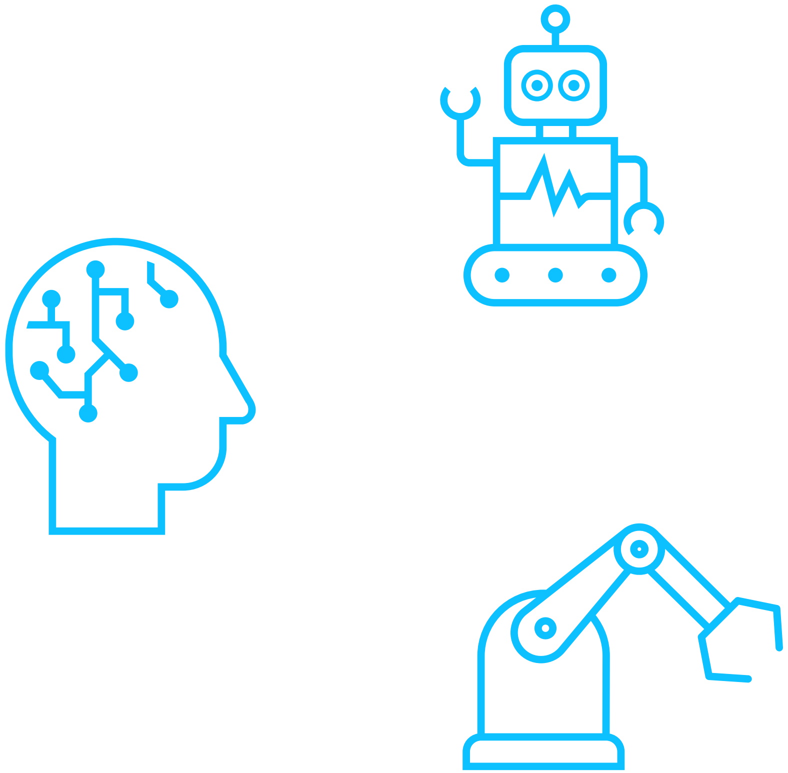
Linear Dynamical Systems

A Review…

{Learn, Create, Innovate};

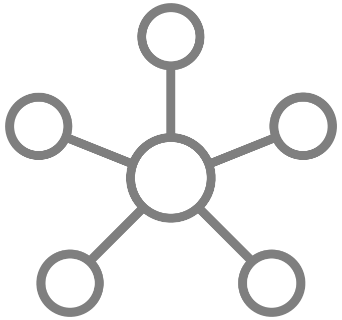
Systems: general aspects

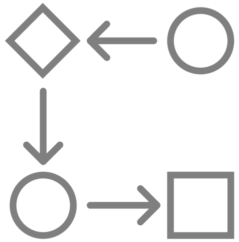
- In general, the notion of a **system** is used in many fields of activity, to delimitate a form of existence in a well-defined space.
- Some examples of systems: the democratic system, the education system of a country, the nervous system, the automatic temperature regulation system, a mobile robot, a robotic manipulator, etc.



Systems: general aspects

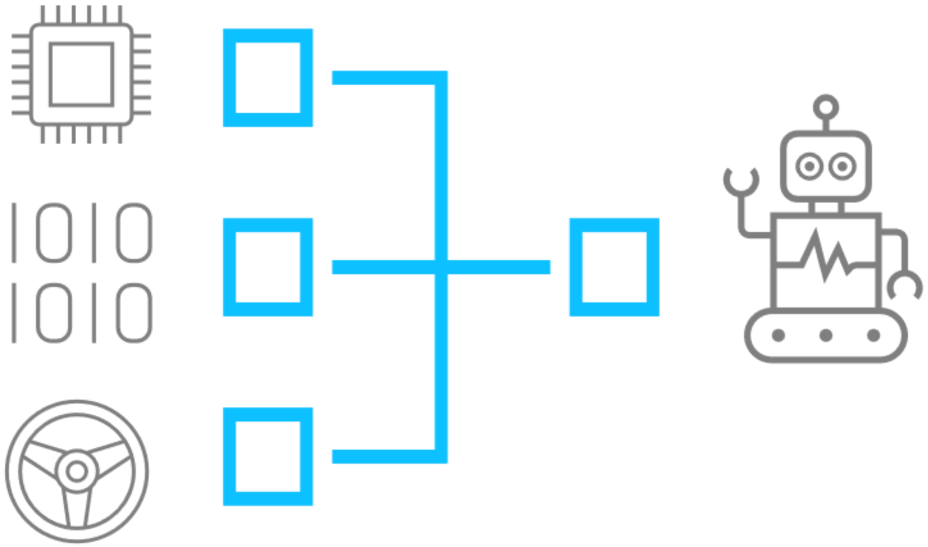
- The notion of system helps us, in a first instance, to delimitate, for example:
  - A state management mechanism
  - A way of education at the national level
  - Part of the components that contribute to the integration of the human body into the environment
  - The elements necessary to obtain a constant temperature in an enclosure
  - The elements of a mobile robot, the elements of a robotic manipulator, etc





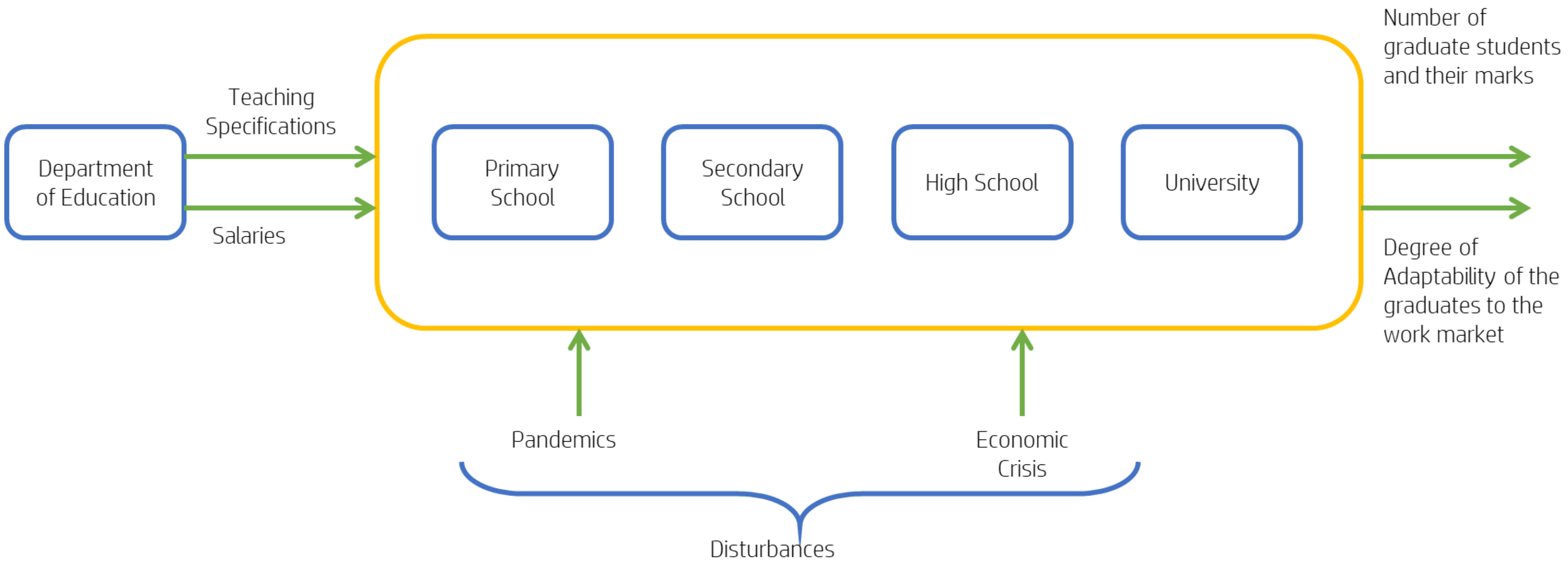
Systems: general aspects

- Generally speaking, a system is a structure that has multiple connections between its component parts.
- Another characteristic is that its elements are structured according to the same criteria or to achieve the same goal.
- In many situations a system can contain subsystems that can be regarded as independent systems.



Systems general aspects

For a better understanding, consider one of the examples stated above, namely the education system. The block diagram of the mentioned system is represented in the following figure:



This is just a representation of a fictitious education system any similarity with reality is purely coincidental. In reality, the education system is more complex and presents other inputs, outputs and disturbances.

Systems general aspects

Aspects regarding the dynamic nature of the education system:

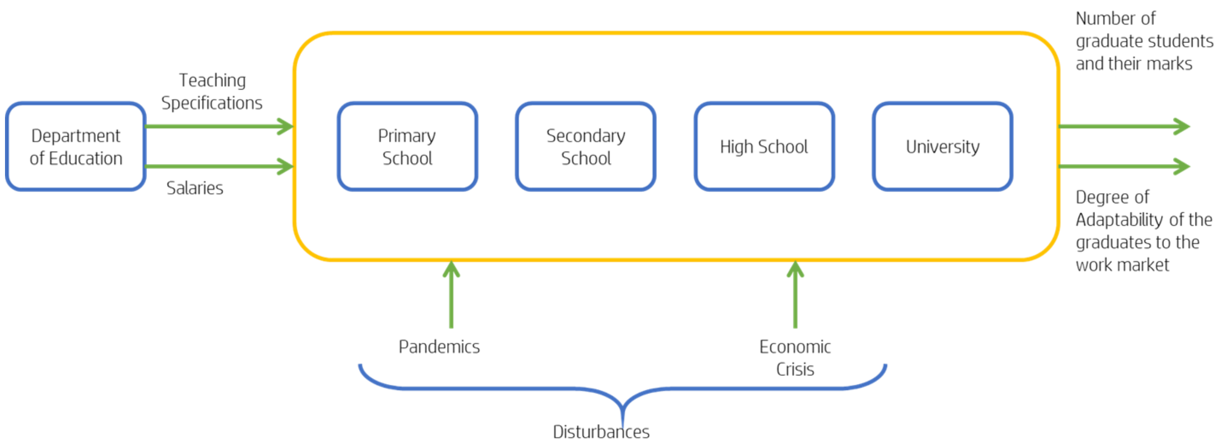
- The output performance depends on the structure of the teaching specifications over a period of time that includes the current year and also a number of previous years

- The salary has an important contribution to the quality of the teaching. Good salary attracts good teachers

- The relations between schools and universities have a decisive role in terms of the quality and continuity of the educational process

- The facilities and the labs of each education institution contribute to student formation

- Another aspect regarding the output performance is related to the presence of disturbances, which can have negative consequences if a rejection mechanism is not applied.



Dynamical Systems

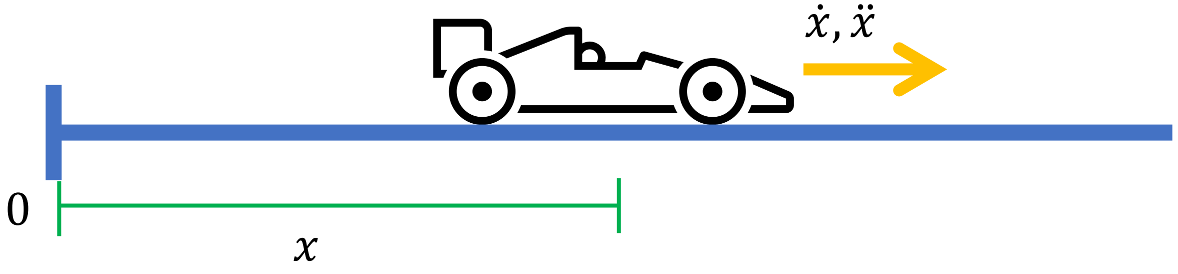
- One subset or type of systems are the **dynamical systems.**
- **A dynamical system is a systems that changes as time evolves, according to specific rules**.
- Almost all, if not all, of the real-world systems are dynamical systems, since they are not static.
- One useful concept of dynamical systems is the "***state***", this can be defined as the minimal information, or variables, that describe the system at a particular point in time and can be used to predict how it will behave in the future.

