Christopher Lum
lum@uw.edu

# Lecture09b
# Similarity Transformation of a Linear System


The YouTube video entitled ‘Similarity Transformation of a Linear Dynamic System’ that covers this lecture is located at https://youtu.be/XMkLNHUmTQM.

## Outline

-Similarity Transformation of a Linear System
- -Changing States
- -Rearranging Order of States
- -Diagonalization: Modal Representation

## Similarity Transformation of a Linear System

Suppose we have a state vector $\overline{x}$ with associated state space model of (see YouTube video entitled ‘State Space Representation of Differential Equations’ at https://youtu.be/pXvAh1IOO4U )

$$\dot{\overline{x}} = A\,\overline{x} + B\,\overline{u} \qquad \textbf{\textit{(Eq.1)}}$$
$$\overline{y} = C\,\overline{x} + D\,\overline{u}$$

What if we are interested in another set of states $\overline{z}$ where the relationship between $\overline{x}$ and $\overline{z}$ is given by the following linear relationship

$$\overline{x} = T\,\overline{z} \qquad \textbf{\textit{(Eq.2.a)}}$$

And by taking a derivative

$$\dot{\overline{x}} = T\,\dot{\overline{z}} \qquad \textbf{\textit{(Eq.2.b)}}$$

Assuming that $T$ is invertible (should be invertible or else the new set of "states" aren't actually states), then substituting in Eq.2.a and Eq.2.b into Eq.1 we obtain

$$T\dot{\bar{z}} = A\,T\,\bar{z} + B\,\bar{u}$$
$$\bar{y} = C\,T\,\bar{z} + D\,\bar{u}$$

$$\dot{\bar{z}} = T^{-1}\,A\,T\,\bar{z} + T^{-1}\,B\,\bar{u}$$
$$\bar{y} = C\,T\,\bar{z} + D\,\bar{u}$$

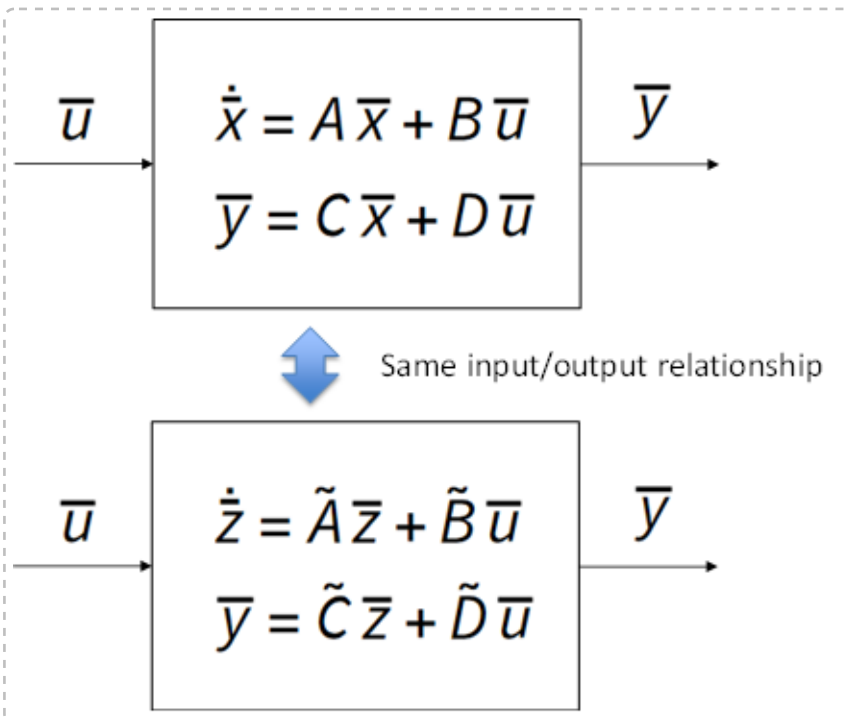
Rewriting this as

$$\dot{\bar{z}} = \tilde{A}\,\bar{z} + \tilde{B}\,\bar{u} \qquad \textbf{(Eq.3)}$$
$$\bar{y} = \tilde{C}\,\bar{z} + \tilde{D}\,\bar{u}$$

where $\tilde{A} = T^{-1}\,A\,T$
$\tilde{B} = T^{-1}\,B$
$\tilde{C} = C\,T$
$\tilde{D} = D$

Note that the $\tilde{A}$ matrix is therefore similar to matrix $A$ but the other matrices are not similar as they are not entire/complete similarity transformations.

