

UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia

Departamento de Computação

Controle e Servomecanismo

Prática 1 - Revisão

Professor: Prof. Roberto Santos Inoue

Alexandre Strabello, 770076, Engenharia Física

São Carlos, 23 de maio de 2023

1 Execução do Tutorial

Conforme autorizado pelo professor da disciplina, a presente prática de revisão foi realizada utilizando o MATLAB® ao invés do SCILAB®, decorrente das futuras atividades a serem desenvolvidas em outras disciplinas.

As seções da atividade são dependentes, tal que os elementos criados na primeira parte são utilizados para as demais. Contudo, a representação das mesmas será feita de maneira individualizada, visando tornar a discussão mais organizada.

Todos os códigos, imagens e documentos utilizados para a escrita do presente relatório estão disponíveis no repositório <https://github.com/alestrab/Controle-e-Servo>, no interior da pasta correspondente à prática 1.

1.1 Vetores e matrizes

A construção dos vetores e matrizes foi realizada de acordo com o código apresentado a seguir:

```
%Vetores
x = [1 2 3 4 5];
y = x';
z = randi([0,100],5,1);
%Matrizes
A = randi([0,100],4,4);
B = A';
C = rand(4) + ones(4,4);
%Apresentacao dos resultados
disp('a')
disp(x)
disp('b')
disp(A)
disp('c')
disp(y)
disp('d')
disp(B)
disp('e')
disp(z)
disp('f')
disp(C)
```

Optou-se por criar as matrizes com elementos aleatórios apenas por explorar as funções disponíveis no software. Para a matriz C havia o requisito de que todos os elementos fossem não inteiros superiores a um, de modo que utilizou-se a função `rand()` para gerar números reais entre 0 e 1, seguido da adição de uma matriz unitária para cumprir o requisito.

1.2 Formatos de apresentação

Como as atividades dessa parte consistem apenas na apresentação dos resultados executados na seção 1.1, foi elaborado o código abaixo:

```
for i=1:5
    switch i
        case 1
            %5 digitos
            format short
            disp("Formato: short")
        case 2
            %5 digitos com exponencial
            format shorte
            disp("Formato: shorte")
        case 3
            %15 digitos
            format long
            disp("Formato: long")
        case 4
            %15 digitos com exponencial
            format longe
            disp("Formato: longe")
        otherwise
            %Apresentacao racional
            format rat
            disp("Formato: rat")
    end
    disp("x")
    disp(x)
    disp("A")
    disp(A)
    disp("z")
    disp(z)
    disp("C")
    disp(C)
    disp("-----")
end
```

Decidiu-se utilizar um *switch case* para reduzir a quantidade de linhas do código. Como foi necessário apresentar as mesmas variáveis em cinco formatos distintos, foi construída uma estrutura condicional, onde cada caso apresentava um formato distinto.

1.3 Números complexos

Para construção das variáveis com números complexos adicionou-se apenas a componente imaginária i na declaração. O código utilizado é dado a seguir:

```

k = [3+5i, 2-10j];
D = [3 + 5i, 2-10i; 7-13i, 1.7-4i];
%Vetor m dulo de k
mod_k = abs(k);
%Vetor fase de k, representado em radianos
ang_k = angle(k);
%Matriz do m dulo de D
mod_D = abs(D);
%Matriz da fase de D, representado em radianos
ang_D = angle(D);
%Matriz transposta
E = D.';
%Matriz conjugada transposta
F = D';
%Apresentacao dos resultados
disp('a')
disp(k)
disp('b')
disp(D)
disp('c')
disp(mod_k)
disp('d')
disp(ang_k)
disp('e')
disp(mod_D)
disp('f')
disp(ang_D)
disp('g')
disp(E)
disp('h')
disp(F)

```

Não houve regulamento quanto ao formato desejado para os ângulos dos números complexos, de modo que optou-se por deixar no padrão presente no MATLAB®, correspondendo a radianos.