- Take as an example a car, when the car is moving the states that describe the system are described in the following figure.

$$x = position$$

$$\dot{x} = \frac{dx}{dt} = velocity$$

$$\ddot{x} = \frac{d^2x}{dt^2} = acceleration$$



Dynamical Systems

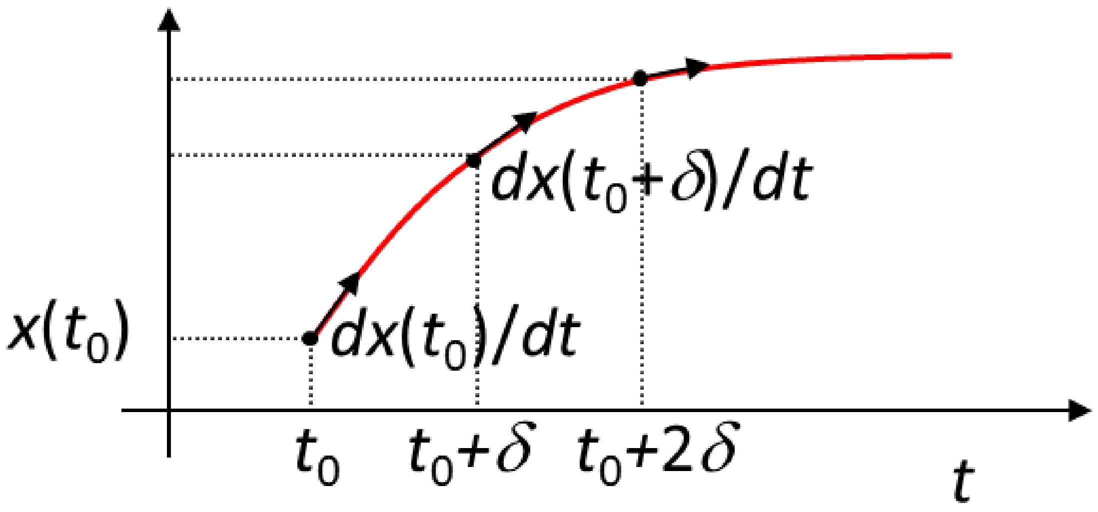
- **Dynamical systems** consists of an abstract **state space**, whose coordinates describe the state at any instant; and a **dynamical rule** that specifies the **immediate future** of all state variables, given only the **present values** of those same state variables.

$$\{f(\mathbf{x}(t), t), \mathbf{x}(t_0)\} \rightarrow \mathbf{x}(t)$$

$f(\mathbf{x}(t))$: Dynamical Rule

$\mathbf{x}(t)$: States

$\mathbf{x}(t_0)$: Initial values



Dynamical Systems

- As stated before, dynamical system consists of two elements:
  - A non-empty space $\mathcal{D}$, e.g. $\mathbb{R}^2$.
  - A map from this space and the time into the same space: $f : \mathcal{D} \times \mathbb{R} \to \mathcal{D}$.
  - Initial conditions $\mathbf{x}(t_0)$
- Then, the dynamical system would be described by the differential equation

$$\dot{\mathbf{x}} = f(\mathbf{x}(t), t)$$

- Loosely speaking, for every point of the space $\mathbf{x} \in \mathcal{D}$, the function $f(\mathbf{x}, t)$ provides the information about the evolution of the system at the instant t.

- [TL;DR] *A dynamical system is anything whose behaviour, evolves over time $\dot{\boldsymbol{x}}$ in a predictable way, given its current state $\boldsymbol{x}(t)$ and rules $f(\boldsymbol{x}, t)$.*

LTI Systems

Definition

{Learn, Create, Innovate};

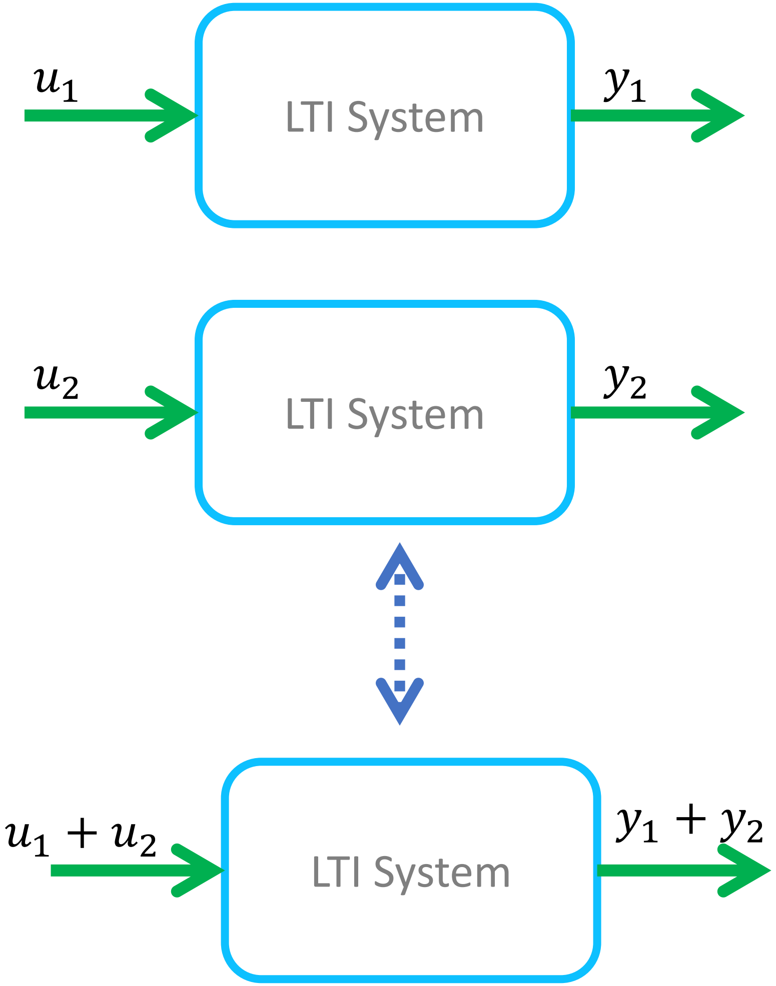
Manchester Robotics

Linear Time Invariant Systems

- Linear Time-Invariant Systems (LTI) are a subset of dynamical systems that comply with two main characteristics:
- Time-invariance: The system does not change with time. In other words, the output does not depend on when an input is applied.

$$\dot{\mathbf{x}} = f(\mathbf{x}(t))$$

- Linearity: A System whose output for a combination of inputs is the same as a linear combination of each individual input (superposition principle).



Linear Time Invariant Systems

- There are many ways to solve differential equations, such as complementary functions (CF), Particular Integrals (PI) and Laplace.
    - Complementary function:
        - Assume the solution is exponential.
        - Assumes the system is homogeneous $(\boldsymbol{u}(t) = 0)$
        - Tries to find an exponential to satisfy the transient part.
    - Particular Integral:
        - Tries to find the steady state behaviour
        - Analysis of the output when $\boldsymbol{t} \to \infty$
    - Laplace Transform: Simplifies this process by analysing the system in the frequency domain, where some operation become simpler such as convolution that becomes a multiplication.
- The key is not to know how to solve it, is more important to know how to analyse it!

Linear Time Invariant Systems

- Linear Dynamical Systems can be **Homogeneous** and **Non-Homogeneous.**
- **Homogeneous Systems:**

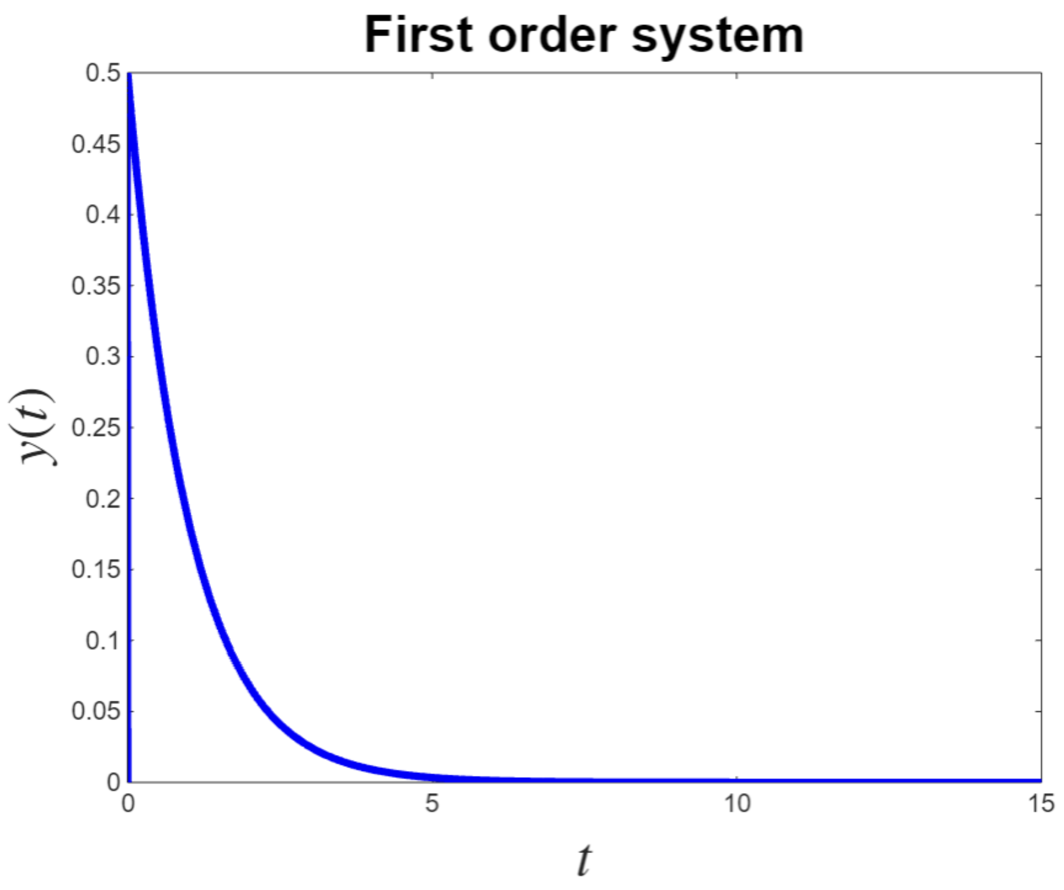
$$\dot{\mathbf{x}} = f\big(\mathbf{x}(t)\big) = \mathrm{A}\mathbf{x}(t) = 0$$

  - No external input/forcing – system evolves only from initial conditions.
  - Solution:

$$y(t) = y_c(t)$$

(complementary function = natural response).

  - Behaviour is dictated by the system poles.
  - Typically decays (if poles are stable).



Linear Time Invariant Systems

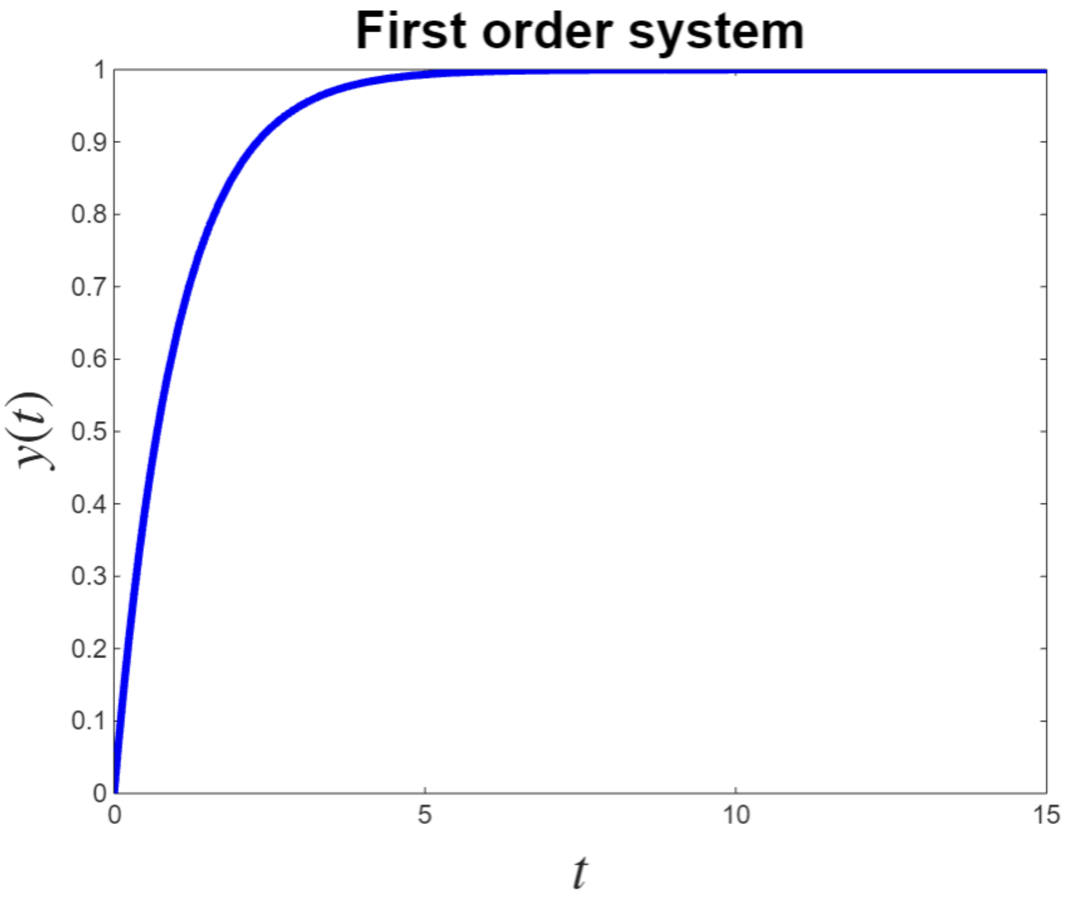
- Non-Homogeneous Systems:

$$\dot{\mathbf{x}} = f(\mathbf{x}(t)) = g(t)$$

  - with a forcing term $g(t) \neq 0$.
  - Driven by an external input.
  - Solution:

$$\mathrm{y(t) = y_c(t) + y_p(t)}$$

    - $\mathrm{y_c(t)}$: "complementary function (transient response)."
    - $\mathrm{y_p(t)}$: "particular solution (steady-state response)."
  - Behaviour is shaped by both **system poles** (transient) and **input poles** (steady-state).



