



Projeto 1 – Pendulo Invertido

Neste projeto os alunos devem implementar um sistemas baseado em regras *fuzzy* para o pendulo invertido, um problema clássico de controle.

Condições e Datas

O projeto deve ser realizado utilizando GNU Octave ou MATLAB em grupos com no máximo três alunos. Não será aceito trabalho feito em outra linguagem de programação ou em grupos com mais de três alunos.

Deve-se fazer o upload no sistema Ensino Aberto Moodle dos seguintes arquivos até o dia 06/04/2018 (23h59min):

- Arquivo texto (`relatório.pdf`), que não deve ter mais que 10 páginas, descrevendo de forma clara e objetiva os procedimentos adotados e as conclusões.
 - Código (`controladorfuzzy.m`), implementado em MATLAB ou GNU/Octave, que fornece a aceleração a .
-

Instruções

O pendulo invertido é descrito pela variável de estado $\mathbf{z} = [z_1, z_2, z_3, z_4, z_5]$, em que

- $z_1 = t$ denota o tempo,
- $z_2 = \theta$ é o ângulo do pendulo com a vertical,
- $z_3 = \dot{\theta}$ é a velocidade angular,
- $z_4 = x$ representa a posição do centro da base,
- $z_5 = \dot{x}$ representa a velocidade da base.

Conhecendo \mathbf{z} (variável independente), o sistema *fuzzy* deve fornecer a aceleração horizontal a da base de modo que a haste não caia e a base não saia de uma região delimitada. Neste problema, a região delimitada corresponde ao intervalo $[-100, 100]$ e o tamanho da haste é $L = 0.5$. A haste deve ser equilibrada por um determinado intervalo de tempo ou até que $\theta = 0$, $\dot{\theta} = 0$ e $\dot{x} = 0$.

A simulação do pendulo invertido é realizada executando em MATLAB ou GNU/Octave o código `simulapendulo.m` disponível para *download* na página da disciplina. A dinâmica do pendulo é determinada através da função `pendmotion.m`, que não deve ser modificada.

O código `simulapendulo` considera o estado inicial $\mathbf{z} = [0, 30, 0, 0, 0]$. Pode-se também utilizar outro estado inicial através do comando `simulapendulo([z1, z2, z3, z4, z5])`. A linha 21 desse código deve ser adaptada da seguinte forma: $a = \text{controladorfuzzy}(\mathbf{z})$, em que `controladorfuzzy` é a função que implementa o sistema de regras *fuzzy*. Essa função deve receber o estado atual \mathbf{z} e retornar a aceleração a .