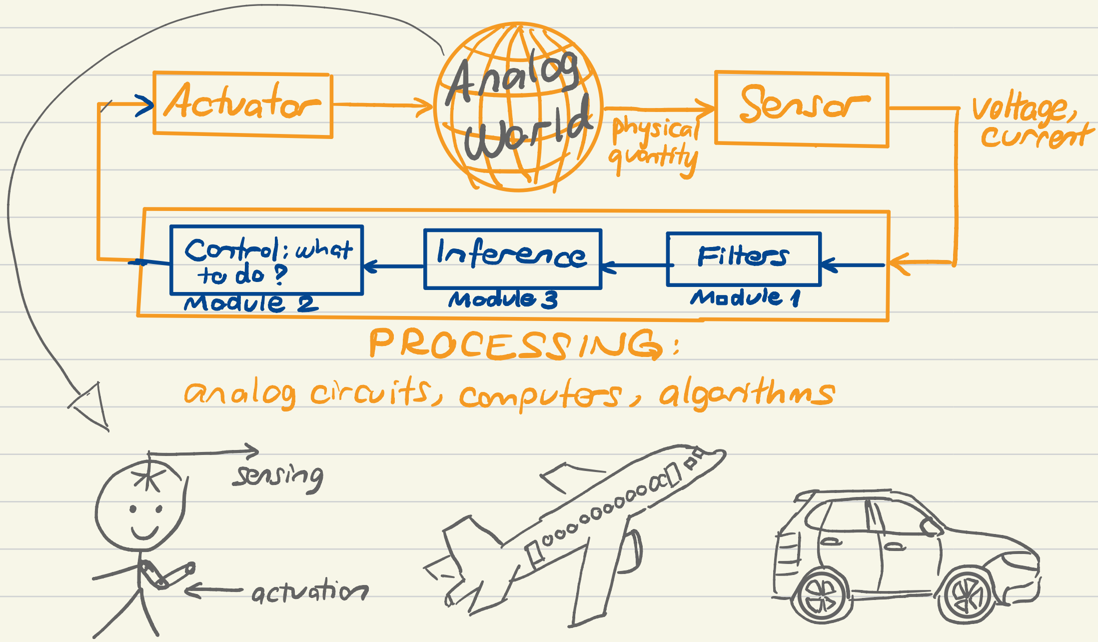
EECS 16B

Spring 2022

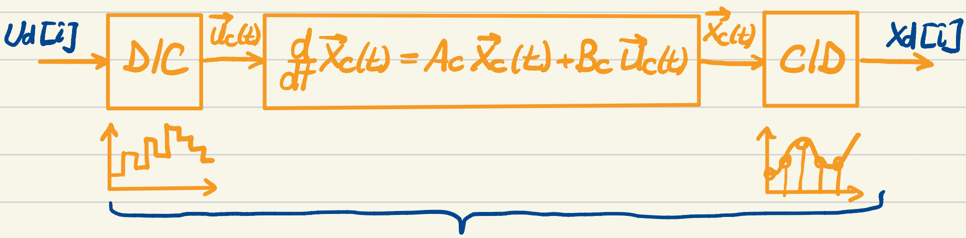
Lecture 28

4/28/2022

LECTURE 28 – Wrap up (part 2)



Discretization:



$$\vec{x}_d[i+1] = A_d x_d[i] + B_d u_d[i]$$

$\vec{x}$: vector of state variables (V_c, I_L in RLC circuits; position, velocity in mechanical systems; concentrations of reactants in a chemical reaction system; flows and pressures in an engine model; etc.)

$\vec{u}$: control input (voltage and current sources in a circuit; forces and torques applied to a mechanical system; etc.)

$$\frac{d}{dt} x_c(t) = \lambda x_c(t) + b u_c(t)$$

sol'n at $t=(i+1)\Delta$ from initial condition $x_c(i\Delta) =: x_d[i]$

$u_c(t) = u_d[i]$ from $t = i\Delta$ to $(i+1)\Delta$

$$\underbrace{x_c((i+1)\Delta)}_{x_d[i+1]} = e^{\lambda\Delta} x_d[i] + \frac{e^{\lambda\Delta}-1}{\lambda}\, b\, u_d[i]$$

(when $\lambda \neq 0$)

$$A_c = V \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix} V^{-1} \Rightarrow A_d = V \begin{bmatrix} e^{\lambda_1\Delta} & & \\ & \ddots & \\ & & e^{\lambda_n\Delta} \end{bmatrix} V^{-1}$$

<u>Stability</u>: when do bounded inputs give bounded states?