LTI Systems

Representation

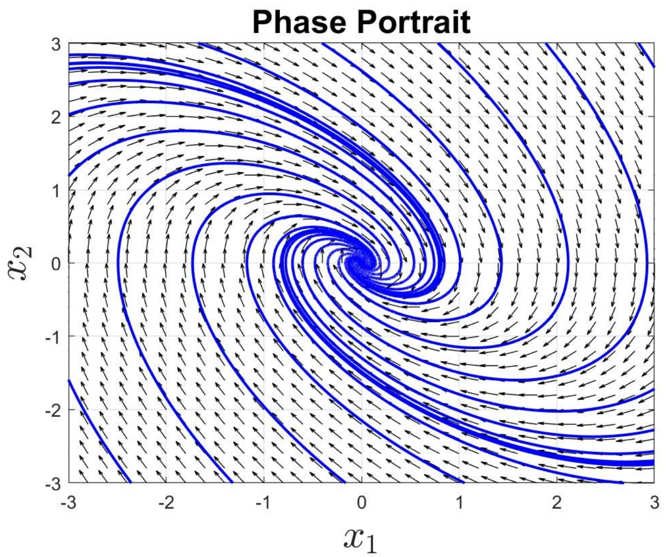
{Learn, Create, Innovate};

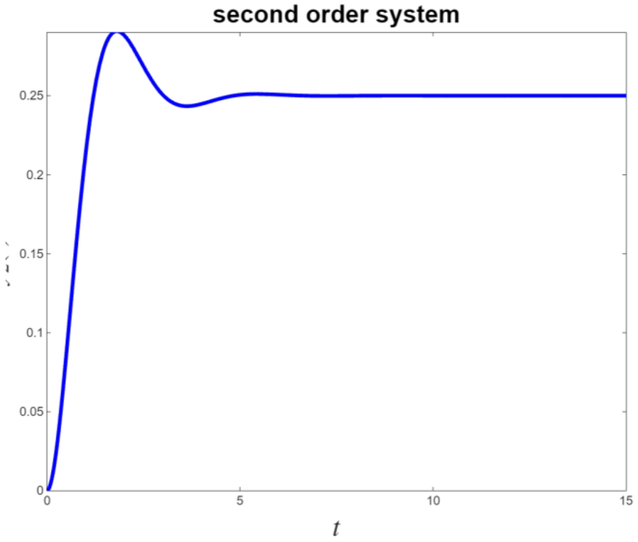
Manchester **Robotics**

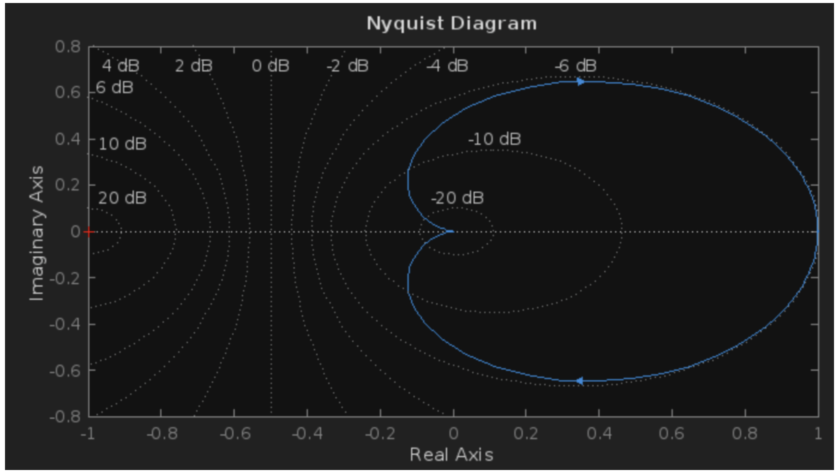
Representation of LTI Systems

Representations

- Different representations exist for LTI systems.
- Each representation conveys different information
- Depending on the usage, some provide more important information than others.
- All of them are used when designing advanced controllers.
- In this section, we will introduce four different representations (keep in mind that there exists many more).







Representation of LTI Dynamical Systems

ODE Representation:

$$\ddot{y} + 2\,\dot{y} + 4\,y = u$$

- Useful when analysing the system in the time domain.
- Useful to solve the differential equations i.e., obtain $y(t)$.
- One solution method is the use of *complementary functions* and *particular integrals* to obtain $y(t)$.
- Some ODE representation can be arbitrarily complex (non-linear) and may not be possible to solve.

File: ex_ode

```
%% Init simulation
clc
clear
close all

%% Use symbolic toolbox to define a 1st order ODE

% dy/dt + Ay(t) = u(t)

% Define variables
syms y(t);

%Input Definition
u=dirac(t); %Impulse

%First order system
Dy=diff(y,t); %Define Derivatives
f1 = Dy(t) + (1)*y(t) == u; %Define Functions

%% Solve ODE
y = simplify(dsolve(f1, [y(0)==0.0]))

%% Plot
figure(1)
fplot(y,[0,15])
```

Representation of LTI Dynamical Systems

Impulse Response Representation

$$\dot{y}(t) + ky(t) = \delta(t), \qquad y(0) = 0$$

$$y(t) = \mathrm{e}^{-\mathrm{kt}}$$

- A LTI dynamic system is characterized its impulse response signal

$$y(t) = \int_{-\infty}^{\infty} g(t-\tau)u(\tau)\, d\tau$$

- Use convolution to calculate $y(t)$ given $\{u(t), g(t), y(0)\}$
- Impulse response of a first order system is equivalent to set initial conditions $y(0) = y_0$
- Impulse response for second order system is equivalent to set initial conditions for $\dot{y}\,(0) = y_0$

```
File: ex_impulse
%% Init simulation
clc
clear
close all
%% Use symbolic toolbox to define a 1st order ODE
% dy/dt + Ay(t) = u(t)
% Define variables
syms y1(t) y2(t);
%Input Definition
u=0.0;
u2 = dirac(t-0.000001);
%First order system
Dy1=diff(y1,t);
Dy2=diff(y2,t);
f1 = Dy1(t) + (1)*y1(t) == u;
f2 = Dy2(t) + (1)*y2(t) == u2;
% Solve ODE's
y1 = simplify(dsolve(f1, [y1(0)==1]))
y2 = simplify(dsolve(f2,[y2(0)==0]))
% Plot
figure(1)
fplot(y1,[0,15],'LineWidth',3,'color','b')
hold
fplot(y2,[0,15],"--",'LineWidth',3,'color','r')
xlabel ('$t$','interpreter','latex','FontSize',22)
ylabel ('$y_1(t), y_2(t)$','interpreter','latex','FontSize',22)
title ('First order systems','FontSize',20)
```

Representation of LTI Dynamical Systems

Transfer Function Representation

$$G(s) = \frac{1}{s+k}$$

- Depicts the Laplace transform of the impulse response of the system.
- **Laplace transform** of the LTI system $g(t)$

$$G(s) = \int_{-\infty}^{\infty} g(t)e^{-st}\,dt$$

- Convolution becomes a **multiplication** in the s-domain

$$Y(s) \;=\; G(s)U(s).$$

- Gives valuable information of the system's behaviour using the poles and zeroes' analysis.
- Provides Frequency response analysis.

File: ex_laplace

```
%% Init simulation
clc
clear
close all
%% Use symbolic toolbox to define a 1st order ODE in Laplace

% dy/dt + Ay(t) = u(t), u(t)=dirac(t)
% G(s)=1/s+A

% Define variables
syms s t;
%Define TF and invert laplace
Y = ((1)/(s+1));
y = simplify(ilaplace(Y))

% Plot
figure(1)
fplot(y,[0,15],'LineWidth',3,'color','b')
hold
xlabel ('$t$','interpreter','latex','FontSize',22)
ylabel ('$y(t)$','interpreter','latex','FontSize',22)
title ('First order systems','FontSize',20)
```

Representation of LTI Dynamical Systems

State-Space Representation

$$\dot{y}(t) = y(t) + u(t)$$

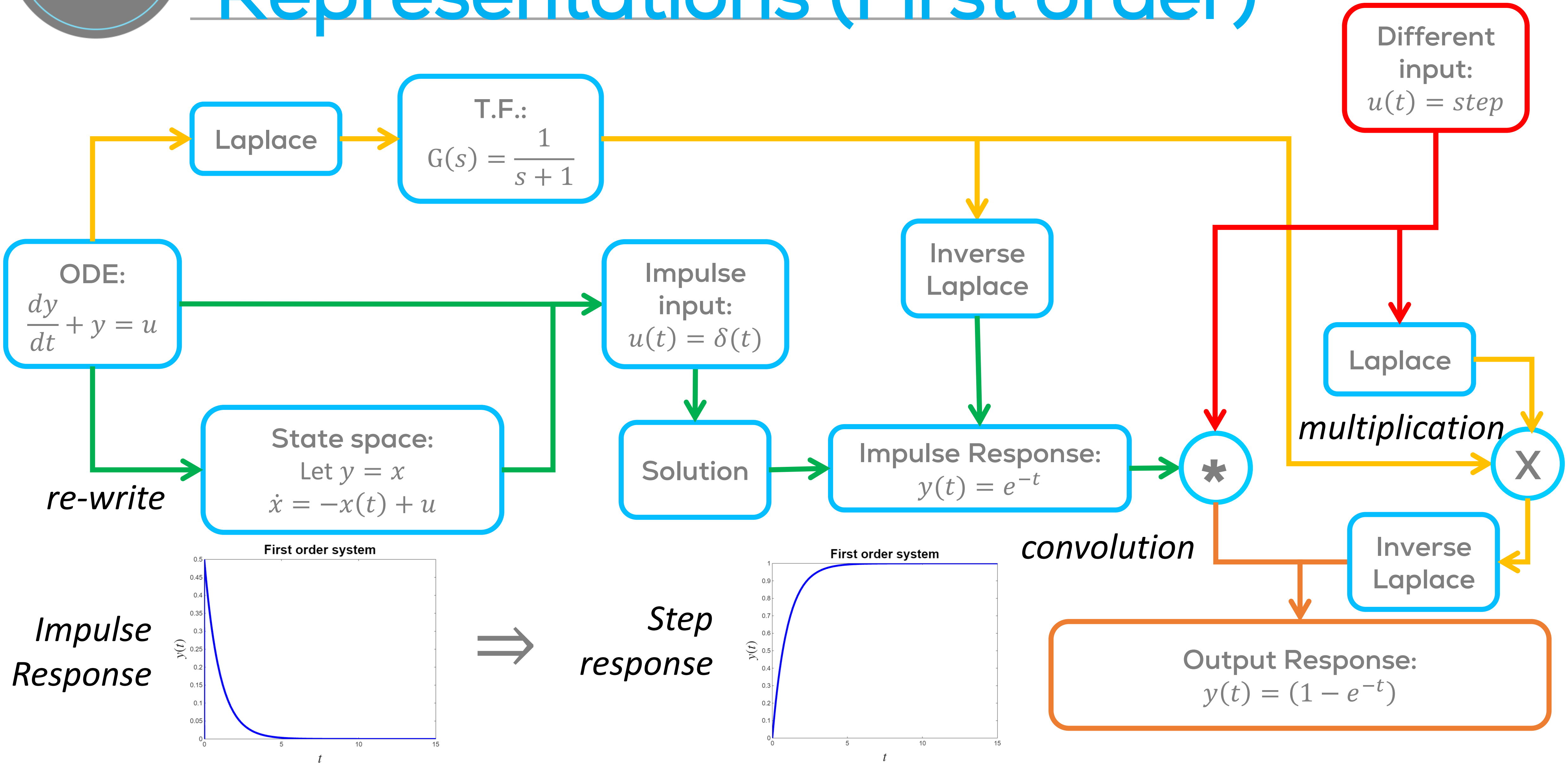
- State space representation allows the user to observe the relationship between the states.
- Like ODEs, state space representation can be arbitrarily complex (non-linear) and it may not be easy or even possible to calculate $y(t)$
- For LTI systems, the solution is given by

$$\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}(0) + \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{B}\mathbf{u}(\tau)\,d\tau$$

