ELEC505/ELEC405 Handout #5
Linear System Theory Thursday, December 10, 2020

Homework Set #4

Due: Wednesday, December 23, 2020.

1. *Complex vs. Real*

   Let

   - $V_1 = \mathcal{C}^n$, i.e., the n-dimensional complex vector space, equipped with the inner product
   $$< \mathbf{x}, \mathbf{y} >_C = \mathbf{y}^* \mathbf{x}, \tag{1}$$
   - $V_2 = \Re^{2n}$, i.e., the $2n$-dimensional real vector space, equipped with the inner product
   $$< \mathbf{u}, \mathbf{v} >_R = \mathbf{v}^T \mathbf{u}, \tag{2}$$
   - $\Gamma : \mathcal{C}^n \to \Re^{2n}$ is a mapping between the two vector spaces, where
   $$\Gamma(\mathbf{x}) = \begin{bmatrix} \Re e\{\mathbf{x}\} \\ \mathcal{I}m\{\mathbf{x}\} \end{bmatrix}, \tag{3}$$
   - $\mathbf{A} \in \mathcal{C}^{n \times n}$ define a linear mapping $f : \mathcal{C}^n \to \mathcal{C}^n$, such that for any $\mathbf{x} \in \mathcal{C}^n$
   $$f(\mathbf{x}) = \mathbf{A}\mathbf{x}, \tag{4}$$
   - $g : \Re^{2n} \to \Re^{2n}$ is the mapping corresponding to f such that, for any $\mathbf{x} \in V_1$
   $$g(\Gamma(\mathbf{x})) = \Gamma(f(\mathbf{x})). \tag{5}$$

   If we explicitly write the linear mapping $\mathbf{g}$ as
   $$g(\mathbf{u}) = \mathcal{A}\mathbf{u}, \qquad \forall \mathbf{u} \in \Re^{2n}, \tag{6}$$

   (a) Write $\mathcal{A}$ in terms of $\mathbf{A}$.

   (b) What type of matrix is $\mathcal{A}$, if $\mathbf{A}$ is Hermitian?

   (c) What type of matrix is $\mathcal{A}$, if $\mathbf{A}$ is Unitary?

   (d) What type of matrix is $\mathcal{A}$, if $\mathbf{A}$ is Skew-Hermitian?

   (e) For $\mathbf{x}, \mathbf{y} \in C^n$, write $< \Gamma(\mathbf{x}), \Gamma(\mathbf{y}) >_R$, in terms of $< .,. >_C$.

2. *Cayley Transformation*

   Prove that if $\mathbf{A}$ is skew hermitian then
   $$\mathbf{U} = (\mathbf{I} - \mathbf{A})(\mathbf{I} + \mathbf{A})^{-\mathbf{1}} \tag{7}$$
   is unitary.

3. *Yet another way to obtain a unitary matrix*

   Prove that if $\mathbf{P_V}$ is an orthogonal projection matrix, then $\mathbf{U} = \mathbf{I} - \mathbf{2P_V}$ is unitary.

4. *Rigid Transformations*

   Let $\mathbf{R}(\theta)$ represent a 2×2 rotation matrix

$$\mathbf{R}(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}, \tag{8}$$

   where given a position vector $\mathbf{x} \in \Re^{\mathbf{2}\times\mathbf{1}}$, $\mathbf{R}(\theta)\mathbf{x}$ corresponds to its θ degree counter-clockwise rotated version.

   (a) Find the eigenvalue decomposition for $\mathbf{R}(\theta)$.

   (b) Given $\theta_1, \theta_2 \in \Re$, can we claim that $\mathbf{R}(\theta_\mathbf{1})\mathbf{R}(\theta_\mathbf{2}) = \mathbf{R}(\theta_\mathbf{2})\mathbf{R}(\theta_\mathbf{1})$?

   (c) A rigid transformation is defined in terms of a rotation $\mathbf{R}(\theta)$ and a translation $\mathbf{t} \in \Re^\mathbf{2}$ as

$$\mathbf{y} = \mathbf{R}(\theta)\mathbf{x} + \mathbf{t}. \tag{9}$$

   Note through a vector space extension trick, we can write

$$\begin{bmatrix} \mathbf{y} \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{R}(\theta) & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}}_{\mathbf{T}(\theta,\mathbf{t})} \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix}, \tag{10}$$

   so that the rigid transformation is defined as a matrix multiplication for the extended vectors.

   Find the eigenvalue decomposition for $\mathbf{T}(\theta, \mathbf{t})$.

   (d) Can we claim that $\mathbf{T}(\theta_\mathbf{1}, \mathbf{t_1})\mathbf{T}(\theta_\mathbf{2}, \mathbf{t_2}) = \mathbf{T}(\theta_\mathbf{2}, \mathbf{t_2})\mathbf{T}(\theta_\mathbf{1}, \mathbf{t_1})$ in general? How about for $\mathbf{T}(\theta, \mathbf{t_1})\mathbf{T}(\theta, \mathbf{t_2}) = \mathbf{T}(\theta, \mathbf{t_2})\mathbf{T}(\theta, \mathbf{t_1})$? How about $\mathbf{T}(\mathbf{0}, \mathbf{t_1})\mathbf{T}(\mathbf{0}, \mathbf{t_2}) = \mathbf{T}(\mathbf{0}, \mathbf{t_2})\mathbf{T}(\mathbf{0}, \mathbf{t_1})$? Can you explain your answers based on what is expected from a rigid motion?

5. *Missed Programming huh?... This is simple, don't worry!*

   Generate a 10×10 random complex unitary matrix in Matlab/Python/Julia. Only built-in functions you are allowed to use are *eig* and *randn*. Everytime your code is executed, it is expected to generate a different random unitary matrix. No online code submission is needed: just write down your code on your solution sheet.