1.4 Operações com matrizes

Nessa parte foram realizadas operações matemáticas utilizando as matrizes construídas na seção 1.1. O código implementado abaixo apresenta cada operação com os nomes *op_a*, *op_b*, *op_c* ..., representando os itens *a*), *b*), *c*) ... solicitados.

```

format short

op_a = A+C;
disp('a')
disp(op_a)

```

```

op_b = A-C;
disp('b')
disp(op_b)

op_c = A*C;
disp('c')
disp(op_c)

op_d = A.*C;
disp('d')
disp(op_d)

op_e = A^-1;
disp('e')
disp(op_e)

op_f = A\C;
disp('f')
disp(op_f)

op_g = A/C;
disp('g')
disp(op_g)

op_h = A./C;
disp('h')
disp(op_h)

op_i = exp(x);
disp('i')
disp(op_i)

op_j = exp(A);
disp('j')
disp(op_j)

op_k = x.^4;
disp('k')
disp(op_k)

op_l = A^2;
disp('l')
disp(op_l)

```

Como a estratégia utilizada para as matrizes era dada por uma geração aleatória, não é possível comparar todos os resultados diretamente com as variáveis produzidas na seção

1.1, já que a cada execução do programa os vetores e matrizes são alterados.

1.5 Vetores com espaçamento controlado

Nessa parte foram construídos diferentes vetores com diferentes passos. O código desenvolvido é dado a seguir:

```
t1 = linspace(0,10,11);  
t2 = linspace(0,10,10/0.1+1);  
t3 = linspace(-10,10,5);  
t4 = linspace(50,0,50/10+1);
```

Haviam dois diferentes vetores solicitados, um informava o espaçamento, o outro, o tamanho e a condição de que os elementos deveriam ser linearmente espaçados. Em ambos foi utilizada a função *linspace()*, porém com estratégias distintas. O terceiro argumento informado na função representa a quantidade de elementos do vetor a ser criado, cujos elementos serão espaçados linearmente entre os limites informados, assim, se é desejado criar um vetor com espaçamento fixo previamente informado, basta dividir o intervalo a ser utilizado pelo passo, depois incrementar o valor 1 para adicionar o limite inferior.

1.6 Manipulação de matrizes

Para a parte de manipulação de matrizes, foi desenvolvido o código abaixo:

```
A2 = A(3:4,1:2);  
disp('A2')  
disp(A2)  
A3 = zeros(length(A)+1,length(A));  
A3(1:4,1:4) = A;  
A3(5,:) = x(1:4);  
disp('A3')  
disp(A3)
```

A matriz *A2* foi criada simplesmente selecionando um intervalo na matriz *A*, enquanto a matriz *A3* foi definida inicialmente como uma matriz nula, com o mesmo tamanho da matriz *A* adicionado de uma linha. Em seguida, foram colocados os elementos solicitados.

1.7 Funções e gráficos

Decidiu-se reunir as seções envolvendo a criação de funções e a apresentação de resultados em gráficos. A função foi implementada em um arquivo distinto do principal, porém presente no mesmo repositório. O código da mesma é dado a seguir:

```
function [p] = funcTeste(t)  
    p = t.^2 + sin(2*t)+50;
```

```
end
```

No programa principal foi chamada a função previamente produzida e realizada a produção do gráfico correspondente. O código implementado é dado a seguir:

```
p1=funcTeste(t1);  
p2=funcTeste(t2);  
plot(t1,p1,'r*',t2,p2,'g-')  
legend('p1 vs t1','p2 vs t2','FontSize',20,'Location','northwest')  
title('Gráfico Atividade 1')  
xlabel('x','FontSize',20)  
ylabel('funcTeste(x)','FontSize',20)  
ax = gca;  
ax.FontSize = 15;  
grid on
```