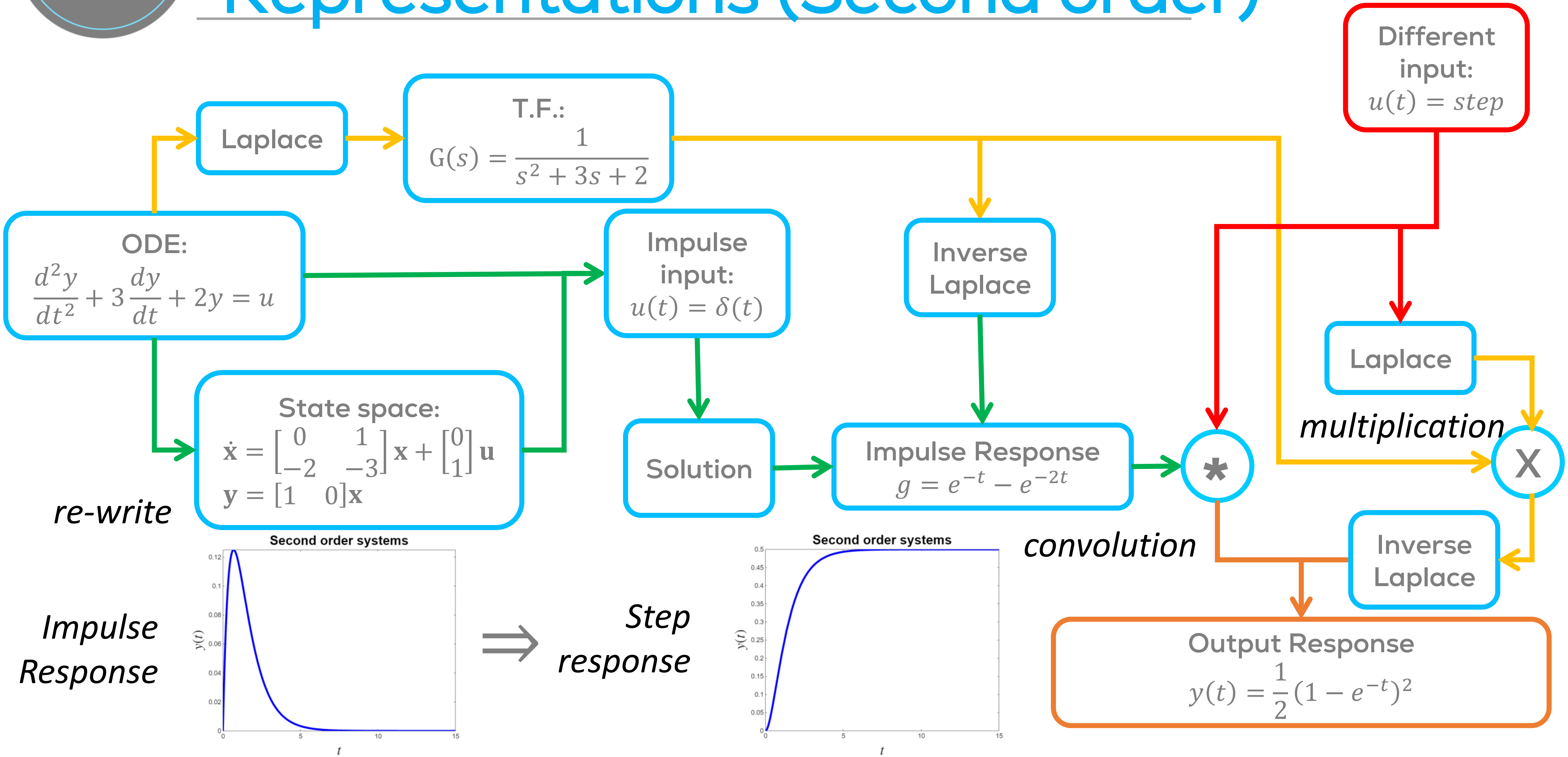
File: ex_ss

```
%% Dynamical System
% dy/dt + k*y(t) = u(t)
%% Initialise simulation
clc
clear
close all
%% Simulation Parameters
y_0=1; % Initial Conditions
dt=0.001; % Sampling time
tf=15; % Final time
%% Euler Approximation configuration
% Vector initialisation
t=0:dt:tf;
y= zeros(length(t),1);
y(1)=y_0;
%% Euler Approximation of the Solution
for k=1:length(t)-1
y(k+1)=y(k)+dt*(-y(k));
t(k+1)=t(k)+dt;
end
%% Plotting
figure(1);
grid on
hold on
axis([0 15 0 2])
% Plot Solution
plot(t,y,'LineWidth',3,'color','r')
%labels
xlabel ('$t$','interpreter','latex','FontSize',22)
ylabel ('$y$','interpreter','latex','FontSize',22)
title ('First Order System','FontSize',20)
```

Examples of LTI Representations (First order)



Examples of LTI Representations (Second order)



Reflections on LTI Representations

All the LTI ODE dynamic system representations are **equivalent**. However, some representations are "more natural"

- **ODEs** Are useful because they depict the problem in terms of "forces".
- **Impulse response signal** is equivalent to the complementary function or transient response component.
- **Transfer functions** are used to simplify the block (control) analysis as **convolution** is simply **multiplication**. Solved using Laplace.
- **State space** can be easily generalized to non-linear & multi-input, multi-output systems. Concepts like controllability, observability and recursive filters can be represented. Solved using (matrix-based) convolution.

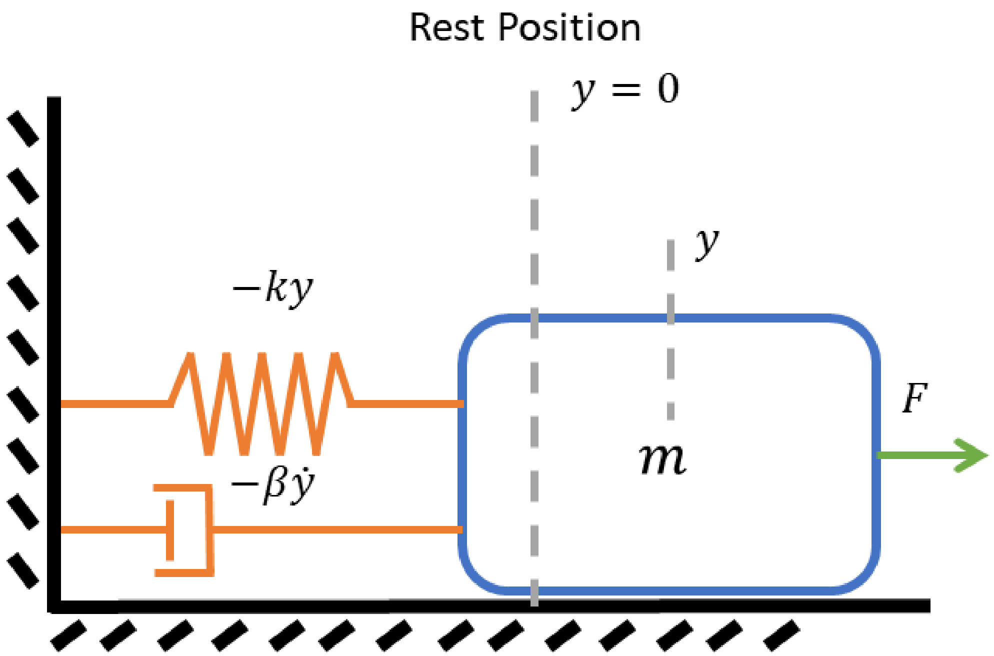
Example

Linear system
Mass Spring

Representation of a linear system

Worked example

- Let us consider an Ideal Mass-Spring-Damper system where an external force F is applied on the mass. The output of the system is the position of the mass y.



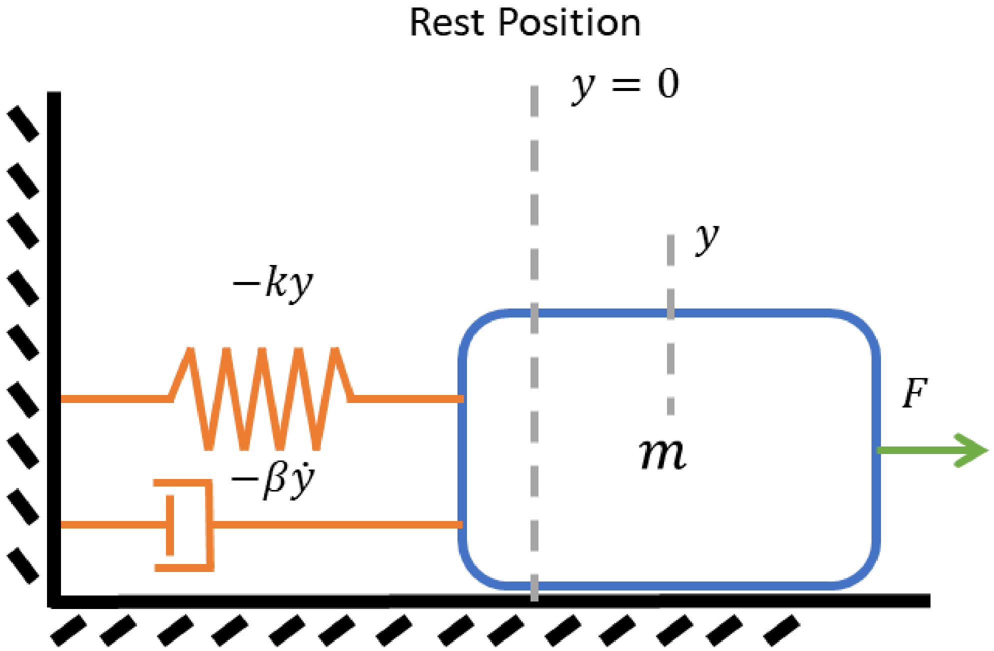
Q: Which are the states (the set of coordinates) that describe the dynamics of this mechanical system?