These two state space representations have the same input/output relationship




To verify this, we can compute the transfer function matrix of each representation (see YouTube video entitled 'State Space to Transfer Function' at https://youtu.be/NNJ0sUmrKu8 ).

$$G(s) = \tilde{C}\left(s\,I - \tilde{A}\right)^{-1}\tilde{B} + \tilde{D}$$

$$= C\,T\left(s\,I - T^{-1}\,A\,T\right)^{-1}\,T^{-1}\,B + D$$

$$= C\,T\left(s\,T^{-1}\,T - T^{-1}\,A\,T\right)^{-1}\,T^{-1}\,B + D$$

$$= C\,T\left[T^{-1}(s\,I - A\,)\,T\right]^{-1}\,T^{-1}\,B + D \qquad \text{recall:} \qquad (M\,N)^{-1} = N^{-1}\,M^{-1}$$

$$= C\,T[(s\,I - A\,)\,T]^{-1}\left[T^{-1}\right]^{-1}\,T^{-1}\,B + D$$

$$= C\,T[(s\,I - A\,)\,T]^{-1}\,T\,T^{-1}\,B + D$$

$$= C\,T[(s\,I - A\,)\,T]^{-1}\,B + D \qquad \text{recall:} \qquad (M\,N)^{-1} = N^{-1}\,M^{-1}$$

$$= C\,T\,T^{-1}(s\,I - A\,)^{-1}\,B + D$$

$$G(s) = C\,(s\,I - A\,)^{-1}\,B + D$$

So we see the dynamics of the system do not change under a similarity transformation (although we already knew this from property 2 of a similarity transformation which states that the eigenvalues of $A$ and $\tilde{A}$ are the same, see YouTube video entitled ‘Similarity Transformation and Diagonalization’ at https://youtu.be/wvRlvDYDIgw ).

## Changing States

In addition to diagonalizing the system, we can use similarity transformations to translate between different sets of states. For example, consider the set of states associated with a 6 DOF aircraft (***<INSERT LINK TO EOMS VIDEO>***).

$$\overline{x} = \begin{pmatrix} u \\ v \\ w \\ p \\ q \\ r \\ \phi \\ \theta \\ \psi \end{pmatrix} \quad \overline{z}' = \begin{pmatrix} V_a \\ \alpha \\ \beta \\ p \\ q \\ r \\ \phi \\ \theta \\ \psi \end{pmatrix} \quad \overline{z}'' = \begin{pmatrix} V_a \\ \alpha \\ \beta \\ p \\ q \\ r \\ \phi \\ \gamma \\ \psi \end{pmatrix}$$

In a previous homework (hw08, p02 and p03), we looked at how linearizing a non-linear relationships yields the transformation matrix. For example, consider the non-linear transformation between $u$, $v$, $w$ and $V_a$, $\alpha$, $\beta$ (See YouTube video entitled ‘Angle of Attack/Sideslip and the Stability/Wind Axes’ at https://youtu.be/4kaK569ug9Q)

$$V_a = \left(u^2 + v^2 + w^2\right)^{1/2}$$

$$\alpha = \text{atan2}(y, x) = \text{atan2}(w, u)$$

$$\beta = \sin^{-1}\left(\frac{v}{\left(u^2+v^2+w^2\right)^{1/2}}\right)$$

In[∘]:=
```
Va[u_, v_, w_] = (u^2 + v^2 + w^2)^(1/2);
α[u_, v_, w_] = ArcTan[u, w];
(*note that Mathematica ArcTan expects arguments in order of [x,y]*)
β[u_, v_, w_] = ArcSin[v / (u^2 + v^2 + w^2)^(1/2)];
```