2 Discussão e Avaliação de Resultados

2.1 Vetores e matrizes

Na Figura 1 estão apresentadas os vetores e matrizes construídos de acordo com o código implementado.

```
a)      1      2      3      4      5  
  
b)      68      9      15      65  
      4      82      66      80  
      7      82      52      45  
      52      72      98      43  
  
c)      1  
      2  
      3  
      4  
      5  
  
d)      68      4      7      52  
      9      82      82      72  
      15      66      52      98  
      65      80      45      43  
  
e)      98      71      50      47      6  
  
f)      1.8253      1.3909      1.3993      1.6280  
      1.0835      1.8314      1.5269      1.2920  
      1.1332      1.8034      1.4168      1.4317  
      1.1734      1.0605      1.6569      1.0155
```

Figura 1: Apresentação dos resultados para criação dos vetores e matrizes

Observa-se que todos estão de acordo com o esperado, com a matriz C cumprindo com o requisito necessário, com todos os números reais e superiores a 1.

2.2 Formatos de apresentação

Os resultados obtidos estão apresentados nas Figuras 2 e 3.

Formato: short						Formato: shorte						Formato: rat				
x	1	2	3	4	5	x	1	2	3	4	5	x	1	2	3	4
A						A						A				
12	58	26	34			12	58	26	34			12	58	26	34	
59	25	83	58			59	25	83	58			59	25	83	58	
22	29	99	10			22	29	99	10			22	29	99	10	
38	62	73	91			38	62	73	91			38	62	73	91	
z						z						z				
65	38	19	43	48		65	38	19	43	48		65	38	19	43	
C						C						C				
1.8797	1.0225	1.1788	1.4709			1.8797e+00	1.0225e+00	1.1788e+00	1.4709e+00			2171/1155	863/844	1299/1102	1543/1049	
1.8178	1.4253	1.4229	1.6959			1.8178e+00	1.4253e+00	1.4229e+00	1.6959e+00			2354/1295	2336/1639	286/201	3475/2049	
1.2607	1.3127	1.0942	1.6999			1.2607e+00	1.3127e+00	1.0942e+00	1.6999e+00			2321/1841	1125/857	1498/1369	1518/893	
1.5944	1.1615	1.5985	1.6385			1.5944e+00	1.1615e+00	1.5985e+00	1.6385e+00			904/567	1158/997	649/406	893/545	

Figura 2: Apresentação das variáveis no formato *short*, *shorte* e *rat*.

Formato: long						Formato: longe					
x	1	2	3	4	5	x	1	2	3	4	5
A						A					
12	58	26	34			12	58	26	34		
59	25	83	58			59	25	83	58		
22	29	99	10			22	29	99	10		
38	62	73	91			38	62	73	91		
z						z					
65	38	19	43	48		65	38	19	43	48	
C						C					
1.879653724481905	1.022512592740232	1.178766186752368	1.470924256358334			1.879653724481905e+00	1.022512592740232e+00	1.178766186752368e+00	1.470924256358334e+00		
1.817760559370642	1.425259320214135	1.422885689100085	1.695949313301608			1.817760559370642e+00	1.425259320214135e+00	1.422885689100085e+00	1.695949313301608e+00		
1.260727959055465	1.31271886820615	1.09422933887735	1.699887849928292			1.260727959055465e+00	1.31271886820615e+00	1.09422933887735e+00	1.699887849928292e+00		
1.594356250664331	1.161484744311750	1.598523668756741	1.638530758271838			1.594356250664331e+00	1.161484744311750e+00	1.598523668756741e+00	1.638530758271838e+00		

Figura 3: Apresentação das variáveis no formato *long* e *longe*.

Para os números inteiros, a alteração do formato de apresentação não resultou em nenhuma diferença, tal que a diferença é evidenciada apenas para a matriz *C*, cujos elementos são reais.