State-space representation of a linear system

Worked example

- Applying Newton's second law, the dynamics of the system are given by:

$$\sum_i F_i = ma = m\,\ddot{y} \tag{18}$$



State-space representation of a linear system

Worked example

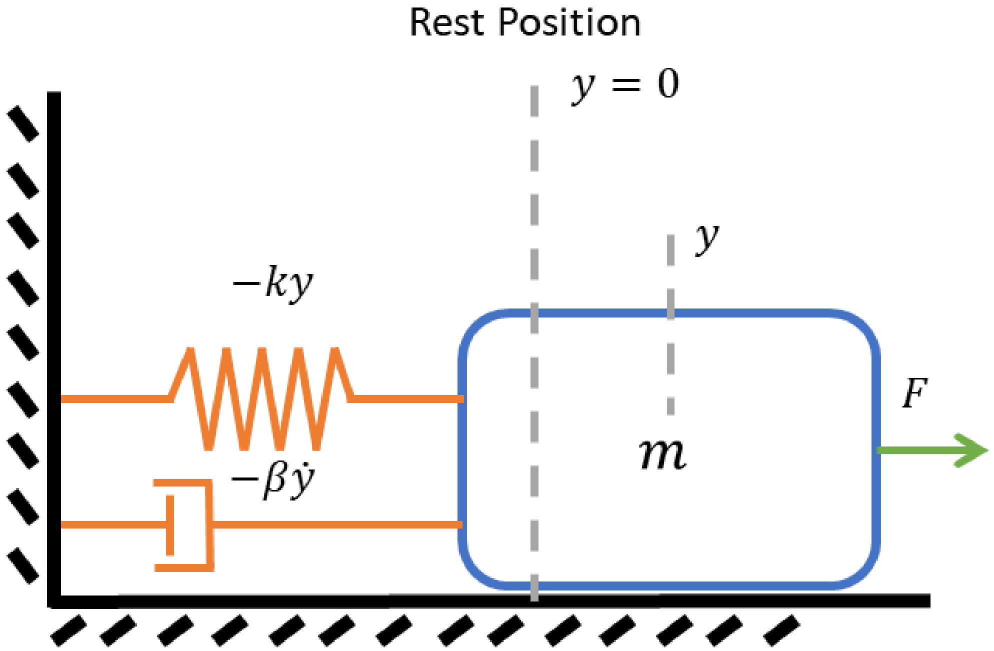
- There are three forces in the direction of y: the spring force ($-ky$), the damper force ($-\beta\,\dot{y}$), and the external force (F).

$$F + (-ky) + (-\beta\,\dot{y}) = m\,\ddot{y} \qquad (19)$$

$$\ddot{y} + \frac{\beta}{m}\,\dot{y} + \frac{k}{m}y = \frac{F}{m} \qquad (20)$$

- Overall dynamic equation is given by

$$\underbrace{\boldsymbol{M}(\boldsymbol{x})\,\ddot{\boldsymbol{x}}}_{\text{“Acceleration” Force}} + \underbrace{\boldsymbol{c}(\boldsymbol{x},\dot{\boldsymbol{x}})\,\dot{\boldsymbol{x}}}_{\text{“Velocity” Force}} + \underbrace{\boldsymbol{g}(\boldsymbol{x})}_{\text{“Position/ gravity” Force}} = \underbrace{\boldsymbol{u}}_{\text{“Input” Force}}$$



Solving ODEs in MATLAB

- Calculate the step response of the LTI ODE.

$$\ddot{y} + \frac{\beta}{m}\dot{y} + \frac{k}{m}y = \frac{1}{m}F$$

Where $\beta = 2, m = 1,\ k = 1,\ F = H(t)$, with zero initial conditions

$$\ddot{y} + 2\,\dot{y} + 1\,y = H(t)$$

a) Using MATLAB symbolic toolbox

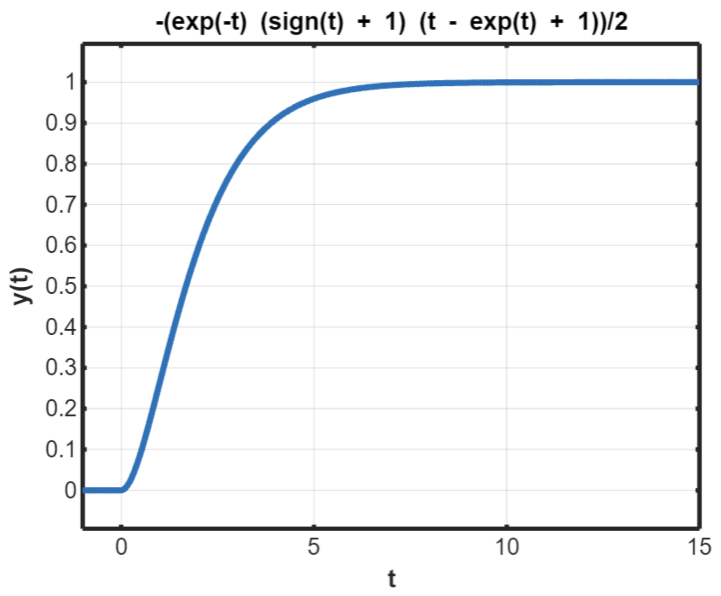
```
m = 1.0; k = 1.0; b = 2.0; f = 1;
syms y(t);
Dy = diff(y,t);
D2y = diff(y,t,2);

y = simplify(dsolve(D2y(t) + (b/m)*Dy(t) +
(k/m)*y(t) == (f/m)*heaviside(t), [y(0)==0,
Dy(0)==0]));

ezplot(y, [-1 15]);
```

b) Using MATLAB symbolic Toolbox and Laplace

```
m = 1.0; k = 1.0; b = 2.0; f = 1;
syms U Y y s t;
U = laplace(heaviside(t));
Y = U*((1/m)/(s^2+(b/m)*s+(k/m)));
y = simplify(ilaplace(Y));
figure(2)
ezplot(y, [0 15]);
```

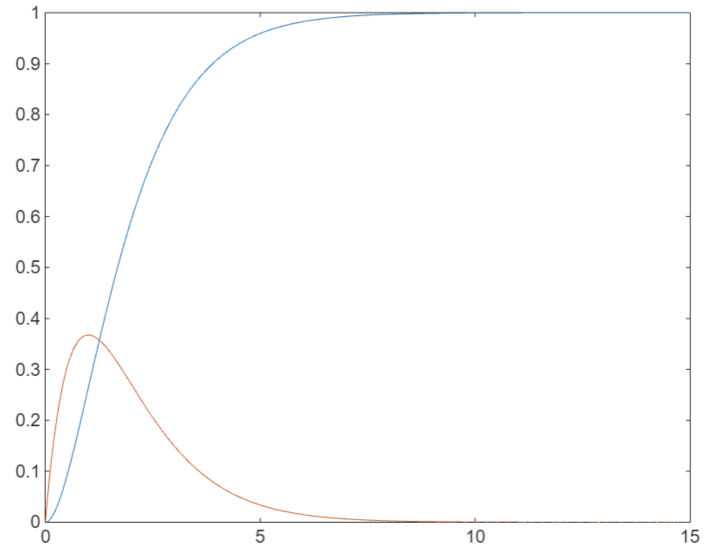


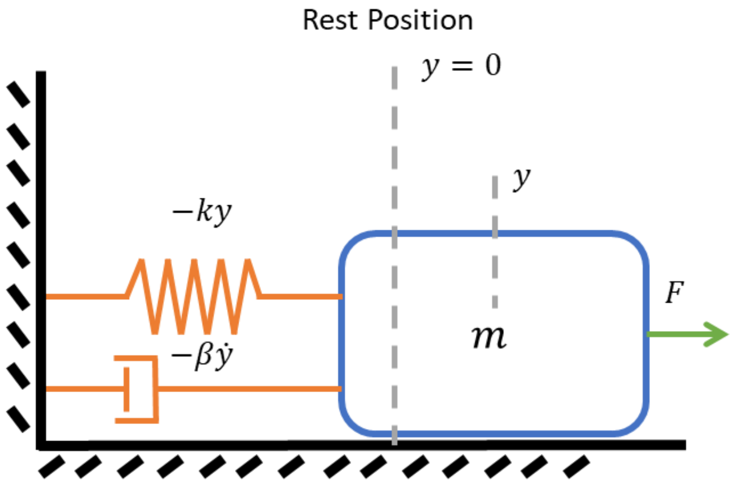
Solving ODEs in MATLAB

b) Using MATLAB ODE integrator

```
%% Initialise Sim
clc
clear all
close all
%% System Parameters
m = 1.0; k = 1.0; b = 2.0; f = 1;
gt = linspace(0,15,1000);
g = heaviside(gt);
tspan = [0 15];
[t,y] = ode45(@(t,y) mass_spring_system(t,y,m,k,b,f,g,gt),
tspan,[0 0]);
figure(6)
plot(t,y)
```

```
function dydt = mass_spring_system(t,y,m,k,b,f,g,gt)
g = interp1(gt,g,t); % Interpolate the data set (gt,g) at time t
dydt = [y(2); -(b/m)*y(2)-(k/m)*y(1)+(1/m)*g];
end
```



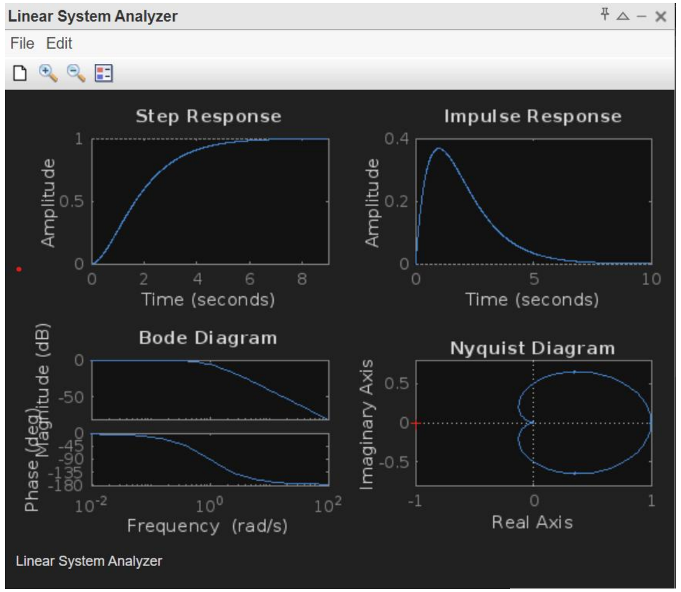


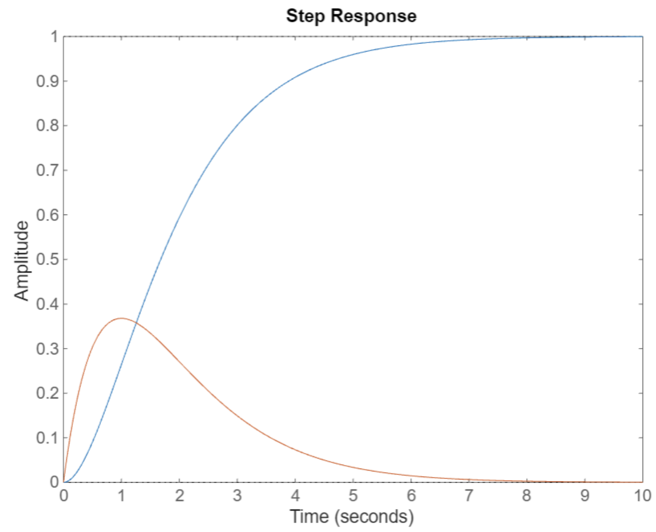
Dynamic Simulation using MATLAB

- Simulate (not solve) the response of the ODE in MATLAB.
- The solution is a vector of output values $y(i)$ at discrete time intervals $t(i)$

a) Control Systems Toolbox

```
%% System Parameters
m = 1.0; k = 1.0; b = 2.0; f = 1;
%% Solution using Control System Toolbox
g = tf([(1/m)],[1 (b/m) (k/m)]);
figure(3)
step(g);
hold
impulse(g);
% Step and impulse response
ltiview(g);  % Handy GUI for analyzing systems
```



