

Programação de Sistemas de Tempo Real

Laboratório 3 - Simulação de Sistemas

*Edson Nascimento Silva - 21350694, *Evaldo Patrik Cardoso - 21453640

I. OBJETIVOS

Este relatório tem como objetivo a simulação e descrição de um sistema de controle, no qual é simulado o sistema de controle de um robô. Cujas variáveis de entrada são a velocidade linear e a velocidade angular, e as variáveis controladas são a posição do centro de massa do robô e o seu ângulo de orientação.

II. PROBLEMA PROPOSTO

De acordo com o enunciado do problema proposto pelo professor, um robô com acionamento diferencial pode ser descrito pelo modelo do seguinte espaço de estados:

$$\dot{x}(t) = \begin{bmatrix} \sin(x_3) & 0 \\ \cos(x_3) & 0 \\ 0 & 1 \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} x(t)$$

Figura 1. Espaço de estados do robô.

Onde $x(t)$, a integral do espaço de estados, é uma matriz que possui as informações de posição em coordenadas cartesianas do robô e o ângulo em graus de sua orientação. Esta matriz multiplica a matriz $u(t)$, que possui informações de velocidade linear e angular do robô, estas sendo as entradas do sistema, que possuem a descrição mostrada na figura 2.

*Engenharia da Computação, Departamento de Eletrônica e Computação, Faculdade de Tecnologia, Universidade Federal do Amazonas, Manaus, AM. E-mail: netosilvan78@gmail.com, evaldo.cardoso@icomp.ufam.edu.br

$$u(t) = \begin{cases} 0 & , \text{para } t < 0 \\ \begin{bmatrix} 1 \\ 0.2\pi \end{bmatrix} & , \text{para } 0 \leq t < 10 \\ \begin{bmatrix} 1 \\ -0.2\pi \end{bmatrix} & , \text{para } t \geq 10 \end{cases}$$

Figura 2. Descrição da entrada do sistema.

O objetivo do presente trabalho é a simulação do sistema proposto para $t=[0,20]$ s e gerar como saída um arquivo contendo os valores de todas as variáveis do sistema a cada segundo de simulação para plotagem de gráficos.

III. INTRODUÇÃO TEÓRICA

A. Espaço de Estados

Em engenharia de controle, uma representação por espaço de estados é um modelo matemático de um sistema físico representado por um conjunto de variáveis de entrada, saída e estado relacionadas por equações diferenciais. Variáveis de estado são valores que evoluem ao longo do tempo de uma forma dependente dos valores impostos pelas variáveis de entrada. Variáveis de saída dependem dos valores das variáveis de estado.

Um Espaço de Estados é o Espaço Euclidiano em que as variáveis nos eixos são variáveis de estado. O Estado do sistema pode ser representado como um vetor dentro deste espaço[1].

IV. AMBIENTE

Em nosso ambiente utilizamos os seguintes arquivos:

- main.c - Arquivo de extensão .c onde são importados nossas dependências e bibliotecas que estamos utilizando em nosso projetos, neste mesmo arquivo se encontra a função main(), *função principal do nosso projeto*.
- simulation.c - Arquivo de extensão .c onde se encontram:
 - Função para cálculo de integrais: *double integral(double *func, double t, double tant)*.
 - Função para cálculo da entrada $u(t)$ do sistema de acordo com a representação da figura 2: *Matrix getSpeed(double t)*.
 - Função para cálculo da posição cartesiana e angular do robô: *Matrix getPosition(double t, double tant)*.
 - Função para cálculo das multiplicações matriciais que compõem a saída $y(t)$ do sistema: *Matrix simulate(double t, double tant)*.

```
#include "simulation.h"
#include "matrix.h"
#include <stdio.h>

int main(int argc, char const *argv[]) {

    int time = 20;
    Matrix id = matrix_identity("identidade", 3, 3);

    for(int i=0; i < time; i++){
        printf("tempo de simulação: %d\n", i);
        matrix_print(getPosition(i, i-1));
        matrix_print(getSpeed(i));
        matrix_print(simulate(i, getSpeed(i)));
    }

}
```

Figura 3. main.c

- simulation.h - Arquivo de extensão .h onde se encontram as definições para as funções do arquivo simulation.c.
- matrix.c - Arquivo de extensão .c onde se encontram as funções que realizam operações com matrizes. Biblioteca que foi conteúdo da atividade de laboratório 2.
- matrix.h - Arquivo de extensão .h onde se encontram as definições para as funções do arquivo matrix.c.
- Makefile.mk - Arquivo de extensão .mk onde estão nossos comandos de execução do nosso projeto.

```
all: Simulation

Simulation: matrix.c main.c simulation.c #arquivos de dependencia
clear
rm -rf Simulation
gcc -o Simulation matrix.c main.c simulation.c -lm

exec: Simulation
./Simulation

clean: #remove arquivo somasub
rm -rf Lab2
clear
```

