

Tim DeChant

ECE 601 - Dr. Gray

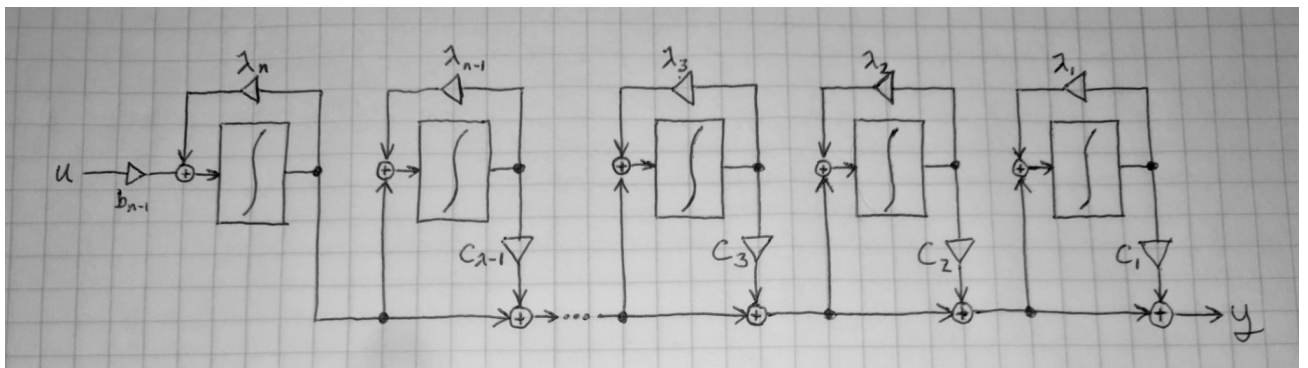
## Homework #4

Due 9/28/2017

### 1. Kailath 2.2-6. Cascade Form

Draw a block diagram corresponding to the realization

$$A = \begin{bmatrix} \lambda_1 & c_2 & c_3 & \cdots & c_{n-1} & 1 \\ & \lambda_2 & c_3 & \cdots & c_{n-1} & 1 \\ & & \lambda_3 & \cdots & c_{n-1} & 1 \\ & & & \ddots & \vdots & \\ & & & & \lambda_{n-1} & 1 \\ & \bigcirc & & & & \lambda_n \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ b_{n-1} \end{bmatrix}, \quad c = [c_1 \quad c_2 \quad c_3 \quad \cdots \quad c_{n-1} \quad 1]$$



**Update:**  $c_{\lambda-1}$  should read  $c_{n-1}$ .

### 2. Kailath 2.2-8. Interconnections of Subsystems

Write state equations for two realizations  $\{A_i, b_i, c_i\}$  connected in:

#### a) Series

$$\begin{aligned} u_1 &= u \\ \dot{x}_1 &= A_1 x_1 + b_1 u_1 = A_1 x_1 + b_1 u \\ u_2 &= y_1 = c_1 x_1 \\ \dot{x}_2 &= A_2 x_2 + b_2 u_2 = b_2 c_1 x_1 + A_2 x_2 \\ y &= y_2 = c_2 x_2 \end{aligned}$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} A_1 & \bigcirc \\ b_2 c_1 & A_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} b_1 \\ \bigcirc \end{bmatrix} u, \quad y = [\bigcirc \quad c_2] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

#### b) Parallel

$$\begin{aligned} u_1 &= u_2 = u \\ y &= y_1 + y_2 = c_1 x_1 + c_2 x_2 \end{aligned}$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} A_1 & \bigcirc \\ \bigcirc & A_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} u, \quad y = [c_1 \quad c_2] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

**c) Feedback, with  $\{A_1, b_1, c_1\}$  in the forward loop and  $\{A_2, b_2, c_2\}$  in the feedback loop**

$$\begin{aligned} u_1 &= u + y_2 = u + c_2x_2 \\ \dot{x}_1 &= A_1x_1 + b_1u_1 = A_1x_1 + b_1c_2x_2 + b_1u \\ u_2 &= y_1 = y = c_1x_1 \\ \dot{x}_2 &= A_2x_2 + b_2u_2 = b_2c_1x_1 + A_2x_2 \end{aligned}$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} A_1 & b_1c_2 \\ b_2c_1 & A_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} b_1 \\ 0 \end{bmatrix} u, \quad y = \begin{bmatrix} c_1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

### 3. Properties of Condition Numbers

Let  $A, B \in \mathbb{R}^{n \times n}$  be arbitrary matrices. (Assume  $A$  is invertible when necessary.) For each statement below, either prove its validity in general or provide a specific counterexample to disprove it.

