ECE 3413 Lab 03
Linear Time Invariant (LTI) system and LTI transfer function

Leomar Durán

20th February 2023

1 Introduction

The purpose of this experiment is to apply the concept of transfer functions to a translational mechanical system.

The principles used in electrical engineering are shared with other disciplines of engineering. We can apply these principles to help us look at everyday problems such as a translational system differently, or we can use this familiarity to better understand the effect of a transfer function in a control system.

This lab also introduces a new type of transfer function, the state-space model which is represented by the `ss` class in Matlab.

2 Procedure

2.1 A translational mechanical system

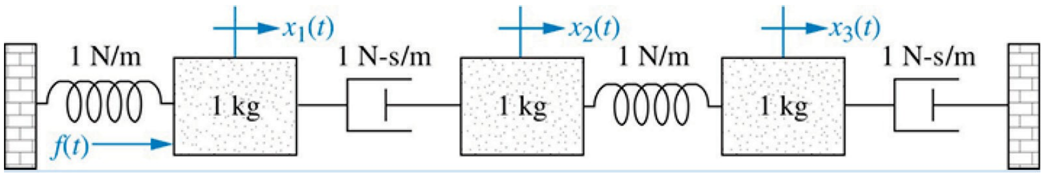


Figure 1: Diagram of the translational mechanical system.

A state-space representation is a mathematical model of a physical system using variables that are in the time domain.

We are given the translational mechanical system in Fig. 1 that outputs $x_3(t)$. From this system, let's define masses

$$\vec{M} := \begin{bmatrix} M_1 & M_2 & M_3 \end{bmatrix}^\mathsf{T} = \begin{bmatrix} 1\,\mathrm{kg} & 1\,\mathrm{kg} & 1\,\mathrm{kg} \end{bmatrix}^\mathsf{T}, \tag{1}$$

spring constants

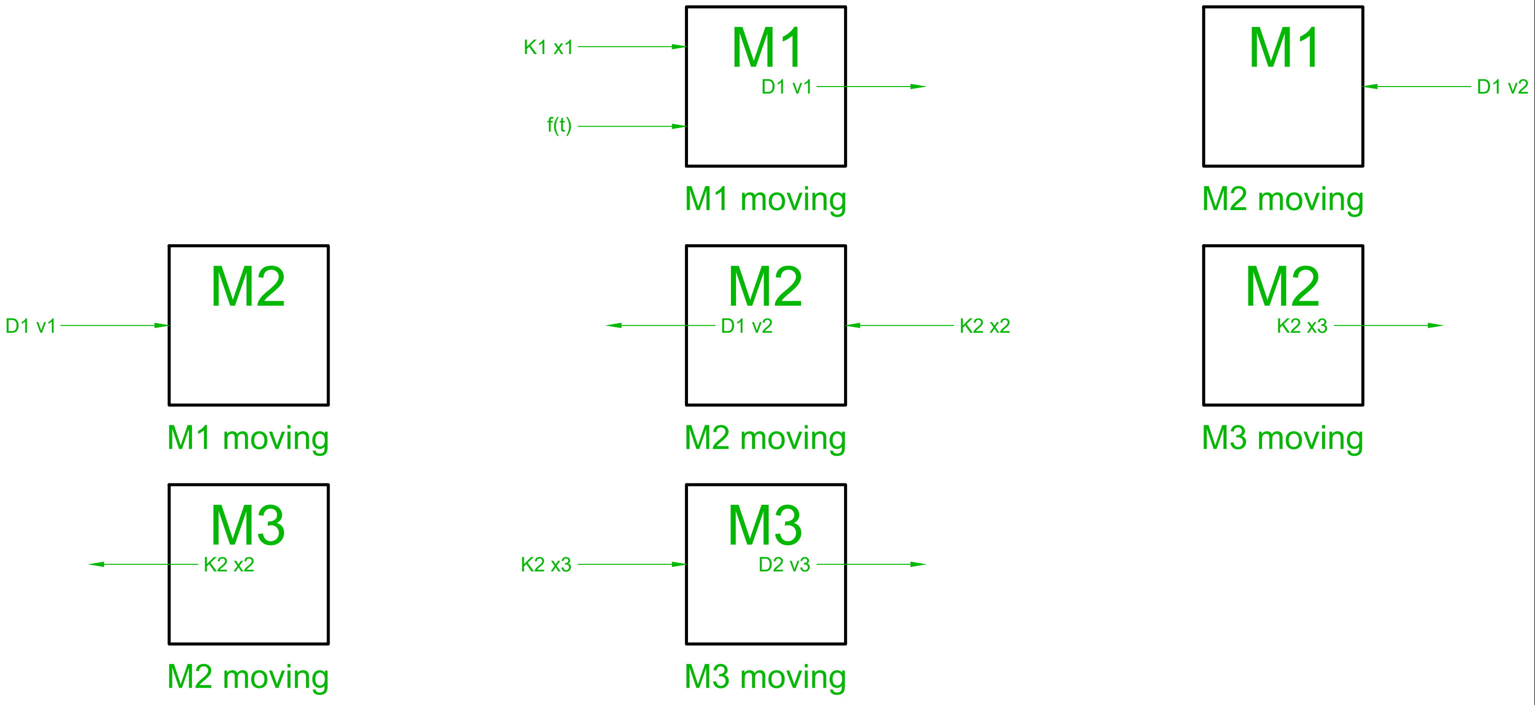
$$\vec{K} := \begin{bmatrix} K_1 & K_2 \end{bmatrix}^\mathsf{T} = \begin{bmatrix} 1\,\mathrm{N/m} & 1\,\mathrm{N/m} \end{bmatrix}^\mathsf{T}, \tag{2}$$

