Universidade Federal do Rio Grande do Norte

EGM0001 - SISTEMAS DINÂMICOS E SERVOMECANISMOS

Atividade Computacional 02

Controle de Sistema de 1ª Ordem com atraso

Discente:
Camila Barbosa Gomes de Araújo

Docente:
Wallace Moreira Bessa

14 de dezembro de 2020

1 Modelo proposto

O sistema a ser controlado trata-se do subsistema Motor-Roda do Rover Perseverance. O modelo dinâmico proposta para esse subsistema está descrito pela Equação 1.

$$J\dot{\omega} = -\mu_v\omega - \mu_d sgn(\omega) + Ki \tag{1}$$

Onde:

- J: Inércia
- ω: Velocidade angular
- μ_v: Coeficiente de Atrito Dinâmico
- μ_d: Coeficiente de Atrito Estático
- K: Constante do Motor
- i: Corrente

Para fins de implementação, serão considerados:

- $J = 2 \times 10^{-4}$
- $K = 4 \times 10^{-2}$

Nesse trabalho, será avaliado o desempenho do controlador quando sujeito a uma frequência de atua¸c ao menor do que a do simulador, ou seja, para cada ação de controle, devem ser considerados n $\geq$ 1 passos de integração, considerando a lei de controle descrita na Equação 2.

$$i = \frac{J(\dot{\omega} - \lambda e)}{K} \tag{2}$$

Sendo:

- $e = \omega - \omega_d \longrightarrow$ Erro
- $\lambda \longrightarrow$ Lambda (Coeficiente de aprendizado)

Nas seções seguintes, serão tratadas a implementação do subsistema em Python, os resultados com gráficos do comportamento do sistema e as devidas conclusões sobre o experimento.

2 Implementação do Sistema

Aproveitando-se da implementação da primeira atividade computacional, o novo sistema proposto foi desenvolveido com algumas alterações:

1. Na sequência da simulação, inclusão de um loop sem o cálculo da corrente com tamanho igual a frequência desejada para a ação de controle
2. Cálculo da corrente (pela Equação 2)
3. Retorno a sequência da simulação

O código precisou ser implementado de forma que a corrente só fosse calculada após n iterações, simulando assim, o atraso da ação de controle. No trecho de código abaixo, a função principal do código é exposta, com as devidas alterações comentads acima.

```
def generate_values(tf, W, Wd, _lambda, J, K, Ud, Uv, n=1):
    _w = []
    _e = []
    _i = []
    t = tf*1000-1
    e = W - Wd
    i = J * (-1 * _lambda * e) / K
    while t > 0:
        for f in range(n):
            _w.append(W) ### saves in a list the next value of w
            _e.append(e) ### save the error to plot later
            _i.append(i)
            W = rungeKutta(W, 0.001, Uv, Ud, K, i, J)
            e = W - Wd
            t = t - 1
            if t < 0:
                break
        i = J * (-1 * _lambda * e) / K ### calculates the current
    return _w, _e, _i
```

O código pode ser encontrado em `https://github.com/camilabga/rover-control`. A função que implementa essa atividade é *plot_frequency*.

3 Resultados e Conclusões

Com o objetivo de analisar o comportamento do sistema com a alteração do delay da ação de controle, foram realizadas diversas simulações. As Figuras 1 e 2 demonstram como a velocidade angular e a corrente se comportam com a variação de n.

Sobre a influência dos diferentes atrasos da ação de controle no sistema, é possível notar um comportamento claro se intensificando com o aumento do valor de n, ou seja, quanto mais lenta for a reação do controlador.

Na Figura 1, podemos notar que quando n=1, ou seja, quando a ação de controle é instantânea a variações no sistema, o *target* é atingido mais lentamente e que a ação de controle é mais suave. Quanto maior o n, é possível notar que a ação de controle fica mais acelerada e cada vez menos suave. Isso ocorre pois, quanto maior o atraso do controlador com relação ao sistema, mais o sistema sofrerá alterações antes que o controlador tome conhecimento, e o erro acaba por sofrer variações bruscas entre as iterações de controle.

Para os valores de n maiores que 1000, a diferença entre as retas que representam a variação das velocidades angulares são mais evidentes. Chegando a ter um overshoot com n = 5000x.

Na Figura 2, o comportamento da corrente com a variação do delay da ação de controle é exposto. Aqui, é possível visualizar com mais clareza as ações de controle distintas.

