University Badji Mokhtar-Annaba
Department of Electronics
3rd year License in Automatique L3 (S5)
Academic year: 2025/2026

TP on Linear Systems Control (CSL)

TP3 : The state representation in canonical forms

Introduction :

The transfer function has the advantage of being simple to use, but this simplicity is lost in the case of multivariable transfer functions. Moreover, initial conditions are not easily taken into account, and only the observable and controllable parts are represented. Despite everything, frequency representations, at the base of these representations, provide an irreplaceable view of the external behaviors of systems.

We will present in this TP the modeling of linear systems using another approach called state representation, which allows for the modeling of a dynamic system using state variables.

This representation allows us to determine the state of the system at any future time if we know the state at the initial time and the behavior of the exogenous variables that influence the system. The state representation of the system allows us to know its "internal" behavior and not just its "external" behavior as is the case with its transfer function.

State equations:

Many physical processes can be described by differential and algebraic equations. The state representation for a linear and time-invariant system has the following general form:

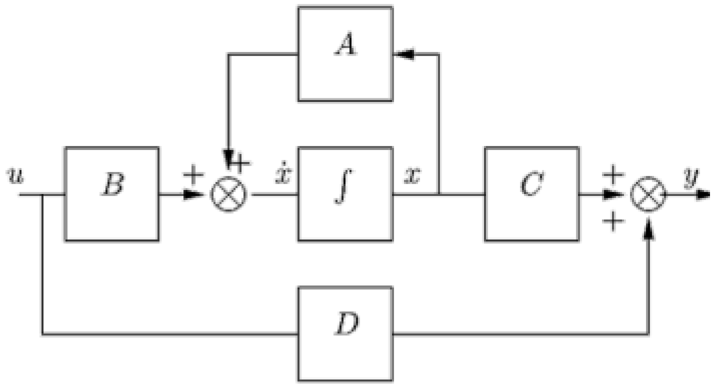
$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases} \qquad x(t_0) = x_0$$

with

— $A(t) \in R^{n\times n}$ est la matrice dynamique,
— $B(t) \in R^{n\times m}$ est la matrice commande ou d'entrée,
— $C(t) \in R^{r\times n}$ est la matrice mesure ou de sortie,
— $D(t) \in R^{r\times m}$ est la matrice de transmission directe.

If matrices A, B, C and D are constant, the system is Linear Time Invariant (LTI). The state representation can be associated with the model and block diagram below:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases} \qquad x(t_0) = x_0$$



Note: The state representation is not unique. This means that for the same physical system, the choice of the state vector is not unique since it depends on the basis in which it is expressed. In the case of state-space modeling, the type of object used is the ss (state-space) type. The associated function, ss(), takes as arguments the 4 matrices defining the LTI system. This will be defined by the command

```
ss(A,B,C,D)
```

Transfer matrix:

Due to the linearity of the Laplace operator, it is possible to apply it to matrices:

$$\begin{cases} pX(p) = AX(p) + BU(p) \Longleftrightarrow X(p) = (pI - A)^{-1} BU(p) \\ Y(p) = CX(p) + DU(p) \end{cases}$$

the initial conditions being considered null since it is about determining G(p). Obviously, this leads to:

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

When studying a system, it is often practical to switch from one representation to another depending on the objectives. Indeed, transfer functions and state representation are two formalisms that have their own analysis tools and methods for synthesizing control laws.

```
% fonction de transfert ⟶ repré-
sentation d'état
G=tf([1 2],[1 3 2])
ss(G)
```

```
% représentation d'état ⟶ fonc-
tion de transfert
A=[-1 4;0 -2];B=[2;3];
C=[1 0];D=0;
sys=ss(A,B,C,D)
G=tf(sys)
```

1-Transition from state space to transfer function

By using the pre-established Matlab function (ss2tf), we obtain:

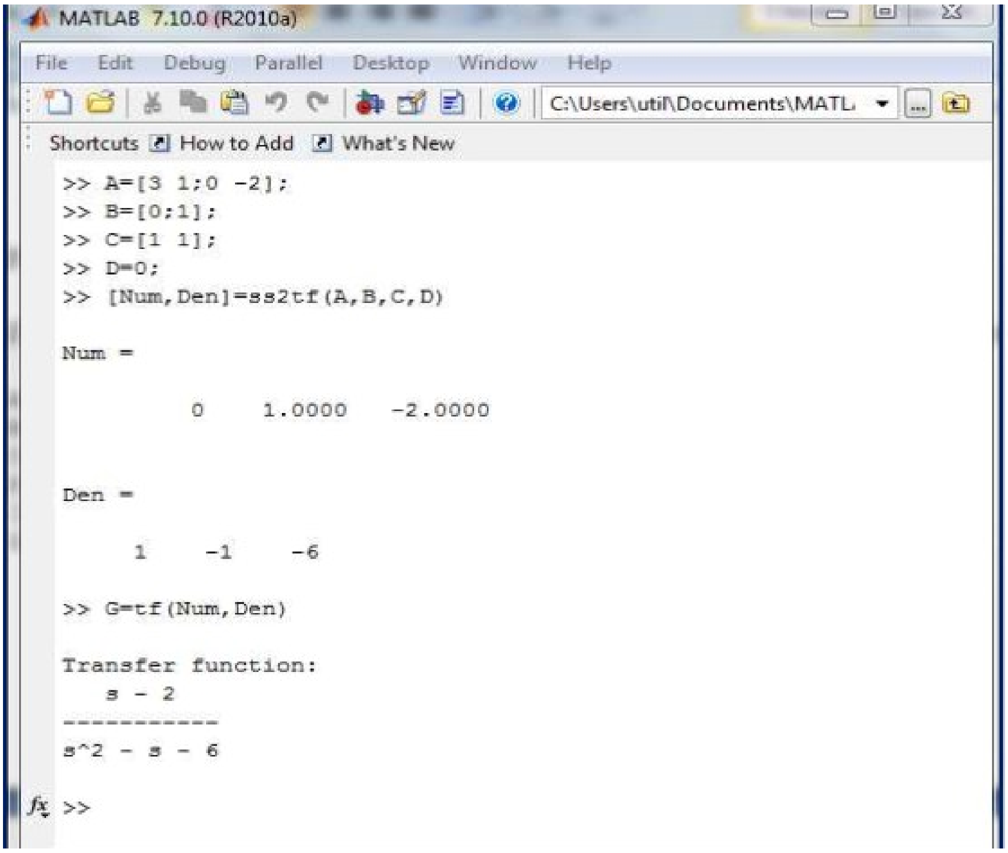


Figure : transition from state representation to transfer function under Matlab

2- Transition from the transfer function to the state space

By using Matlab's pre-established function (tf2ss), we obtain the state representation. Matlab by default only provides a single representation.