We can linearize about the point $u_o$, $v_o$, $w_o$ using (see YouTube video entitled ‘Numerically Linearizing a Dynamic System’ at https://youtu.be/1VmeijdM1qs )

$$\Delta V_a = \frac{\partial V_a(u_o,v_o,w_o)}{\partial u}\,\Delta u + \frac{\partial V_a(u_o,v_o,w_o)}{\partial v}\,\Delta v + \frac{\partial V_a(u_o,v_o,w_o)}{\partial w}\,\Delta w$$

$$\Delta \alpha = \frac{\partial \alpha(u_o,v_o,w_o)}{\partial u}\,\Delta u + \frac{\partial \alpha(u_o,v_o,w_o)}{\partial v}\,\Delta v + \frac{\partial \alpha(u_o,v_o,w_o)}{\partial w}\,\Delta w$$

$$\Delta \beta = \frac{\partial \beta(u_o,v_o,w_o)}{\partial u}\,\Delta u + \frac{\partial \beta(u_o,v_o,w_o)}{\partial v}\,\Delta v + \frac{\partial \beta(u_o,v_o,w_o)}{\partial w}\,\Delta w$$

In[∘]:=
```
replaceString = {u → uo, v → vo, w → wo};

Print["Va"]
D[Va[u, v, w], u] /. replaceString // Simplify
D[Va[u, v, w], v] /. replaceString // Simplify
D[Va[u, v, w], w] /. replaceString // Simplify
Print[" "]

Print["α"]
D[α[u, v, w], u] /. replaceString // Simplify
D[α[u, v, w], v] /. replaceString // Simplify
D[α[u, v, w], w] /. replaceString // Simplify
Print[" "]

Print["β"]
D[β[u, v, w], u] /. replaceString // Simplify
D[β[u, v, w], v] /. replaceString // Simplify
D[β[u, v, w], w] /. replaceString // Simplify
Print[" "]
```

Va

Out[]= $\frac{uo}{\sqrt{uo^2 + vo^2 + wo^2}}$

Out[]= $\frac{vo}{\sqrt{uo^2 + vo^2 + wo^2}}$

Out[]= $\frac{wo}{\sqrt{uo^2 + vo^2 + wo^2}}$

*α*

Out[]= $-\frac{wo}{uo^2 + wo^2}$

Out[]= 0

Out[]= $\frac{uo}{uo^2 + wo^2}$

*β*

Out[]= $-\frac{uo\ vo}{\sqrt{\frac{uo^2+wo^2}{uo^2+vo^2+wo^2}}\ \left(uo^2 + vo^2 + wo^2\right)^{3/2}}$

Out[]= $\frac{\sqrt{\frac{uo^2+wo^2}{uo^2+vo^2+wo^2}}}{\sqrt{uo^2 + vo^2 + wo^2}}$

Out[]= $-\frac{vo\ wo}{\sqrt{\frac{uo^2+wo^2}{uo^2+vo^2+wo^2}}\ \left(uo^2 + vo^2 + wo^2\right)^{3/2}}$

So we have

$$\Delta V_a = \frac{u_o}{\sqrt{u_o{}^2+v_o{}^2+w_o{}^2}}\ \Delta u + \frac{v_o}{\sqrt{u_o{}^2+v_o{}^2+w_o{}^2}}\ \Delta v + \frac{w_o}{\sqrt{u_o{}^2+v_o{}^2+w_o{}^2}}\ \Delta w$$

$$\Delta \alpha = -\frac{w_o}{u_o{}^2+w_o{}^2}\ \Delta u + 0 * \Delta \mathrm{v} + \frac{u_o}{u_o{}^2+w_o{}^2}\ \Delta w$$

$$\Delta\beta = -\frac{u_o\, v_o}{\sqrt{\frac{u_o{}^2+w_o{}^2}{u_o{}^2+v_o{}^2+w_o{}^2}}\ \left(u_o{}^2+v_o{}^2+w_o{}^2\right)^{3/2}}\ \Delta u + \frac{\sqrt{\frac{u_o{}^2+w_o{}^2}{u_o{}^2+v_o{}^2+w_o{}^2}}}{\sqrt{u_o{}^2+v_o{}^2+w_o{}^2}}\ \Delta v - \frac{v_o\, w_o}{\sqrt{\frac{u_o{}^2+w_o{}^2}{u_o{}^2+v_o{}^2+w_o{}^2}}\ \left(u_o{}^2+v_o{}^2+w_o{}^2\right)^{3/2}}\ \Delta w$$

