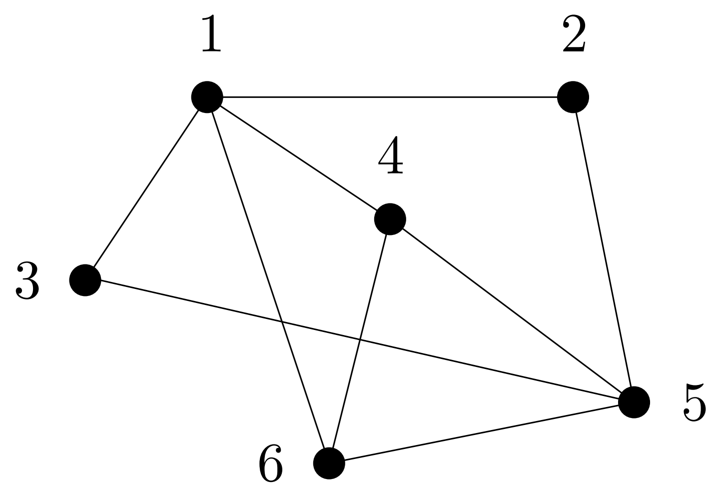
EE263: Introduction to Linear Dynamical Systems
Review Session 7

- synchronizing a communication network
- a method for rapidly driving the state to zero
- square root of a matrix
- logarithm of a matrix

Synchronizing a communication network

- the communication network



- each node has a clock. Clocks run at same speed but are not synchronized

- let x_i be the offset of ith clock with respect to some standard clock (Nodes do not know the offsets)

Process of synchronization

- the problem is to synchronize the clocks (with the standard clock if possible)
- at $t = 1, 2, \ldots$, the neighboring nodes find out their *relative time offset*, $i.e.$, at time t neighbors i, j, find out $x_i(t) - x_j(t)$
- each node *adjusts* its clock by $a_i(t)$, $i.e.$,

$$x_i(t+1) = x_i(t) + a_i(t),$$

where $a_i(t)$ is a function of the relative offsets known to node i at time t

Adjustment algorithm

for each node:

- compute the average of the relative offsets
- add this average to node's current offset

for example, for node 2

$$a_2(t) = \frac{(x_1(t) - x_2(t)) + (x_5(t) - x_2(t))}{2}$$

Questions

what happens ?

- do all the clocks become synchronized with the standard clock?
  ($i.e.$, $x(t) \to 0$ as $t \to \infty$)

- do the clocks become synchronized with each other?
  ($i.e.$, do all $x_i(t) - x_j(t)$ converge to zero as $t \to \infty$)

- does the system become synchronized no matter what the initial offsets are, or only for some initial offsets?

- if the clocks gets synchronized, what is the steady state value?

Solution

for node 2:

$$
\begin{aligned}
x_2(t+1) &= x_2(t) + \frac{1}{2}((x_1(t) - x_2(t)) + (x_5(t) - x_2(t))) \\
&= \frac{1}{2}(x_1(t) + x_5(t))
\end{aligned}
$$