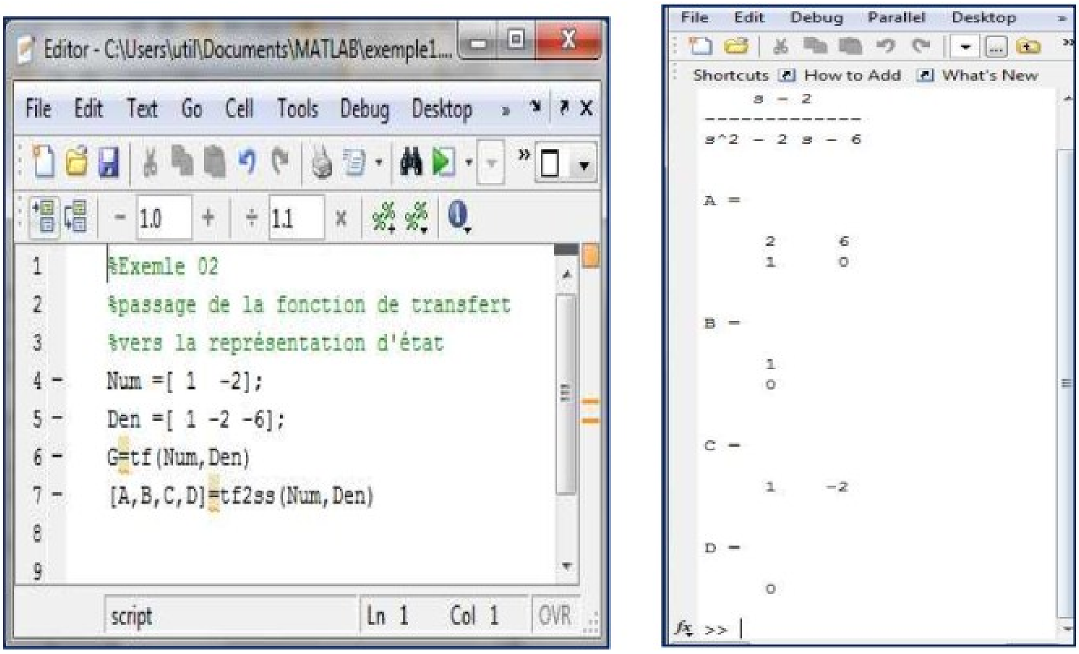
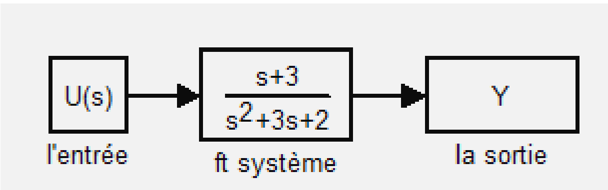


Figure 1.4 : Transition from the transfer function to the state representation under Matlab

Example

We consider the system given by the following transfer function

$$\frac{Y(s)}{U(s)} = \frac{s+3}{s^2+3s+2}$$



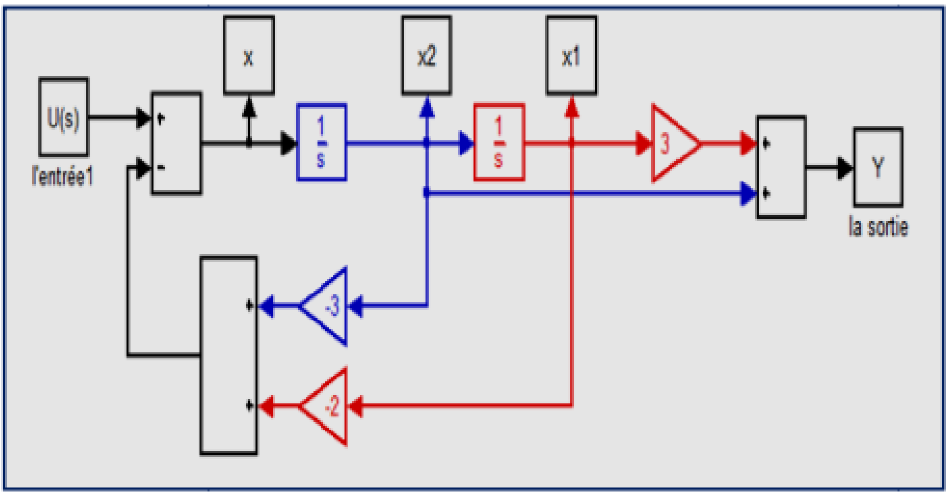
We can obtain the three canonical forms of representation.

1- Commandable canonical form

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

$$y = \begin{bmatrix} 3 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

1-



The functional diagram of the controllable form (Matlab/Simulink)

Under Matlab/Simulink, another way to schematize a system is thru its state representation given by the figure where the state-space block is in the Simulink/continuous library by modifying the matrices A, B, C, D according to the chosen canonical form.