So our transformation matrix can be written as

$$\overline{x} = T\,\overline{z}'$$

$$T^{-1}\bar{x} = \bar{z}'$$

$$\begin{pmatrix} \begin{matrix} \frac{u_o}{\sqrt{u_o^2+v_o^2+w_o^2}} & \frac{v_o}{\sqrt{u_o^2+v_o^2+w_o^2}} & \frac{w_o}{\sqrt{u_o^2+v_o^2+w_o^2}} \\ -\frac{w_o}{u_o^2+w_o^2} & 0 & \frac{u_o}{u_o^2+w_o^2} \\ -\frac{u_o v_o}{\sqrt{\frac{u_o^2+w_o^2}{u_o^2+v_o^2+w_o^2}}\left(u_o^2+v_o^2+w_o^2\right)^{3/2}} & \frac{\sqrt{\frac{u_o^2+w_o^2}{u_o^2+v_o^2+w_o^2}}}{\sqrt{u_o^2+v_o^2+w_o^2}} & -\frac{v_o w_o}{\sqrt{\frac{u_o^2+w_o^2}{u_o^2+v_o^2+w_o^2}}\left(u_o^2+v_o^2+w_o^2\right)^{3/2}} \end{matrix} & \text{zeros}(3,6) \\ \text{zeros}(6,3) & \text{eye}(6) \end{pmatrix} \begin{pmatrix} \Delta u \\ \Delta v \\ \Delta w \\ \Delta p \\ \Delta q \\ \Delta r \\ \Delta \phi \\ \Delta \theta \\ \Delta \psi \end{pmatrix} = \begin{pmatrix} \Delta V_a \\ \Delta \alpha \\ \Delta \beta \\ \Delta p \\ \Delta q \\ \Delta r \\ \Delta \phi \\ \Delta \theta \\ \Delta \psi \end{pmatrix}$$

## Rearranging Order of States

Another application is we can use a similarity transformation to rearrange the state vector.

Consider the original state vector for the aircraft model

$$\bar{x} = \begin{pmatrix} u \\ v \\ w \\ p \\ q \\ r \\ \phi \\ \theta \\ \psi \end{pmatrix} \quad \text{(original state vector)}$$

Suppose we want to reorder the states as follows

$$\bar{z} = \begin{pmatrix} u \\ w \\ q \\ \theta \\ v \\ p \\ r \\ \phi \\ \psi \end{pmatrix} \quad \text{(desired new state vector)}$$

We write the similarity transformation equation of

$$\overline{z} = T^{-1}\,\overline{x}$$

At this point, all we need to do is simply fill in the $T^{-1}$ matrix to make the left side equal the right side. We see this involves filling in with 1's and 0's.

$$\begin{pmatrix} u \\ w \\ q \\ \theta \\ v \\ p \\ r \\ \phi \\ \psi \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u \\ v \\ w \\ p \\ q \\ r \\ \phi \\ \theta \\ \psi \end{pmatrix}$$

So we have

$$T^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Calculating the inverse gives the transformation matrix $T$

In[∘]:=
```
Tinv = {{1, 0, 0, 0, 0, 0, 0, 0, 0},
        {0, 0, 1, 0, 0, 0, 0, 0, 0},
        {0, 0, 0, 0, 1, 0, 0, 0, 0},
        {0, 0, 0, 0, 0, 0, 0, 1, 0},
        {0, 1, 0, 0, 0, 0, 0, 0, 0},
        {0, 0, 0, 1, 0, 0, 0, 0, 0},
        {0, 0, 0, 0, 0, 1, 0, 0, 0},
        {0, 0, 0, 0, 0, 0, 1, 0, 0},
        {0, 0, 0, 0, 0, 0, 0, 0, 1}};

T = Inverse[Tinv];
T // MatrixForm
```

Out[∘]//MatrixForm=

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

We can test this matrix

In[∘]:=
```
z = Tinv.{u, v, w, p, q, r, ϕ, θ, ψ};

z // MatrixForm
```

Out[∘]//MatrixForm=

$$\begin{pmatrix} u \\ w \\ q \\ \theta \\ v \\ p \\ r \\ \phi \\ \psi \end{pmatrix}$$

We will investigate this further in our discussion on longitudinal and lateral/directional models. ***<u><NEED TO INSERT LINK TO VIDEO></u>***

## Diagonalization: Modal Representation

We can diagonalize the $\tilde{A}$ matrix by using $T$ = eigenvectors($A$) as discussed in the previous lecture. We will investigate this in the longitudinal and lateral/directional modes video ***<NEED TO INSERT LINK TO VIDEO>***