and damping constants

$$\vec{D} := \begin{bmatrix} D_1 & D_2 \end{bmatrix}^\mathsf{T} = \begin{bmatrix} 1\,\mathrm{N\,s/m} & 1\,\mathrm{N\,s/m} \end{bmatrix}^\mathsf{T}. \tag{3}$$

2.1.1 Free body diagram

This system has the free body diagram shown in the CAD drawing on the following page.



UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES TOLERANCES .X ±.010 ANGULAR ±2° .XX ±.005 FRACTIONS: ±1/84" .XXX ±.001 REMOVE BURRS & SHARP EDGES	Temple University
DRAWN BY: L.D.	PART NAME: Lab 03-01 translational mechanical system
CHECKED BY: TBD	SIZE A · Course Name: ECE 3413 · LAB NO: 3 · Sheet: 1 of 1
	Material: Unspecified · Scale: Not to scale · Date: 2023-02-16

2.1.2 Mathematical state-space representation

Then we find the total force at each mass applying superposition for each possible state.

$$\begin{cases} M_1\ddot{x}_1 = \sum F_{M1\text{x}} = K_1x_1 + f(t) + D_1(\dot{x}_1 - \dot{x}_2), \\ M_2\ddot{x}_2 = \sum F_{M2\text{x}} = D_1(\dot{x}_1 - \dot{x}_2) + K_2(-x_2 + x_3), \\ M_3\ddot{x}_3 = \sum F_{M3\text{x}} = K_2(-x_2 + x_3) + D_2\dot{x}_3. \end{cases} \tag{4}$$

Next, we relate states $\underline{\hat{x}} \in \mathbb{R}^6$ to position as in Table 1.