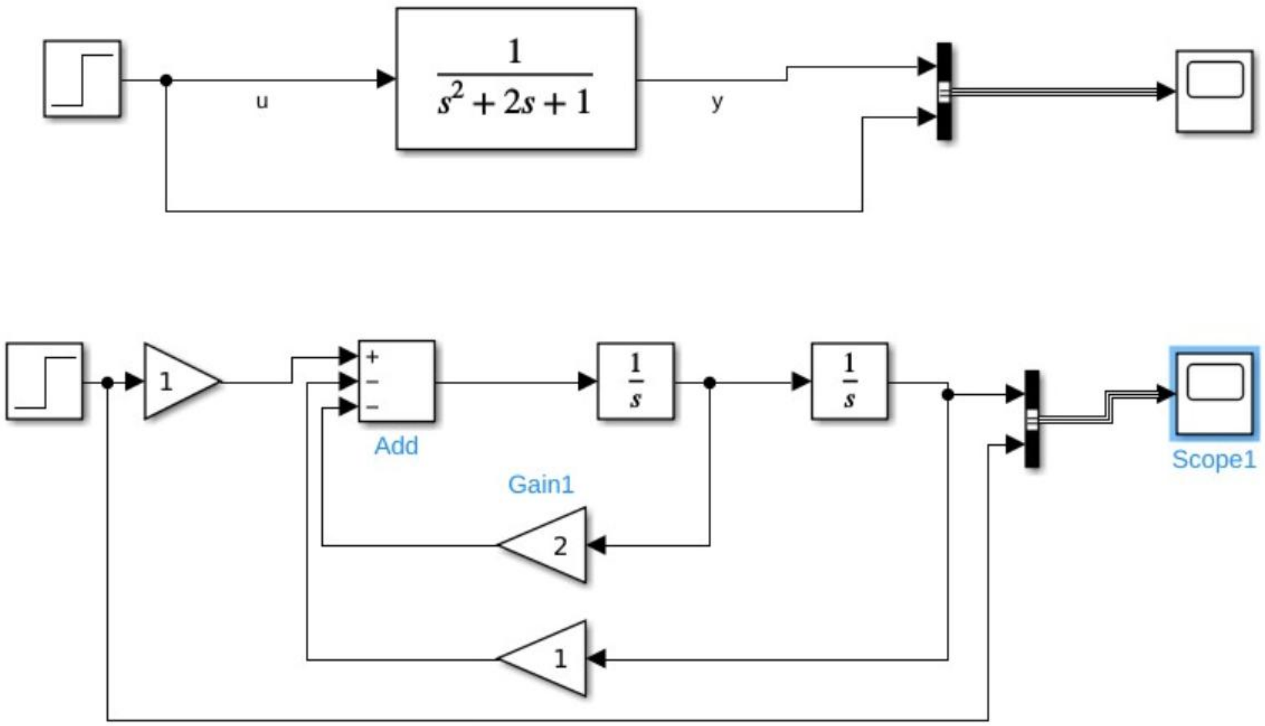


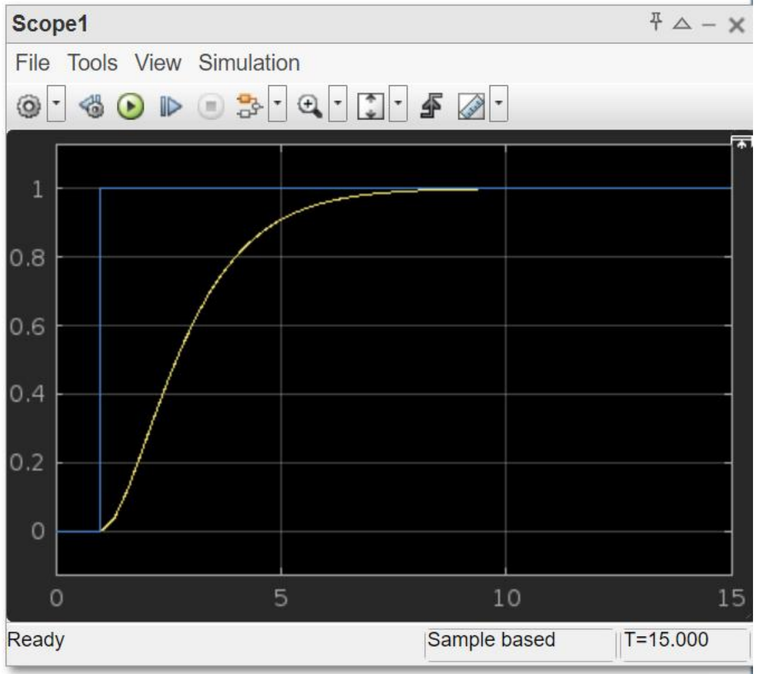
Dynamic Simulation using MATLAB

b) Using **Simulink**

b) Using TF

c) Using Math Operations blocks





LTI Systems

TF: Pole, Zero Analysis

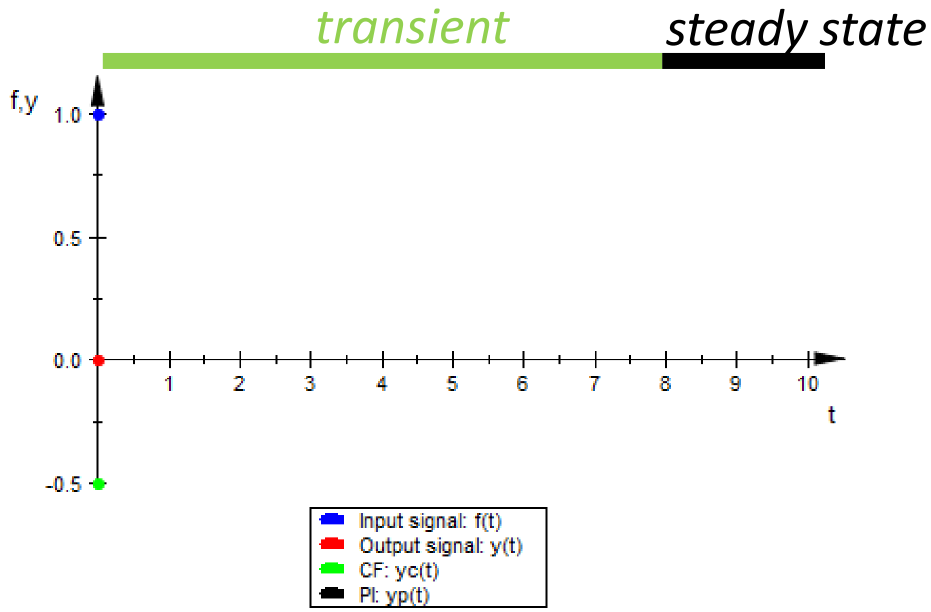
{Learn, Create, Innovate};

Manchester **Robotics**

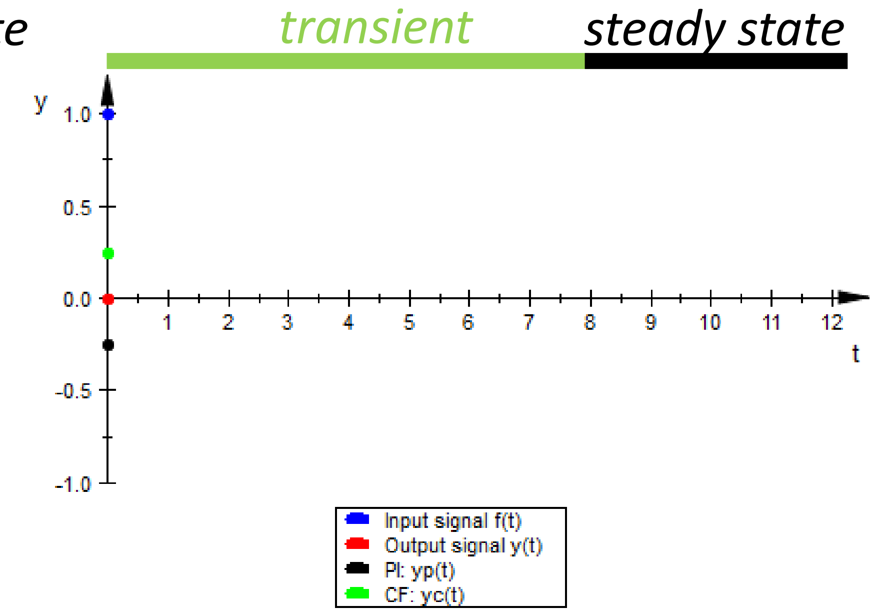
Transients and steady state

- For simple inputs like steps or impulses ($u(t)$), the user can focus in the **transient** and **steady state** phase.
- For pulsed or periodic input signals, the output of a stable, LTI system will converge to a regular, **steady state** pattern after the initial **transient** phase has finished.

$$4\frac{\partial}{\partial t}y(t) + 2\,y(t) = 1, \qquad y(0) = 0$$



$$\frac{\partial^2}{\partial t^2}y(t) + \frac{\partial}{\partial t}y(t) + 2\,y(t) = \cos(2\,t),$$



Poles

- The **transient** phase of **LTI ODE**s and **exponential** signals are **synonymous**:

$$Y(s) = G(s)U(s)$$

- Roughly, if G has poles $\{s1, \dots, sn\}$ and U has poles $\{p1, \dots, pm\}$, then Y's poles are the union of the two sets, i.e.

$$denom\{Y(s)\} = (s - s_1)\cdots(s - s_n)(s - p_1)\cdots(s - p_m)$$

$$Y(s) = \frac{A_1}{s - s_1} + \cdots + \frac{A_n}{s - s_n} + \frac{B_1}{s - p_1} + \cdots + \frac{B_m}{s - p_m}$$

- Taking inverse Laplace transforms

$$y(t) = A_1 e^{s_1 t} + \cdots + A_n e^{s_n t} + B_1 e^{p_1 t} + \cdots + B_m e^{p_m t}$$

Transient response, a "scaled" version of $g(t)$

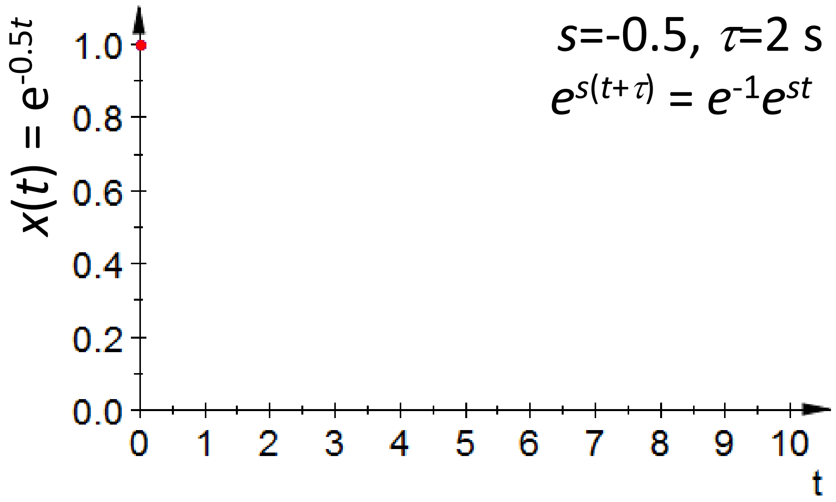
Steady state response, a "scaled" version of $u(t)$