Evalue test: $\mathrm{Re}\,\lambda_i(A_c) < 0$ in continuous-time

$|\lambda_i(A_d)| < 1$ in discrete-time

Question: does discretization preserve stability?

Yes. If λ_i is an evalue of A_c, then $e^{\lambda_i\Delta}$ is the corresponding evalue of A_d.

$$e^{\lambda_i\Delta} = e^{(\mathrm{Re}\,\lambda_i + j\,\mathrm{Im}\,\lambda_i)\Delta} = e^{(\mathrm{Re}\,\lambda_i)\Delta} e^{j(\mathrm{Im}\,\lambda_i)\Delta}$$

$$|e^{\lambda_i\Delta}| = |e^{(\mathrm{Re}\,\lambda_i)\Delta}|\,\underbrace{|e^{j(\mathrm{Im}\,\lambda_i)\Delta}|}_{=1}$$

$\mathrm{Re}\,\lambda_i < 0 \Rightarrow e^{(\mathrm{Re}\,\lambda_i)\Delta} < 1$ because $e^{\text{negative \#}} < 1$

<u>Stabilization</u>: $\vec{x}[i+1] = A\vec{x}[i] + Bu[i] + \vec{w}[i]$

$$u[i] = F\vec{x}[i]$$

$$\vec{x}[i+1] = \underbrace{(A+BF)}_{A_{CL}}\vec{x}[i] + \vec{w}[i]$$

Can we assign evalues of A_{CL} arbitrarily with choice of F?

Yes if $\underbrace{[A^{n-1}B, \ldots AB, B]}_{=:\, C_n}$ is invertible.

Same condition for <u>controllability</u>: ability to move from any $\vec{x}[0]$ to any $\vec{x}_{target}$.

$$\vec{x}[l] = A^l\vec{x}[0] + \underbrace{[A^{l-1}B, \ldots, AB, B]}_{=:\, C_l} \begin{bmatrix} u[0] \\ \vdots \\ u[l-1] \end{bmatrix}$$

$$\underbrace{\vec{x}[l]}_{\vec{x}_{target} \in \mathbb{R}^n} - A^l\vec{x}[0] = C_l \begin{bmatrix} u[0] \\ \vdots \\ u[l-1] \end{bmatrix}$$

If $\mathrm{Col}(C_l)$ is $\mathbb{R}^n$ then an input sequence exists to match LHS to RHS.

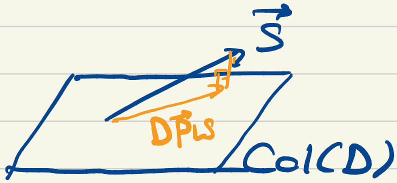
We showed:

If C_l doesn't have n indep. columns when $l=n$, it won't with $l>n$. Thus, controllability test:

$$\mathrm{Col}(C_n) = \mathbb{R}^n \quad \text{i.e. } C_n \text{ invertible.}$$

Module 3 (Inference) started with System ID using <u>LS</u>:

$$\vec{s} = D\vec{p} + \vec{e}$$



LS solution $\vec{p}_{LS}$ matches $D\vec{p}_{LS}$ to projection of $\vec{s}$.

Projection easy when subspace has orthonormal basis $\vec{q}_1, \vec{q}_2, \ldots$

Projection of $\vec{s}$: $\quad \langle \vec{s}, \vec{q}_1 \rangle \vec{q}_1 + \langle \vec{s}, \vec{q}_2 \rangle \vec{q}_2 + \cdots$

$$= \vec{q}_1 \underbrace{\langle \vec{s}, \vec{q}_1 \rangle}_{\vec{q}_1^* \vec{s}} + \vec{q}_2 \underbrace{\langle \vec{s}, \vec{q}_2 \rangle}_{\vec{q}_2^* \vec{s}}$$

$$= \underbrace{QQ^*}_{\text{projection matrix}} \vec{s} \qquad Q = [\vec{q}_1 \; \vec{q}_2 \; \cdots]$$

Orthonormal bases enabled <u>upper-triangularization</u>:

- critical when not diagonalizable
- even if diagonalizable, useful because can upper triangularize with an orthogonal matrix U:

$$U^{-1} A U = U^T A U = T$$

Symmetric matrices can be diagonalized with orthogonal V:

$$V^{-1}AV = V^T A V = \Lambda$$

Spectral Thm. tell us that symmetric matrices admit orthonormal evectors, thus $V = [\vec{v}_1 \cdots \vec{v}_n]$ orthogonal matrix.