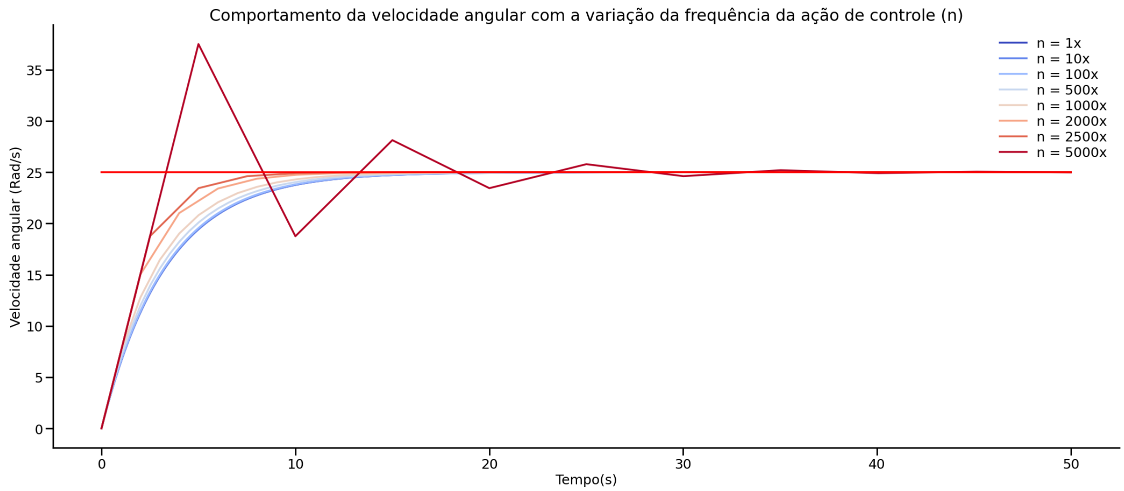

Figura 1: Plot com os valores fixos de: $\omega_d = 25$, μ_d e $\mu_v = 0$

Enquanto a curva da corrente de n=1 se assemelha a uma exponencial tendendo a um valor próximo de zero, quanto maior o n, podemos notar os degraus que caracterizam ao atraso do controlador, mantendo a mesma ação de controle por um determinado tempo. Tempo esse que aumenta com o n.

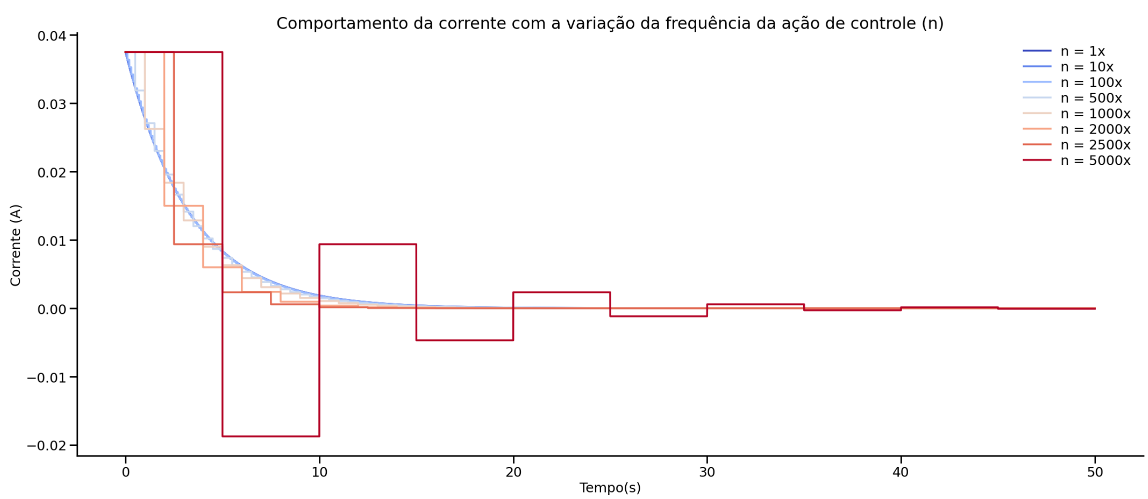

Figura 2: Plot com os valores fixos de: $\omega_d = 25$, μ_d e $\mu_v = 0$

Os experimentos para esse trabalho foram realizados com os valores fixos de $\omega_d = 25$, μ_d e $\mu_v = 0$, $\lambda = 0.3$ e tempo de execução de 50 segundos. O passo do controlador é em milisegundos.