Real, Imaginary & Complex Exponentials

A. Real exponent

$s=-0.5,\ \tau=2$ s

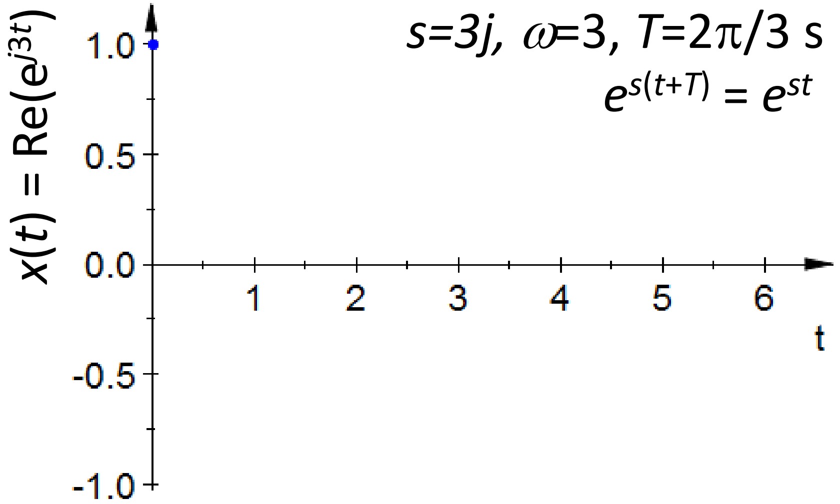
$e^{s(t+\tau)} = e^{-1}e^{st}$



B. Imaginary exponent

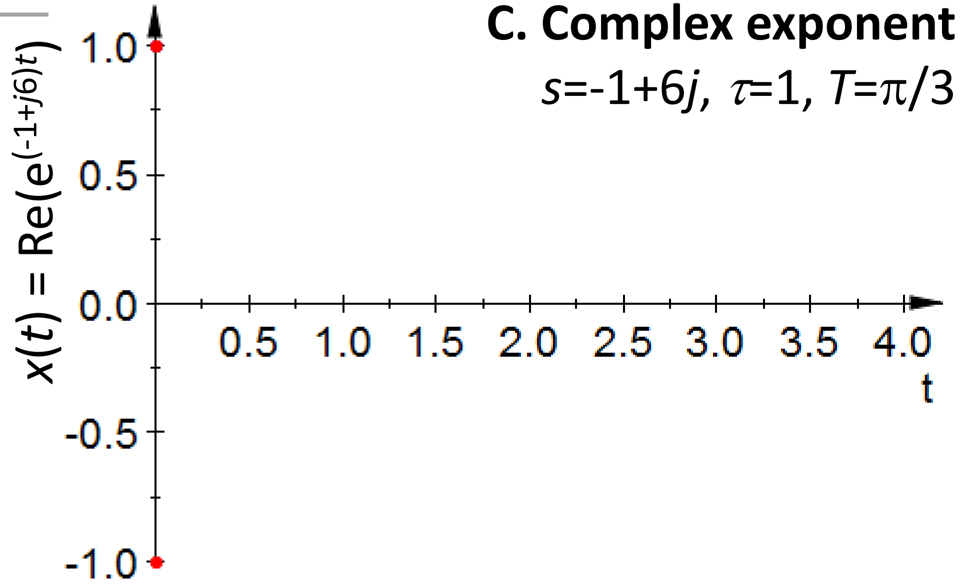
$s=3j,\ \omega=3,\ T=2\pi/3$ s

$e^{s(t+T)} = e^{st}$



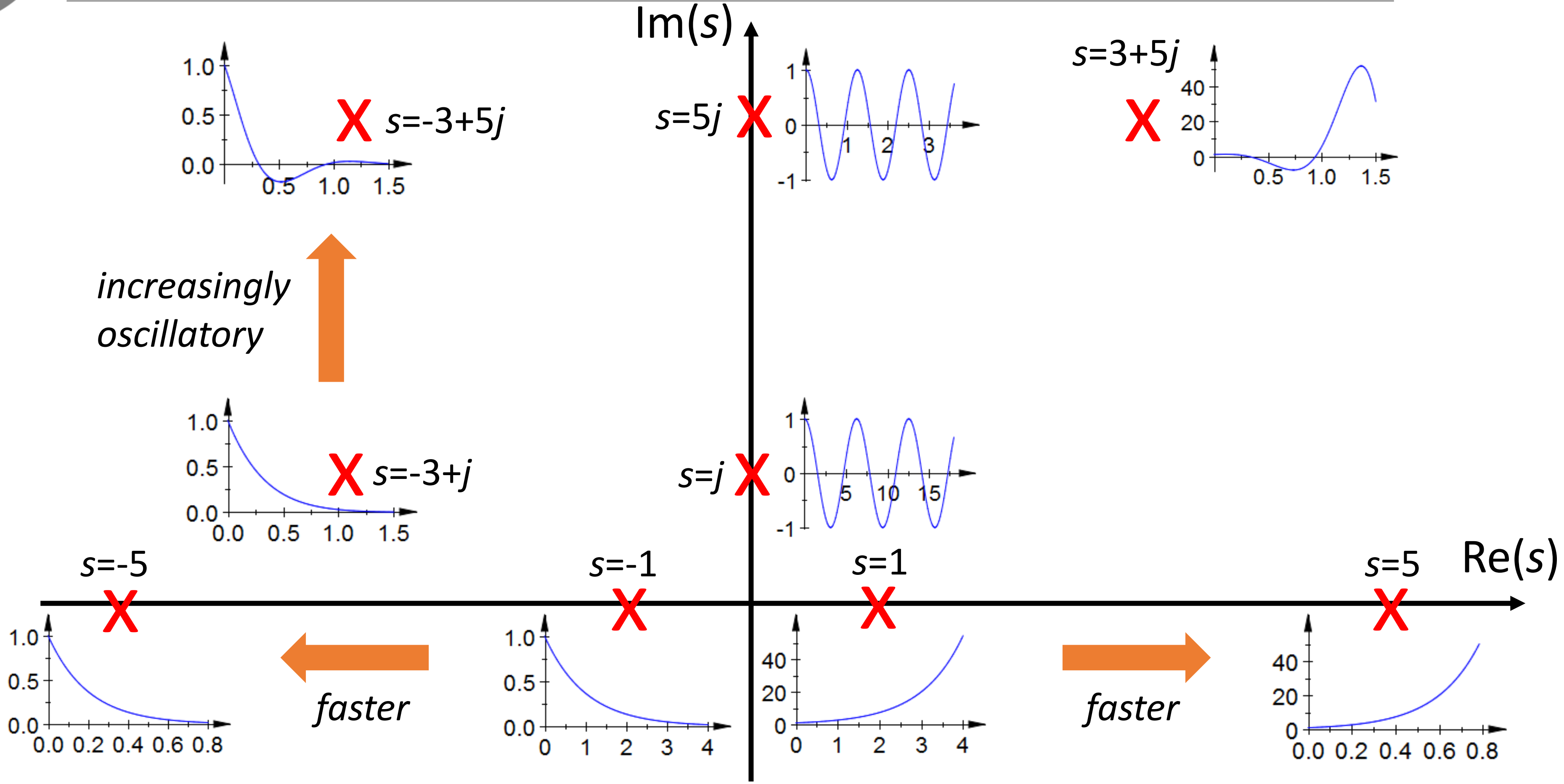
C. Complex exponent

$s=-1+6j,\ \tau=1,\ T=\pi/3$



- In exponential signals, e^{st}, the **exponent** s simply **scales time**.
- For **real** valued exponents, scaling time corresponds to a **time constant**
- For **imaginary** valued exponents, scaling time corresponds to a **time period**.
- Complex valued exponents combine both elements

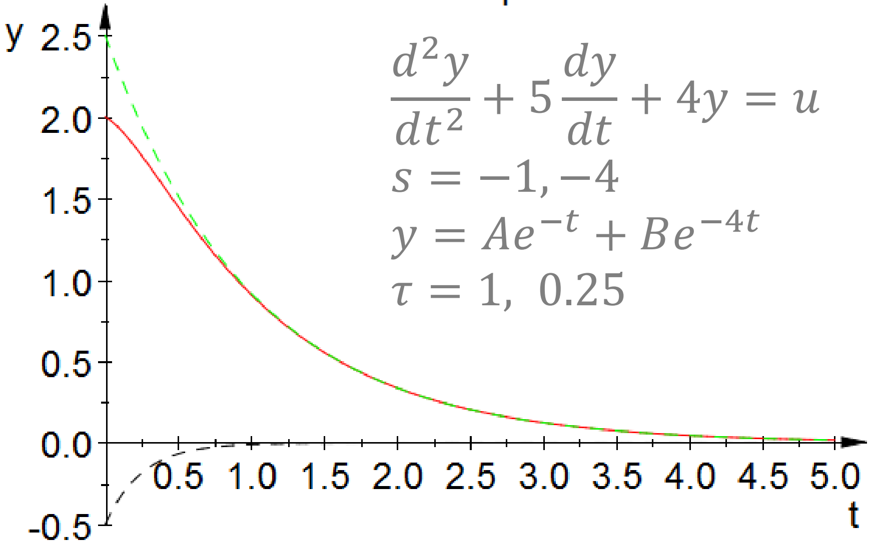
s-plane - e^{st}



Stability means having **poles** in the **left half plane** means that the **transient response** has **exponentials** with negative real parts, i.e. they **decay**.

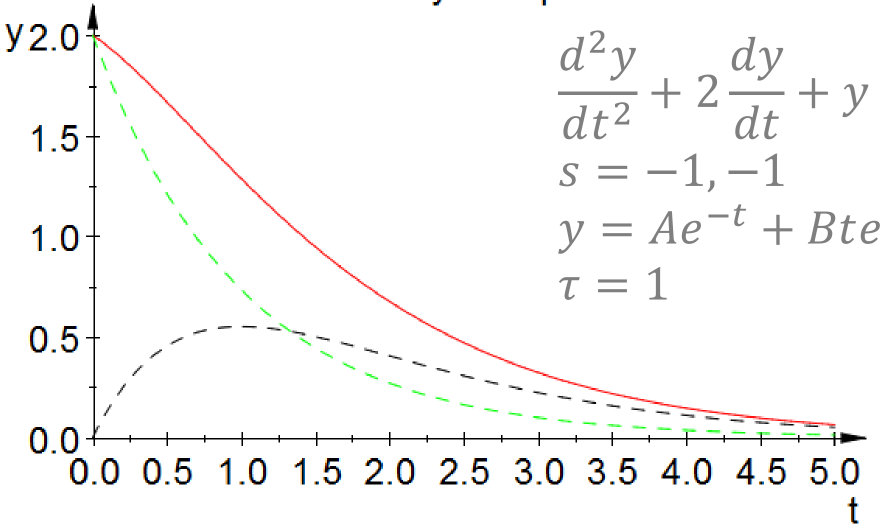
Transient – Pole, 2nd Order, Unforced (CF) Examples

1. Overdamped



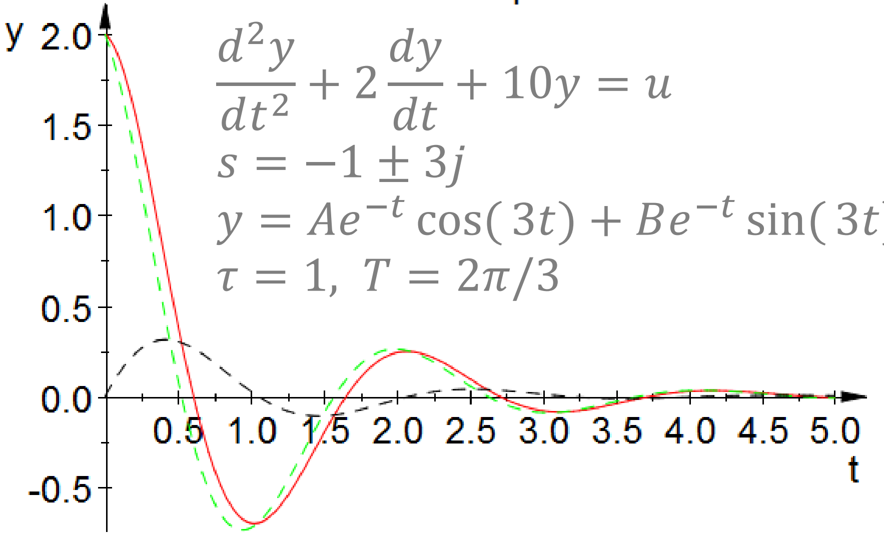
$$\frac{d^2y}{dt^2} + 5\frac{dy}{dt} + 4y = u$$
$$s = -1, -4$$
$$y = Ae^{-t} + Be^{-4t}$$
$$\tau = 1,\ 0.25$$

2. Critically Damped



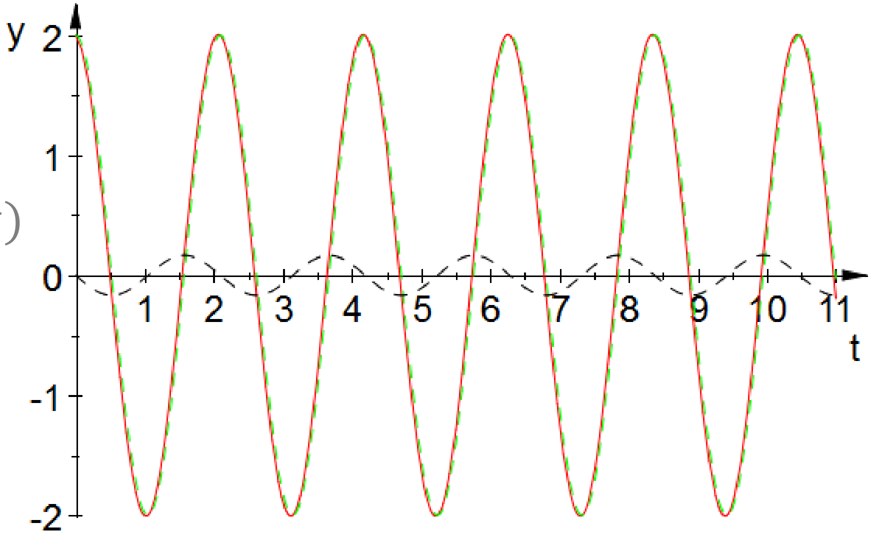
$$\frac{d^2y}{dt^2} + 2\frac{dy}{dt} + y = u$$
$$s = -1, -1$$
$$y = Ae^{-t} + Bte^{-t}$$
$$\tau = 1$$

3. Underdamped



$$\frac{d^2y}{dt^2} + 2\frac{dy}{dt} + 10y = u$$
$$s = -1 \pm 3j$$
$$y = Ae^{-t}\cos(3t) + Be^{-t}\sin(3t)$$
$$\tau = 1,\ T = 2\pi/3$$

4. Resonant



$$\frac{d^2y}{dt^2} + 9y = u,$$
$$y = A\cos(3t) + B\sin(3t),$$
$$s = \pm 3j,$$
$$T = 2\pi/3$$

Zeros

A **zero** represents is a value of s for which **numerator** of the **transfer function** equals **zero**. It operates on / directly affects the input

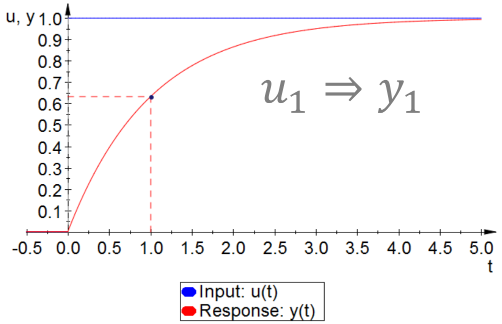
Three interpretations can be given:

1. Pre-filter on the input given by the system actuators.
2. The corresponding input signal $u = e^{st}$ will produce an identically zero output (filter design?)
3. The zeros determine the **constant multipliers** associated with the **exponentials** in the **transient response** $y = Ae^{-t} + Be^{-4t}$
   - A related interpretation is that of **pole-zero cancellation**. You can view this as cancelling a common factor in the transfer function or as the corresponding exponential having a **zero constant multiplier** and hence the effective dynamical order is reduced by 1
4. Analysis for these systems can be done using the superposition principle. In other words how will the system react to the sum of the inputs. Zeros can cause unexpected (non-oscillatory) overshoot or non-minimum phase behaviour.