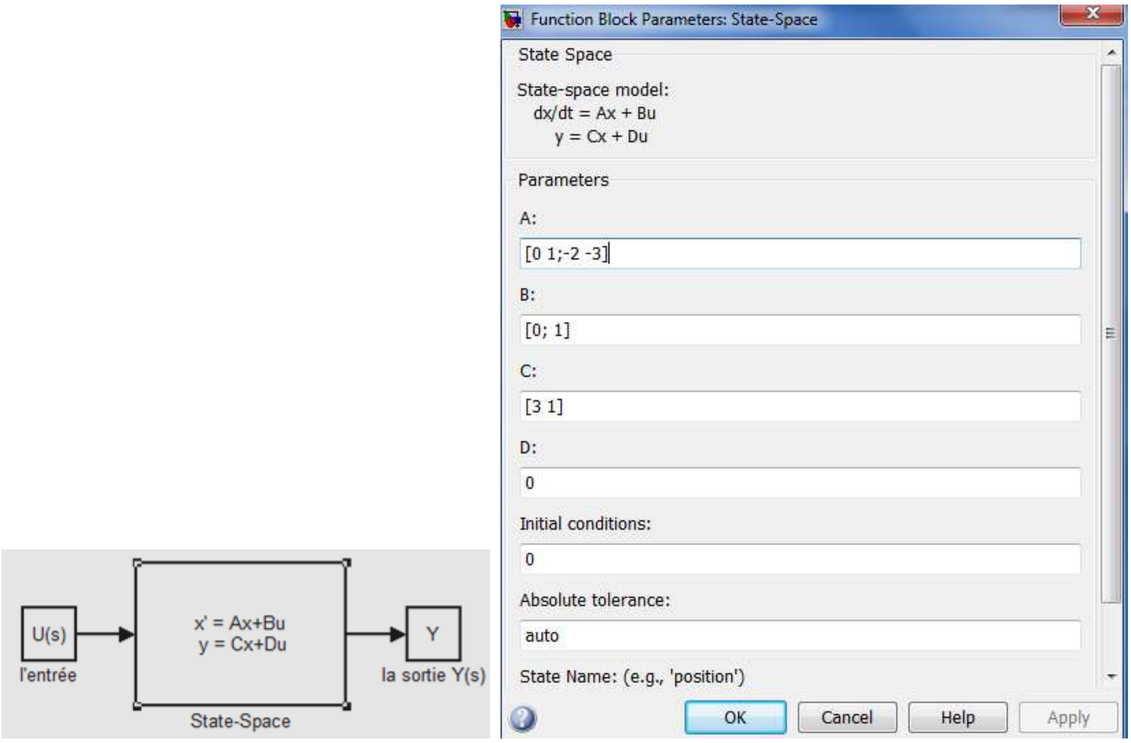
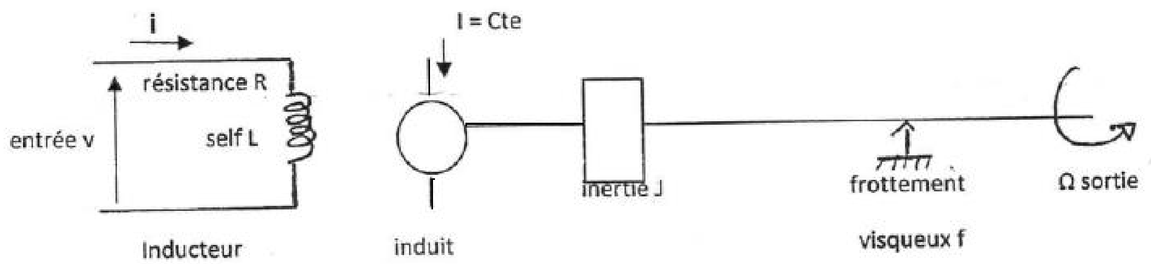


Figure The diagram of the controllable form

Application: The chosen model is a voltage-controlled direct current motor, assuming that the system operates in a linear regime (motor torque is proportional to the inductor current i(t), which particularly assumes that the induced current is constant)



The differential equations governing the behavior of the motor are: The electrical equation:
The electrical equation:

$$v(t) = Ri(t) + \frac{Ldi(t)}{de}$$

L'équation du couple :

$$\Gamma(t) = Ki(t)$$

Une équation mécanique :

$$\Gamma(t) = f.\Omega(\mathrm{t}) + \frac{Jd\Omega(\mathrm{t})}{dt}$$

Using the appropriate MATLAB functions, write the program that will allow you to establish the state model of the DC motor.