adjustment algorithm gives us the autonomous system,

$x(t+1) = Ax(t)$, where

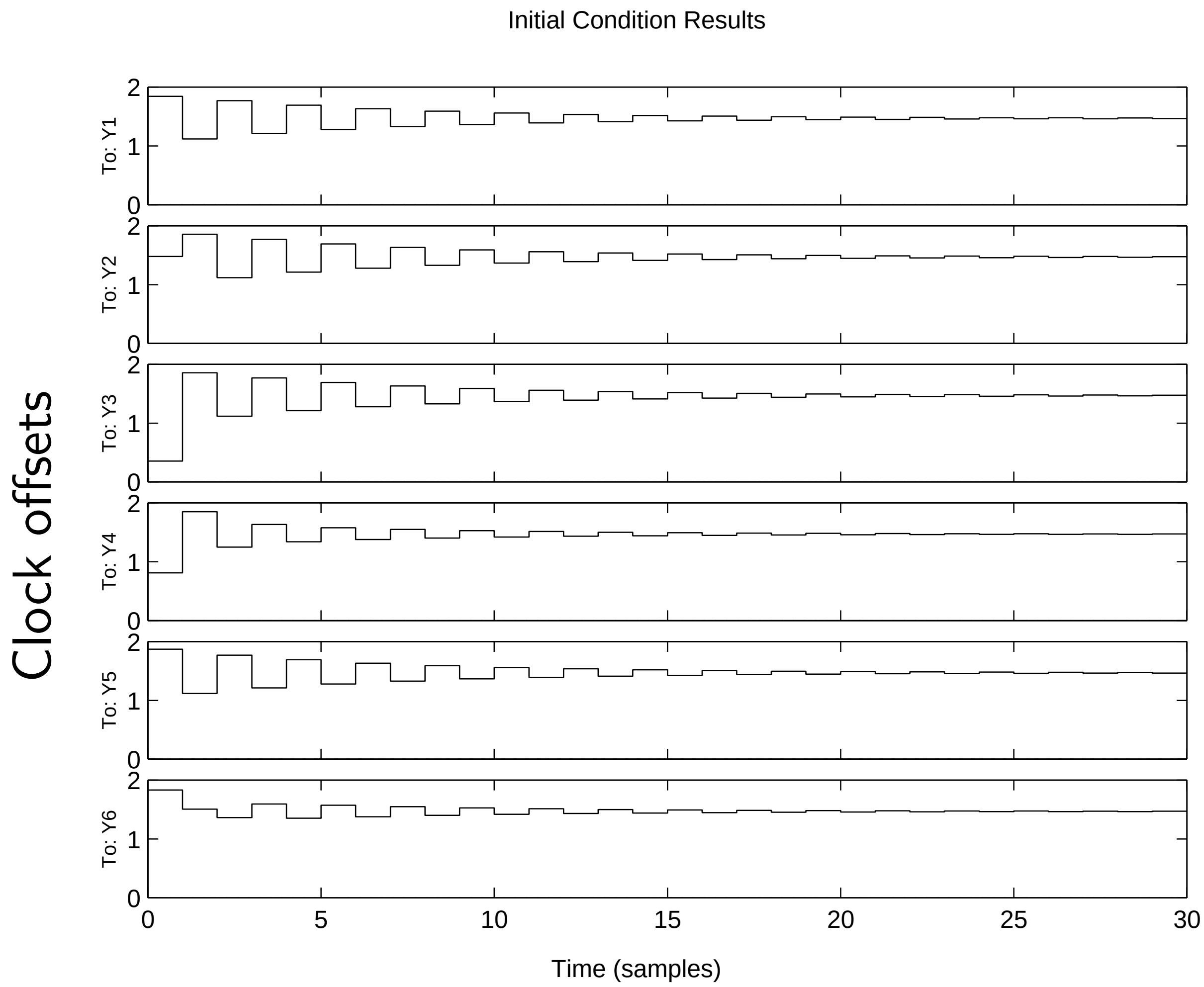
$$
A = \begin{bmatrix}
0 & 1/4 & 1/4 & 1/4 & 0 & 1/4 \\
1/2 & 0 & 0 & 0 & 1/2 & 0 \\
1/2 & 0 & 0 & 0 & 1/2 & 0 \\
1/3 & 0 & 0 & 0 & 1/3 & 1/3 \\
0 & 1/4 & 1/4 & 1/4 & 0 & 1/4 \\
1/3 & 0 & 0 & 1/3 & 1/3 & 0
\end{bmatrix}
$$

```
>> [v,d] = eig(A)

v =
   -0.4333    0.4082   -0.0996    0.7071    0.0629    0.0000
    0.5036    0.4082   -0.5139   -0.0000    0.7043   -0.0000
    0.5036    0.4082   -0.5139   -0.0000   -0.7043   -0.0000
    0.2420    0.4082    0.4754    0.0000   -0.0000    0.7071
   -0.4333    0.4082   -0.0996   -0.7071   -0.0629   -0.0000
    0.2420    0.4082    0.4754    0.0000    0.0000   -0.7071

d =
   -0.8604         0         0         0         0         0
         0    1.0000         0         0         0         0
         0         0    0.1937         0         0         0
         0         0         0    0.0000         0         0
         0         0         0         0         0         0
         0         0         0         0         0   -0.3333
```

the offsets do not go to zero



the clocks do get synchronized with each other

Another algorithm

- compute the average of the relative offsets
- add half of this average to node's current offset