Figura 4. Makefile.h

No arquivo de compilação Makefile temos as seguinte opções:

- make all ou make Simulation: este comando irá limpar o console *\$clear*, remover o arquivo executavel Simulation caso ele exista com o comando *\$rm -rf Simulation*, e fazer a compilação do projeto *\$gcc -o Simulation matrix.c main.c simulation.c -lm*
- make exec: este comando irá executar o arquivo Simulation caso ele exista com o comando *./Simulation \$output* e exportará como resultado um arquivo de extensão .ascii, o qual é passado como argumento na execução do projeto.

```
[x]-[strneoh@parrot]-[~]
$make exec output=nome_do_arquivo.ascii
```

Figura 5. Exemplo de uso do arquivo de execução Makefile

- make clean: este comando irá remover o arquivo executável Simulation caso ele exista com o comando *\$rm*

-rf Simulation, remover o arquivo .ascii passado como argumento na execução do projeto e depois limpar a tela com o comando *\$clear*.

V. GRÁFICOS E RESULTADOS

Como determinado no enunciado da atividade, alguns gráficos foram gerados a partir da simulação do sistema. Esta foi realizada em 2000 passos, entre os segundos 0 e 20 de simulação e os gráficos plotados com a ajuda do programa MATLAB.

A. Posição

O gráfico das coordenadas plotado indica que ao longo do tempo o robô se move em círculos em volta de um mesmo centro:

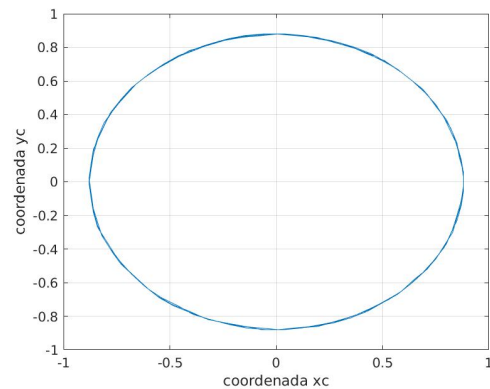
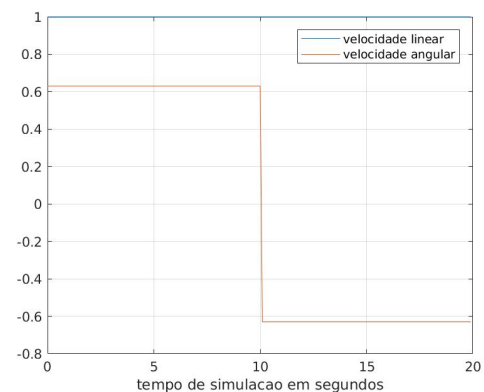


Figura 6. Gráfico yc x xc

B. Velocidade

Outro gráfico plotado foi tempo versus velocidades linear e angular, que apresentaram um comportamento previsível: enquanto a velocidade linear é descrita como uma linha reta ao longo de toda a simulação, a angular é um degrau que cai para um valor negativo em $t = 10s$. Valores compatíveis com a entrada $u(t)$ do sistema.

Figura 7. Gráfico de $u(t)$

C. Saída $y(t)$

O gráfico da saída $y(t)$ é descrito como as coordenadas x e y do robô e seu ângulo posicional θ plotados contra o tempo de simulação. Enquanto x e y apresentaram um comportamento senoidal defasado, o ângulo mostrou comportamento igual ao da velocidade angular de entrada do sistema.

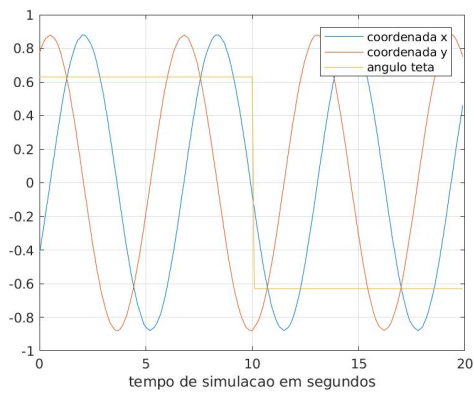


Figura 8. Gráfico de $y(t)$

VI. CONCLUSÃO

O objetivo do trabalho foi, entre outras coisas, expandir e desenvolver as bibliotecas iniciadas em semanas anteriores, além de desenvolver conhecimentos de escrita em arquivos e utilização de ponteiros e funções como parâmetros em outras funções. O atual trabalho também requereu o desenvolvimento de funções matriciais não presentes na biblioteca desenvolvida semana passada (como matriz identidade) além de um entendimento básico de sistemas de controle e espaço de estados.

VII. REFERÊNCIAS

- [1] Katalin M. Hangos; R. Lakner M. Gerzson (2001). Intelligent Control Systems: An Introduction with Examples. Springer. p. 254. ISBN 978-1-4020-0134-5.