2.3 Números complexos

Na Figura 4 são apresentadas as variáveis complexas produzidas de acordo com as questões solicitadas na prática.

```

a)
3.0000 + 5.0000i    2.0000 -10.0000i

b)
3.0000 + 5.0000i    2.0000 -10.0000i
7.0000 -13.0000i    1.7000 - 4.0000i

c)
5.8310    10.1980

d)
1.0304    -1.3734

e)
5.8310    10.1980
14.7648    4.3463

f)
1.0304    -1.3734
-1.0769    -1.1689

g)
3.0000 + 5.0000i    7.0000 -13.0000i
2.0000 -10.0000i    1.7000 - 4.0000i

h)
3.0000 - 5.0000i    7.0000 +13.0000i
2.0000 +10.0000i    1.7000 + 4.0000i

```

Figura 4: Representação de números complexos produzidos no MATLAB®.

É possível identificar o funcionamento das operações de transposição de matrizes e da obtenção da matriz conjugada, mas com relação ao ângulo, acredita-se que a representação

em graus seria mais simples para visualização.

2.4 Operações com matrizes

As operações com matrizes são apresentadas na Figura 5.

a)	78.5853 46.5060 28.5472 13.8407 81.2238 66.6991 69.1386 51.2543 19.7513 72.8909 67.1493 97.8143 50.2551 77.9593 17.2575 35.2435	g)	1.0e+04 * 0.1651 -1.6755 0.2394 1.0974 0.1031 -1.0026 0.1400 0.6584 -0.1734 1.7934 -0.2555 -1.1717 0.0078 -0.1304 0.0238 0.0880
b)	75.4147 43.4940 25.4528 10.1593 78.7762 63.3009 66.8614 48.7457 16.2487 69.1091 64.8507 94.1857 47.7449 74.0407 14.7425 32.7565	h)	48.5722 29.8813 17.4507 6.5192 65.3695 38.2561 59.7212 39.8634 10.2783 37.5482 57.4266 52.9134 39.0409 38.7895 12.7236 27.3416
c)	239.4825 266.9830 216.4947 262.0859 388.2101 457.4625 338.8152 414.3333 351.4982 460.6333 305.2664 361.3081 241.3813 299.7921 223.4930 256.8290	i)	2.7183 7.3891 20.0855 54.5982 148.4132
d)	122.0656 67.7681 41.7748 22.0886 97.9050 110.4400 77.4265 62.7141 31.5228 134.2541 75.8534 174.1713 61.4997 148.9061 20.1201 42.2798	j)	1.0e+41 * 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 4.9235 0.0000 0.0000 0.0000 0.0000
e)	0.0691 -0.0440 0.0231 -0.0249 -0.0705 0.0487 -0.0317 0.0429 -0.0865 0.0816 -0.0375 0.0165 0.0986 -0.0838 0.0554 -0.0384	k)	1 16 81 256 625
f)	0.0650 0.0243 0.0521 0.0830 -0.0539 0.0006 -0.0361 -0.0729 -0.0822 -0.0303 -0.0633 -0.1044 0.1025 0.0355 0.0724 0.1290	l)	10603 9219 7113 6174 15034 16453 11868 12438 12958 17407 11206 13366 11807 10865 8091 7080

Figura 5: Resultado das operações matemáticas realizadas com os vetores e matrizes.

Conforme foi comentado durante a execução das atividades, não há maneira de comparar os resultados obtidos na Figura 5 fazem sentido de acordo com os valores existentes na Figura 1, já que foram produzidas em diferentes momentos. A estratégia de utilizar uma geração aleatória para algumas variáveis poupou tempo para a produção dos códigos, mas prejudicou a interpretação dos resultados. A despeito desses problemas, observa-se que as operações envolvendo o vetor x estão de acordo com o esperado, apresentando fortes indícios que as demais também devem estar corretas.