For example for node 2,

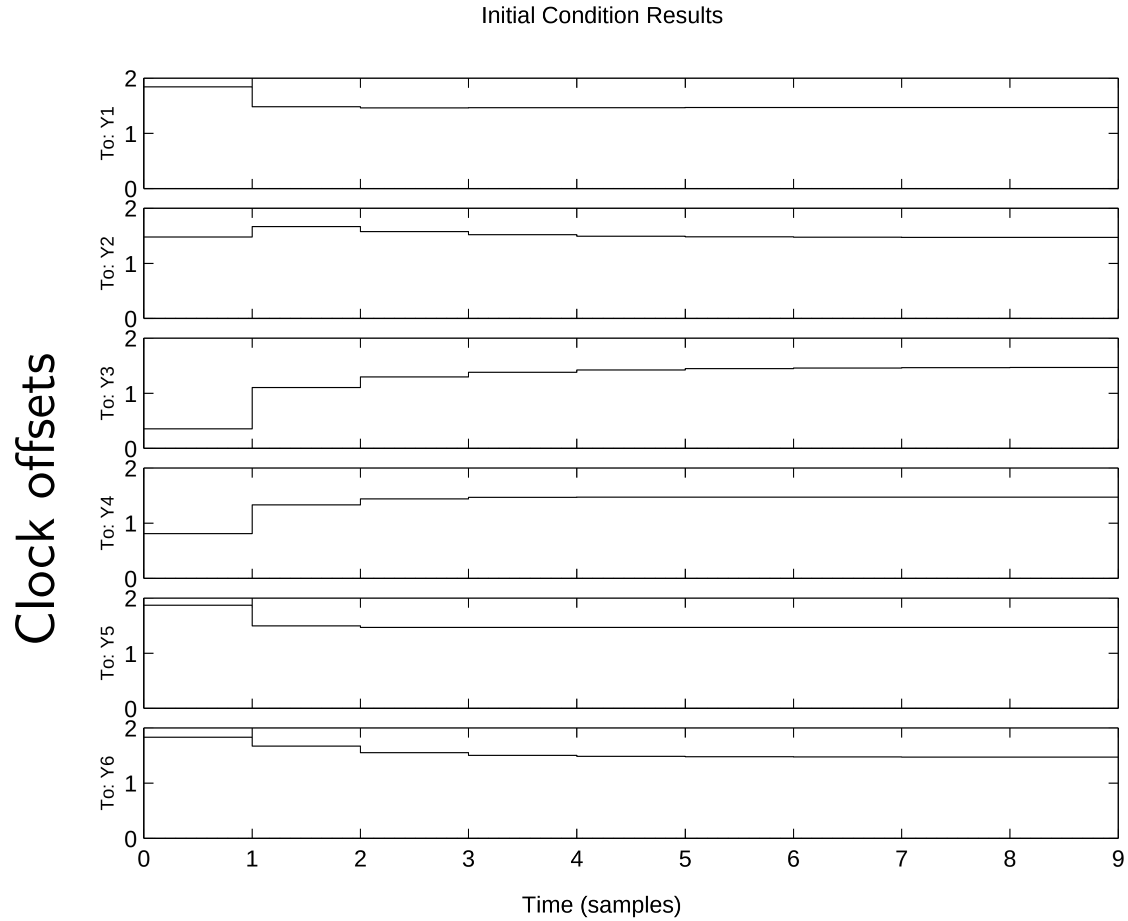
$$a_2(t) = \frac{1}{2}\,\frac{(x_1(t) - x_2(t)) + (x_5(t) - x_2(t))}{2}$$

we have $x(t+1) = Bx(t)$, where $B = (I + A)/2$

```
>> [v,d] = eig(B)
v =
   -0.4333    0.4082   -0.0996   -0.7071    0.3751   -0.0000
    0.5036    0.4082   -0.5139   -0.0000    0.5994    0.0000
    0.5036    0.4082   -0.5139   -0.0000   -0.5994    0.0000
    0.2420    0.4082    0.4754    0.0000    0.0000   -0.7071
   -0.4333    0.4082   -0.0996    0.7071   -0.3751    0.0000
    0.2420    0.4082    0.4754    0.0000   -0.0000    0.7071

d =
    0.0698         0         0         0         0         0
         0    1.0000         0         0         0         0
         0         0    0.5969         0         0         0
         0         0         0    0.5000         0         0
         0         0         0         0    0.5000         0
         0         0         0         0         0    0.3333
```

the offsets do not go to zero



A method for rapidly driving the state to zero.