Table 1: How states relate to the position variables

state	position
$\hat{x}_1 =$	$\dot{x}_1,$
$\hat{x}_2 =$	$x_1,$
$\hat{x}_3 =$	$\dot{x}_2,$
$\hat{x}_4 =$	$x_2,$
$\hat{x}_5 =$	$\dot{x}_3,$
$\hat{x}_6 =$	$x_3.$

Thus, we can the derivatives of the states to the states with the output $y := x_3 = \hat{x}_6$ represented by the first row and using the force equations for even rows, representing accelerations, and the state–position relations for odd rows after the first, representing velocities, giving the state-space representation

$$\overset{\left[\begin{array}{c} y \\ \hline \dot{\underline{\hat{x}}} \end{array}\right] =}{\left[\begin{array}{c} y \\ \hline \dot{\hat{x}}_1 \\ \dot{\hat{x}}_2 \\ \dot{\hat{x}}_3 \\ \dot{\hat{x}}_4 \\ \dot{\hat{x}}_5 \\ \dot{\hat{x}}_6 \end{array}\right]} = \overset{\left[\begin{array}{c|c} D & \mathbf{C} \\ \hline \vec{B} & \mathbf{A} \end{array}\right] =:}{\left[\begin{array}{c|cccccc} 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \hline {}^{1}/_{M_1} & +{}^{D_1}/_{M_1} & {}^{K_1}/_{M_1} & -{}^{D_1}/_{M_1} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & +{}^{D_1}/_{M_2} & 0 & -{}^{D_1}/_{M_2} & -{}^{K_2}/_{M_2} & 0 & +{}^{K_2}/_{M_2} \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -{}^{K_2}/_{M_3} & {}^{D_2}/_{M_3} & +{}^{K_2}/_{M_3} \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{array}\right]} \overset{\left[\begin{array}{c} f \\ \hline \underline{\hat{x}} \end{array}\right] =}{\left[\begin{array}{c} f \\ \hline \hat{x}_1 \\ \hat{x}_2 \\ \hat{x}_3 \\ \hat{x}_4 \\ \hat{x}_5 \\ \hat{x}_6 \end{array}\right]}. \tag{5}$$

2.1.3 State-space representation in Matlab

We can use the Matlab function `ss` to create a state-space representation object. This function accepts the matrices of coefficients $\mathbf{A}$, $\vec{B}$, $\mathbf{C}$ and D and returns a transfer function in state-space representation form.

This is a transfer function, just like the rational of polynomials `tf` and zero/pole/gain form `zpk`. The 3 types of objects are mutually convertible, by using the target class's function on any transfer function object.

2.2 Rational of polynomials form

2.2.1 Conversion using Matlab

As stated in the previous section, we may use the `tf` to convert from the state-space representation to the rational of polynomials form just as we have with the zero/pole/gain form. This gives us the transfer function

$$T(s) = \left(\frac{x_3}{F}\right)(s). \tag{6}$$

2.2.2 Equation for the transfer function

Given a translational mechanical system with m masses in movement, we may find the transfer function using the equation

$$T(s) = \mathbf{C}\big(s\mathbf{I}_{(2m)} - \mathbf{A}\big)\vec{B} \tag{7}$$

We use Matlab to calculate the result and compare it to the converted transfer function.

3 Results

Part 1 – Translational mechanical system

As a state-space model

is represented by the matrices of coefficients

```
Mssr =

  A =
       x1  x2  x3  x4  x5  x6
   x1   1   1  -1   0   0   0
   x2   1   0   0   0   0   0
   x3   1   0  -1  -1   0   1
   x4   0   0   1   0   0   0
   x5   0   0   0  -1   1   1
   x6   0   0   0   0   1   0

  B =
       u1
   x1   1
   x2   0
   x3   0
```

```
     x4   0
     x5   0
     x6   0

   C =
         x1  x2  x3  x4  x5  x6
     y1   0   0   0   0   0   1

   D =
         u1
     y1   0

 Continuous-time state-space model.
```

Part 2 – Rational of polynomials form

1. Conversion using Matlab

The transfer function from the state-space representation

```
Mtf =

                       -s
   -------------------------------------------------
   s^6 - s^5 - s^4 - 2 s^3 + 2 s^2 + 2 s + 2.053e-16

 Continuous-time transfer function.
```

2. Equation for transfer functions

```
T =
                 -s - 1.307e-16
  -------------------------------------------------
  s^6 - s^5 - s^4 - 2 s^3 + 2 s^2 + 2 s - 2.435e-16

Continuous-time transfer function.
```

Coefficient of determination R^2

To compare the transfer functions, let's find the R^2 value of all coefficients.