2.5 Vetores com espaçamento controlado

Em virtude do tamanho do vetor $t2$, não serão apresentados todas as variáveis produzidas, contudo, na Figura 6 estão apresentados os vetores $t1$ e $t3$, para verificar se a função `linspace()` foi suficiente para produzir as variáveis solicitadas.

t1	0	1	2	3	4	5	6	7	8	9	10
t3	-10	-5	0	5	10						

Figura 6: Vetores criados com espaçamento fixo.

Como pode ser verificado na Figura 6, os dois vetores seguem com o esperado pelo enunciado do tutorial, demonstrando que a função `linspace()` está devidamente implementada, bem como a lógica para construção de vetores com espaçamento fixo.

2.6 Manipulação de Matrizes

Os resultados para as manipulações das matrizes são apresentados na Figura 7.

```

A2
    18    71
    49    76

A3
    77    45    27    12
    80    65    68    50
    18    71    66    96
    49    76    16    34
     1     2     3     4

```

Figura 7: Criação de matrizes com elementos previamente existentes.

Pode-se avaliar que as operações foram realizadas com sucesso dada a comparação da matriz $A3$ com a $A2$. Essa última é dada como um segmento da matriz A , que pode ser identificada com sucesso na matriz $A3$, cuja construção é dada apenas pelo incremento dos quatro primeiro elementos de x em uma linha da matriz A .

2.7 Funções e gráficos

Para essa seção, a verificação do funcionamento se dá pela análise do gráfico produzido, que pode ser visualizado na Figura 8.

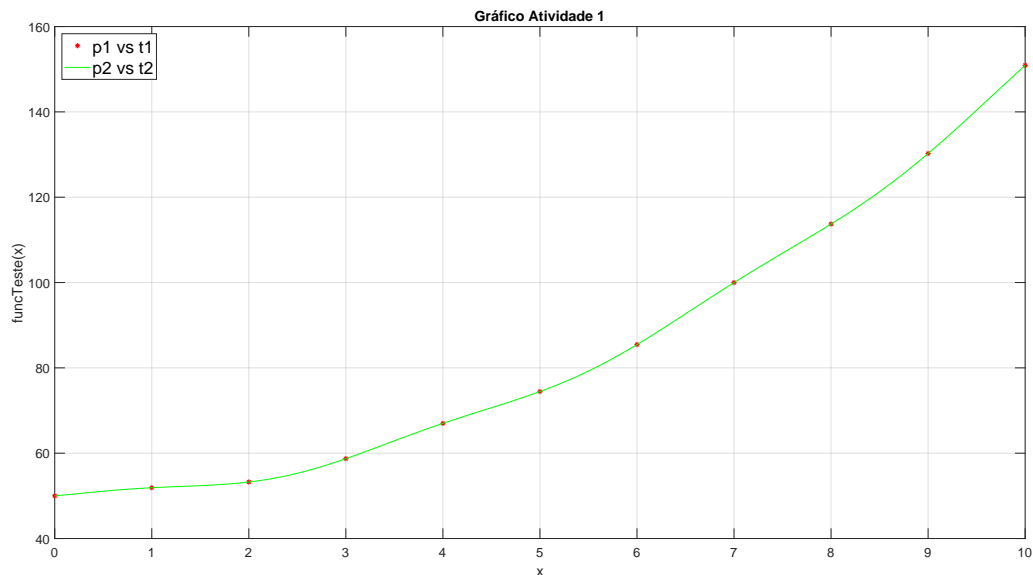


Figura 8: Apresentação da função Teste nos vetores $t1$ e $t2$.

Os vetores $t1$ e $t2$ apresentam os mesmos limites, sendo distintos apenas com relação ao passo utilizado para construção dos mesmos. A função 'Teste' implementada é dada por:

$$f(t) = t^2 + \text{sen}(2t) + 50$$

Espera-se então uma parábola com uma trajetória levemente oscilatória, já que o termo senoidal contribui com um termo limitado entre ± 1 . Assim, pode-se assumir que o comportamento apresentado na Figura 8 segue com o esperado.