- we consider the discrete-time linear dynamical system

$$x(t+1) = Ax(t) + Bu(t),$$

where $A \in \mathbf{R}^{n \times n}$ and $B \in \mathbf{R}^{n \times k}$, $k < n$, is full rank

- choose an input u that causes $x(t)$ to converge to zero as $t \to \infty$
- we propose to choose, at time t, $u(t)$ that minimizes $\|x(t+1)\|$: the norm of the state is made as small as possible at every step

does this method work?

we should choose $u(t)$ such that $\|x(t+1)\| = \|Ax(t)+Bu(t)\|$ is minimized. This is simply a least-squares problem in the form $\min_x \|\bar{y} - \bar{A}\bar{x}\|$ where $\bar{y} := Ax(t)$, $\bar{A} := -B$ and $\bar{x} := u(t)$

therefore the minimizing $u(t)$ is

$$u(t) = \bar{x}_{\text{ls}} = (\bar{A}^T\bar{A})^{-1}\bar{A}^T\bar{y} = -(B^TB)^{-1}B^TAx(t)$$

we have

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) \\ &= Ax(t) - B(B^TB)^{-1}B^TAx(t) \\ &= (I - B(B^TB)^{-1}B^T)Ax(t) \end{aligned}$$

therefore, $x(t+1) = Fx(t)$ with $F = (I - B(B^TB)^{-1}B^T)A$

now let us consider a specific case:

$$A = \begin{bmatrix} 0 & 3 \\ 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 1 \end{bmatrix},$$

with A and B as given

$$F = \begin{bmatrix} 0 & 1.5 \\ 0 & -1.5 \end{bmatrix}$$

- the eigenvalues of A are 0 and 0. Thus, $x(t+1) = Ax$ is stable
- the eigenvalues of F are 0 and -1.5. Thus, $x(t+1) = Fx(t)$ is unstable

this method, though reasonable sounding, not only does not rapidly drive the state to zero — it can actually destabilize a stable system!

Square root of a matrix

- let $A \in \mathbf{R}^{n\times n}$ be a diagonalizable matrix, $i.e.$,

  $A = T\Lambda T^{-1}$ with T, $\Lambda \in \mathbf{C}^{n\times n}$ and $\Lambda = \mathbf{diag}(\lambda_1, \ldots, \lambda_n)$

- $B = \sqrt{A}$ if $B^2 = A$

- $B = T\,\mathbf{diag}(\mu_1, \ldots, \mu_n)T^{-1}$, where $\mu_i^2 = \lambda_i$, $i = 1, \ldots, n$

- are these 2^n all the possible square roots?

Logarithm of a matrix

- if $a, b \in \mathbf{R}$, $a > 0$, $b = \log a$ then b is unique
- if $a, c \in \mathbf{C}$, $a \neq 0$, and $e^c = a$ then $c = \log |a| + i(\arg(a) + 2\pi k)$, $k \in \mathbf{Z}$
- let $A \in \mathbf{R}^{n \times n}$ be a diagonalizable nonsingular matrix, *i.e.*,

  $A = T \Lambda T^{-1}$ with T, $\Lambda \in \mathbf{C}^{n \times n}$ and $\Lambda = \mathbf{diag}(\lambda_1, \ldots, \lambda_n)$, $\lambda_i \neq 0$, $i = 1, \ldots, n$
- $B = \log A$ if $e^B = A$
- $B = T \, \mathbf{diag}(\mu_1, \ldots, \mu_n) T^{-1}$, where $e^{\mu_i} = \lambda_i$, $i = 1, \ldots, n$
- if $\lambda_i \in \mathbf{R}$, $\lambda_i > 0$, $i = 1, \ldots, n$ and $B = \log A$ then is B unique?