Then the R^2

```
R2 = 1.0000

shows that the coefficients have a strong correlation

and are therefore equivalent.
```

4 Discussion

After finding the state-space representation, this experiment becomes very straightforward. However, that is the difficult part.

What I remembered to help me finish it is that springs act on displacement into the moving mass and viscous dampers aft on velocity away from the moving mass. After this, the sum of all forces must equal the acceleration of the mass. Then setting up the matrix of coefficients is not difficult although I may not have been able to think of coming up with the variables myself. It's a really clever way of solving this problem.

This experiment shows application not only in electrical engineering, but also mechanical and civil engineering of the concept of a control system. It's interesting to imagine how many of the concepts that we have learned in electrical engineering may be applied to other engineering disciplines or may have even come from methods in other engineering disciplines.

Often times, it seems that less intuitive techniques may make a problem much easier to solve simply by changing the domain or a basis. This is the principle behind using the Laplace transform to handle transfer functions more easily.

A Appendix

A.1 Part 1 – State-space representation, Matlab Live Script

```
% part01_translational_mechanical_system_mlx.m

%% Part 1 $-$ Translational mechanical system

clear

% masses
M1 = 1; % [kg]
M2 = 1; % [kg]
M3 = 1; % [kg]
% spring constants
K1 = 1; % [N/m]
K2 = 1; % [N/m]
% damping constants
D1 = 1; % [Ns/m]
D2 = 1; % [Ns/m]

%% As a state-space model
% is represented by the matrices of coefficients
Mssr = ss( ...
    [   [+D1 K1 -D1   0  0   0]/M1;
           1  0   0   0  0   0    ;
        [+D1  0 -D1 -K2  0 +K2]/M2;
           0  0   1   0  0   0    ;
        [  0  0   0 -K2 D2 +K2]/M3;
           0  0   0   0  1   0
    ], ...
    [1 0 0 0 0 0]', ...
    [      0  0   0   0  0   1     ], ...
    0 ...
)
```

A.2 Part 2 – Rational of polynomials form, Matlab Live Script

```
% part02_ratio_of_polynomials_form_mlx.m

%% Part 2 $-$ Rational of polynomials form

% use Mssr the state-space representation from part 01

%% 1. Conversion using Matlab
% The transfer function from the state-space representation
Mtf = tf(Mssr)

%% 2. Equation for transfer functions

% create a transfer function that is just s
s = tf([1 0], 1);
T = Mssr.C*(s*eye(size(Mssr.A)) - Mssr.A)^-1*Mssr.B

%% Coefficient of determination $R^2$
% To compare the transfer functions, let's find the $R^2$ value of all
% coefficients.

% calculate the needed padding
MtfMinusTNum = (numel(T.Num{1}) - numel(Mtf.Num{1}));
MtfMinusTDen = (numel(T.Den{1}) - numel(Mtf.Den{1}));
% zero pad the numerators if necessary
MtfNumPadded = [ zeros(1, MtfMinusTNum), Mtf.Num{1} ];
TNumPadded = [ zeros(1, -MtfMinusTNum), T.Num{1} ];
% zero pad the denominators if necessary
MtfDenPadded = [ zeros(1, MtfMinusTDen), Mtf.Den{1} ];
TDenPadded = [ zeros(1, -MtfMinusTDen), T.Den{1} ];

% group the terms together
MtfTerm = [MtfNumPadded MtfDenPadded];
TTerm = [TNumPadded TDenPadded];

%%
```

```
% Then the R^2
R2 = prod(corrcoef(MtfTerm, TTerm), 'all')^2
assert(abs(1 - R2) < 0.05, ...
    ['Unexpectedly low correlation between converted and calculated' ...
     ' transfer functions.'])
disp('shows that the coefficients have a strong correlation')
disp('and are therefore equivalent.')
```