<u>SVD</u>: "X ray" of a matrix: $A \in \mathbb{R}^{m \times n}$

$$A = \underbrace{[U_r \; U_{m-r}]}_{U} \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix} \underbrace{[V_r \; V_{n-r}]}_{V}{}^T$$

$$Col(V_{n-r}) = Null(A)$$

$$Col(U_r) = Col(A)$$

$\Sigma_r = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_r \end{bmatrix}$ padded with zeros to become $m \times n$

<u>Pseudoinverse</u>:

$$A^{+} = V \begin{bmatrix} \Sigma_r^{-1} & 0 \\ 0 & 0 \end{bmatrix} U^T$$

$\Sigma_r^{-1} = \begin{bmatrix} \frac{1}{\sigma_1} & & \\ & \ddots & \\ & & \frac{1}{\sigma_r} \end{bmatrix}$ padded with zeros to become $n \times m$

Unifies LS and Min Norm solutions of

$$A\vec{x} = \vec{y}$$

$A^{+} = (A^T A)^{-1} A^T$ when A tall and full column rank:

$A^{+}\vec{y}$ is LS solution.

$A^+ = A^T(AA^T)^{-1}$ when A wide and full row rank:

$$A^+\vec{y} \text{ is Min Norm Solution.}$$

What if A is tall but has dependent columns?

Can't apply usual Least Sq. formula, but

$$A^+\vec{y}$$

is still a LS solution.

Dependent columns mean nontrivial nullspace.
LS solution not unique because if $\vec{x}$ minimizes $\|A\vec{x}-\vec{y}\|$, so does $\vec{x}+\vec{v}$ for any $\vec{v} \in Null(A)$: $A(\vec{x}+\vec{v}) = A\vec{x}$.

In fact $A^+\vec{y}$ is the LS solution with min. norm.

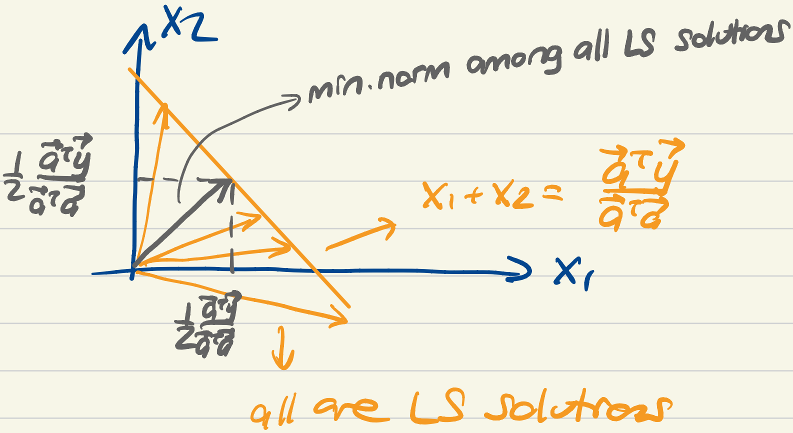
Example: $A = [\vec{a}\ \vec{a}]$ (two identical columns)

$A\vec{x} = \vec{y}$ means:

$$[\vec{a}\ \vec{a}]\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \vec{a}x_1 + \vec{a}x_2 = \vec{a}(x_1+x_2) = \vec{y}$$

now a single column

$$(x_1+x_2)_{LS} = (\vec{a}^T\vec{a})^{-1}\vec{a}^T\vec{y}$$
$$= \frac{\vec{a}^T\vec{y}}{\vec{a}^T\vec{a}}$$