**(a)  $c(A) \geq 1$**

$$\sigma_1 \geq \sigma_p \rightarrow c(A) = \frac{\sigma_1}{\sigma_p} \geq 1$$

**(b) If  $A^{-1} = A^T$  then  $c(A) = 1$**

$$\begin{aligned} \sigma_i &= \lambda_i^{1/2}(AA^T) = \lambda_i^{1/2}(I) \\ \sigma_i &= 1, \quad \forall i \\ c(A) &= \frac{\sigma_1}{\sigma_p} = 1 \end{aligned}$$

**(c) If  $c(A) = 1$  then  $A^{-1} = A^T$**

False. The condition number does not account for scaling:

In [1]:

```
A = 3*eye(3);
c = cond(A)
A_inv = inv(A)
A_trans = A'
```

c =

1

A\_inv =

```
0.3333      0      0
      0 0.3333      0
      0      0 0.3333
```

A\_trans =

```
3      0      0
0      3      0
0      0      3
```

**(d)**  $c(A^T) = c(A)$

Since singular values are eigenvalues of both  $AA^T$  and  $A^TA$ , this quickly follows:

$$\begin{aligned} \sigma_i(A) &= \lambda_i^{1/2}(AA^T) = \lambda_i^{1/2}(A^TA) = \sigma_i(A^T) \\ c(A) &= \frac{\sigma_1(A)}{\sigma_p(A)} = \frac{\sigma_1(A^T)}{\sigma_p(A^T)} = c(A^T) \end{aligned}$$

**(e)**  $c(A^{-1}) = (c(A))^{-1}$

Since  $c(A^{-1}) \geq 1$ , this is impossible for  $c(A) > 1$ :

```
In [2]: A = [1 2; 3 4];
c_inv = cond(inv(A))
inv_c = 1/cond(A)
```

c\_inv =  
14.9330

inv\_c =  
0.0670

**(f)**  $c(\alpha A) = \alpha c(A), \forall \alpha \in \mathbb{R}$

This will never be true for negative  $\alpha$ , since  $c(\alpha A) \geq 1$  **-and-**  $c(A) \geq 1$ .

```
In [3]: c_neg = cond(-A)
neg_c = -cond(A)
```

c\_neg =  
14.9330

neg\_c =  
-14.9330

**(g)**  $c(A + B) \leq c(A) + c(B)$

False. Summing matrices can easily result in something singular, or at least sensitive.

```
In [4]: B = [4 3; 2 1.00001];
c_sum = cond(A+B)
sum_c = cond(A)+cond(B)
```

c\_sum =  
2.0000e+06

sum\_c =  
29.8664

(h)  $c(AB) \leq c(A)c(B)$

$$\begin{aligned} c(AB) &= \|AB\| \|(AB)^{-1}\| \\ &= \|AB\| \|B^{-1}A^{-1}\| \end{aligned}$$

$$\|AB\| \leq \|A\| \|B\|$$

$$\begin{aligned} c(AB) &\leq \|A\| \|B\| \|B^{-1}\| \|A^{-1}\| \\ c(AB) &\leq (\|A\| \|A^{-1}\|)(\|B\| \|B^{-1}\|) \\ c(AB) &\leq c(A)c(B) \end{aligned}$$

### 4. Singular Value Decomposition

Consider a linear operator  $\mathcal{A}$  represented by the matrix

$$A = \begin{bmatrix} -1 & 2 & 3 \\ 1 & 0 & -1 \\ 1 & 2 & 0 \end{bmatrix}.$$

(a) Compute the singular value decomposition of  $A$ .

```
In [5]: A = [-1 2 3; 1 0 -1; 1 2 0];
[Q1,S,Q2] = svd(A)
```

```
Q1 =

    -0.9379    0.1096   -0.3291
     0.2540   -0.4290   -0.8669
    -0.2362   -0.8966    0.3745
```

```
S =

    3.9798         0         0
         0    2.2609         0
         0         0    0.2223
```

```
Q2 =

     0.2402   -0.6348   -0.7344
    -0.5900   -0.6962    0.4088
    -0.7708    0.3351   -0.5417
```

(b) Are singular values invariant under a similarity transformation? Explain.

No, not generally. A similarity transformation may change the shape or scale of the operation, which would alter the singular values.

```
In [6]: T = [1 2 3; 4 5 6; 7 8 10];
A_ = T*A*inv(T)
S_ = svd(A_)
```

A\_ =

-7.0000	22.0000	-11.0000
-11.0000	50.0000	-26.0000
-17.0000	84.0000	-44.0000

S\_ =

115.0022
2.5461
0.0068

**(c) If your answer to (b) is *yes*, give another representation of the operator having the same singular values. If the answer is *no*, can you provide another type of transformation under which the singular values are always preserved?**

If we limit the transform to pure rotations, we can preserve the singular values; for example, we could apply the scale-less components of the SVD  $T = \overline{Q_1} \overline{\Sigma} \overline{Q_2}^T$  to create SV invariant transform  $\overline{A} = \overline{Q_1} A \overline{Q_2}^T$  :

```
In [7]: [Q1_,ST,Q2_] = svd(T);
A_ = Q1_*A*Q2_'
S_ = svd(A_)
```

A\_ =

-1.0181	0.5855	-1.7172
1.9652	1.4414	-2.0080
-0.5653	1.2843	-2.1751

S\_ =

3.9798
2.2609
0.2223