, como dado abaixo, onde "u"é o vetor de torques nas juntas.**

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = u \tag{30}$$

- **Por que essa matriz de massa $M(q)$ não é do tipo da matriz de massa abaixo de um corpo rígido, que é uma matriz $6x6$ com uma matriz diagonal M e outra matriz de inércia do corpo J?**

$$\bar{M} = \begin{bmatrix} M & 0 \\ 0 & J \end{bmatrix} \tag{31}$$

5 Referências

[1] Roteiro do projeto disponibilizado para os alunos

[2] Peter Corke, Robotics Toolbox, disponível em http://www.petercorke.com/Robotics_Toolbox.html

[3] Código ini_Rbt.m fornecido pelo professor

[4] Código Rbtsquare.m fornecido pelo professor

[5] Herman Høifødt, Dynamic Modeling and Simulation of Robot Manipulators, The Newton-Euler Formulation, disponível em http://www.diva-portal.org/smash/get/diva2:436733/FULLTEXT01.pdf