$A^+\vec{y}$ recovers this min. norm LS sol'n:

$$A = [\vec{a}\ \vec{a}] = \vec{a}\,[1\ 1]$$

$$= \underbrace{\sqrt{2}\,\|\vec{a}\|}_{\sigma_1} \cdot \underbrace{\frac{1}{\|\vec{a}\|}\vec{a}}_{\vec{u}_1}\ \underbrace{\frac{1}{\sqrt{2}}[1\ 1]}_{\vec{v}_1^{\,T}}$$

$$A^+ = \frac{1}{\sigma_1}\vec{v}_1\vec{u}_1^{\,T} = \frac{1}{\sqrt{2}\,\|\vec{a}\|}\frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}\frac{1}{\|\vec{a}\|}\vec{a}^T$$

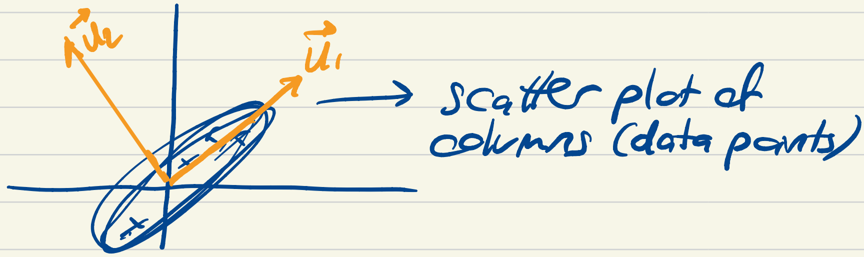
$$= \frac{1}{2\|\vec{a}\|^2}\begin{bmatrix}1\\1\end{bmatrix}\vec{a}^T$$

$$A^+\vec{y} = \frac{1}{2\|\vec{a}\|^2}\begin{bmatrix}1\\1\end{bmatrix}(\vec{a}^T\vec{y}) = \begin{bmatrix}\frac{\vec{a}^T\vec{y}}{2\|\vec{a}\|^2}\\[4pt]\frac{\vec{a}^T\vec{y}}{2\|\vec{a}\|^2}\end{bmatrix}$$

compare to figure at the top

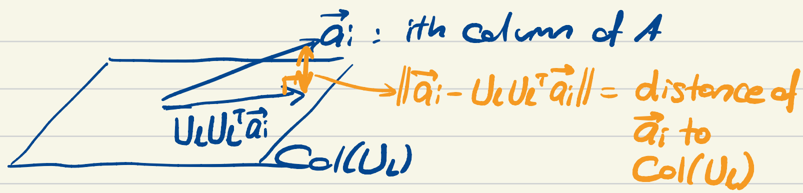
SVD enables low rank approximation of data sets (PCA).

Suppose $m=2$ rows (features) and $n \geq 2$ columns (samples)



Columns condense around $\vec{u}_1$ direction when $\sigma_1 >> \sigma_2$.

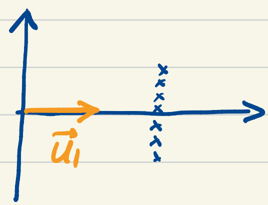
In general, if first ℓ singular values dominant then columns of A condense around $Col(U_\ell)$ where $U_\ell = [\vec{u}_1 \cdots \vec{u}_\ell]$. $Col(U_\ell)$ is the best fit to data among all ℓ-dimensional subspaces in the sense that it minimizes sum of squares of distances of columns to subspace:



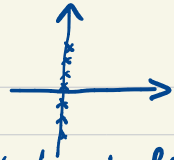
When the data is centered around the origin (we achieve this by subtracting from each column the average of all columns: $\underbrace{\vec{a}_i \to \vec{a}_i - \frac{1}{n}\sum_{j=1}^{n} \vec{a}_j}_{\text{"de-meaning"}}$) then $\vec{u}_1$ is the direction of largest variation in the data: most informative components of the data.

Example (why de-meaning matters):

Suppose data look like this:



Of all 1-dimensional subspaces (lines passing through the origin), the horizontal axis minimizes sum of squares of distances to data points, which is why $\vec{u}_1$ is horizontal in the figure. But the greatest variation in data is in the vertical direction.

Demean so data centered at origin: 

Now the line through the origin that best fits the data is the vertical axis, which matches the direction of largest variation.