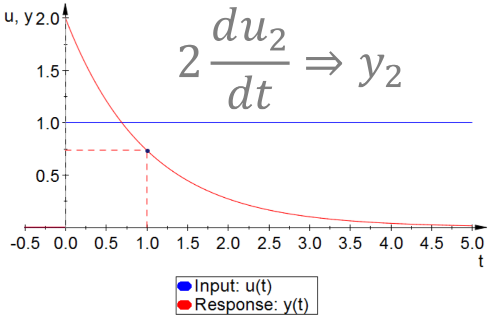
Zeros: 1st & 2nd Order Examples

1st order, step response (biproper)

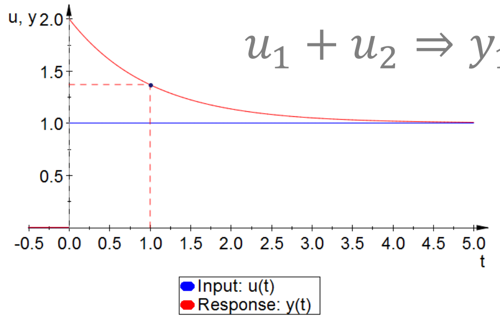
$$\frac{dy}{dt} + y = 2\frac{du}{dt} + u$$



+



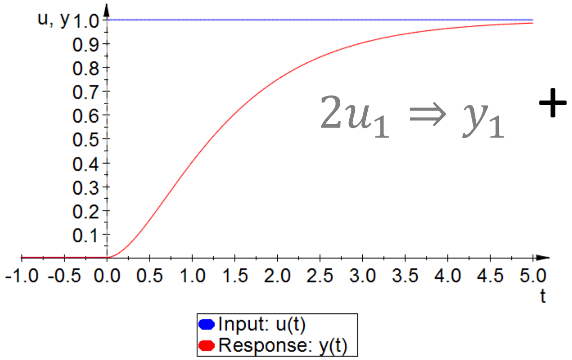
=



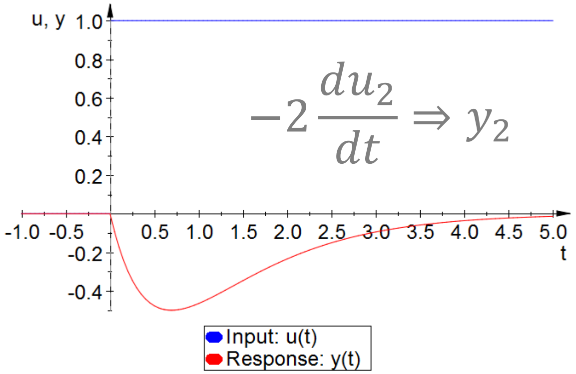
- Transfer function is biproper which causes an initial output jump

2nd order, step response (non minimum phase)

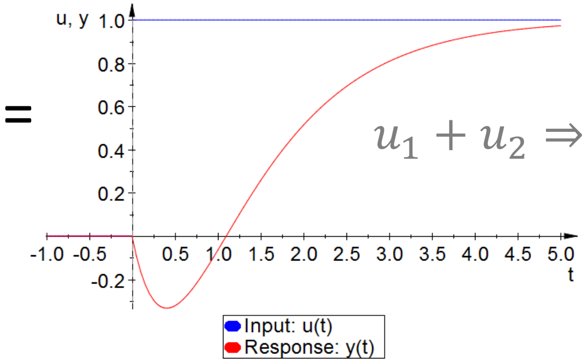
$$\frac{d^2y}{dt^2} + 3\frac{dy}{dt} + 2y = -2\frac{du}{dt} + 2u$$



+



=



- The non minimum phase causes a negative initial response

LTI Systems

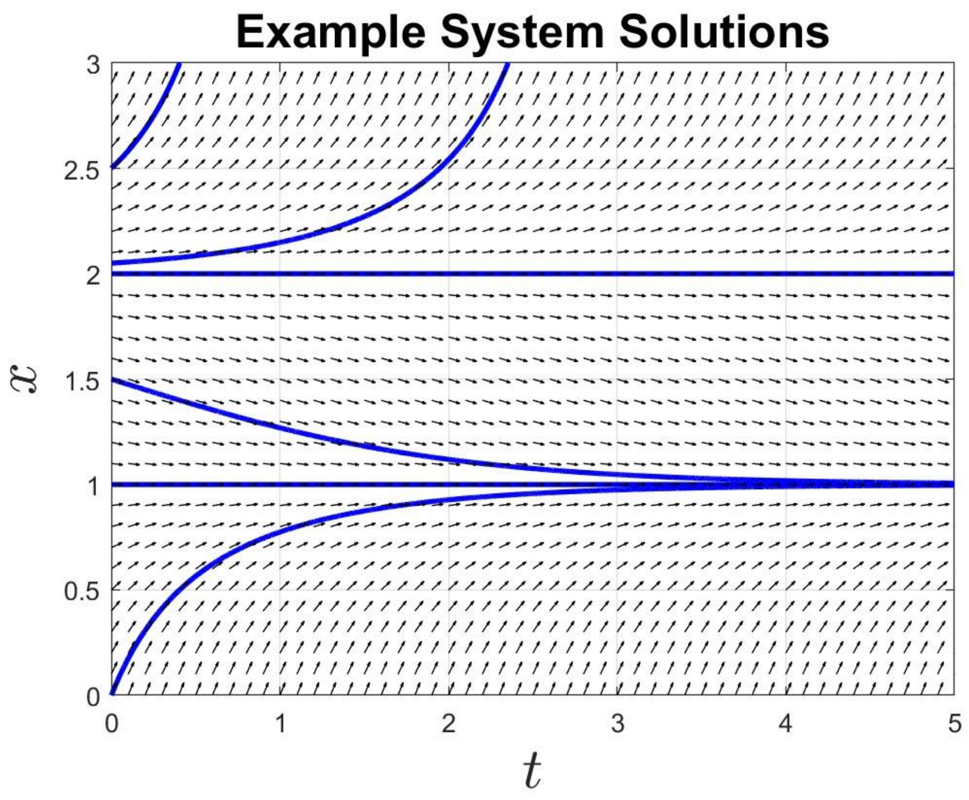
SS Analysis

`{Learn, Create, Innovate};`

Manchester **Robotics**

State-space analysis of a linear system

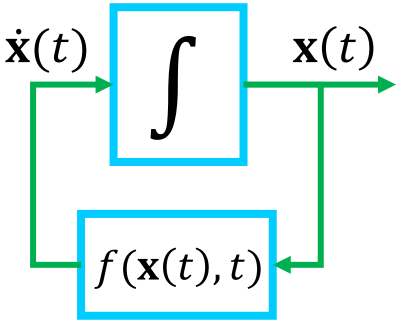
- The key point in control engineering and systems theory is interaction.
- We are interested in studying the dynamical evolution of interconnected systems.
- In particular, feedback systems are the most important for us as robotics and control engineers.
- State space analysis allows the user to observe the relationship between the states.



State-space analysis of a linear system

- The state-space representation and its use in control are the foundation of modern control theory.
- Modelling in state space means describing the system directly in the time domain.

$$\dot{\mathbf{x}} = f(\mathbf{x}(t), t)$$



- **The key idea:** any dynamic system of order n can be expressed as a set of n first-order differential equations, instead of a single n^{th} order differential equation.
- **The main advantage**: making analysis, controller design, and observer design much more systematic.
- Let a system be described as follows

$$\ddot{y}(t) + 3\dot{y}(t) + 2y(t) = 0$$

Let

$$x_1(t) = y(t), \qquad x_2(t) = \dot{y}(t)$$

then

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = -3x_2 - 2x_1$$

Linear Time Invariant Systems

- For LTI systems, f is a linear map of the coordinates $\mathbf{x}$, hence $f(\mathbf{x}(t)) = \mathbf{A}\mathbf{x}(t)$ where $\mathbf{A}$ is a **square matrix**, i.e.

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t)$$

- Loosely speaking, the function f is a linear map from $\mathbb{R}^n$ into $\mathbb{R}^n$.
- The system is isolated from the rest of the universe, its evolution only depends on itself, we can say that is an *autonomous system*.

Stable Node

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

* Any linear map from $\mathbb{R}^n$ into $\mathbb{R}^m$ is represented by a m-by-n matrix.

Example

Worked example

- Let us consider the electrical circuit in Figure 1.

Q: Which are the states (the set of coordinates) which can describe the dynamics of this electrical circuit?

A: The dynamics of the circuit can be described using infinite set of coordinates, but two sets are straightforward:

- The changes at the capacitors $q = (q_1, q_2)$
- The current $i = (i_1, i_2)$.

In this example, we are going to model the same circuit using both sets of coordinates.

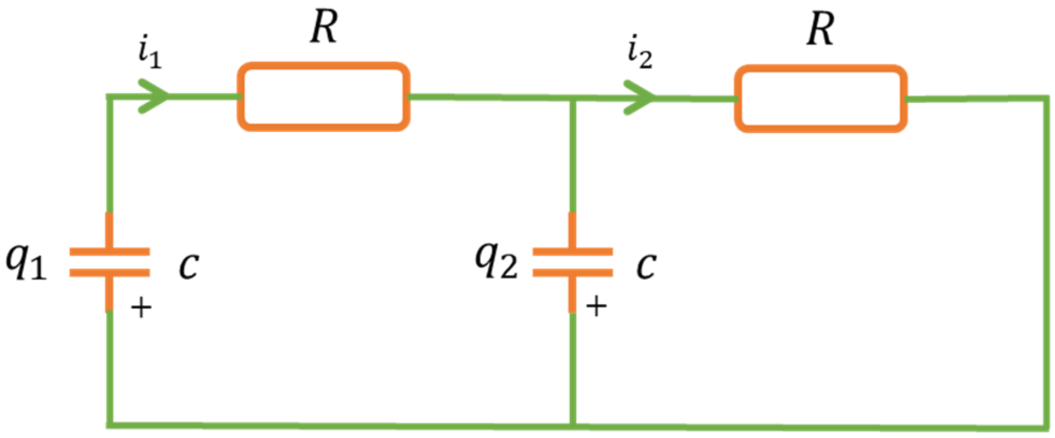


Figure 1: Electrical Circuit

Example

Worked example: Using charge q

- The system can be analysed using two methodologies: using the charges $\boldsymbol{q}$ or the current $\boldsymbol{i}$.

1. Using the charge $\boldsymbol{q}$:
   - Applying Kirchhoff's Voltage Law (KVL) on the left mesh:

$$\sum_i V_i^{left} = \frac{1}{c}q_1 + i_1 R - \frac{1}{c}q_2 = 0 \Rightarrow \Rightarrow i_1 = -\frac{1}{cR}q_1 + \frac{1}{cR}q_2 \quad (2)$$

   - And using KVL on the right:

$$\sum_i V_i^{right} = i_2 R + \frac{1}{c}q_2 = 0 \Rightarrow \Rightarrow i_2 = -\frac{1}{cR}q_2 \quad (3)$$

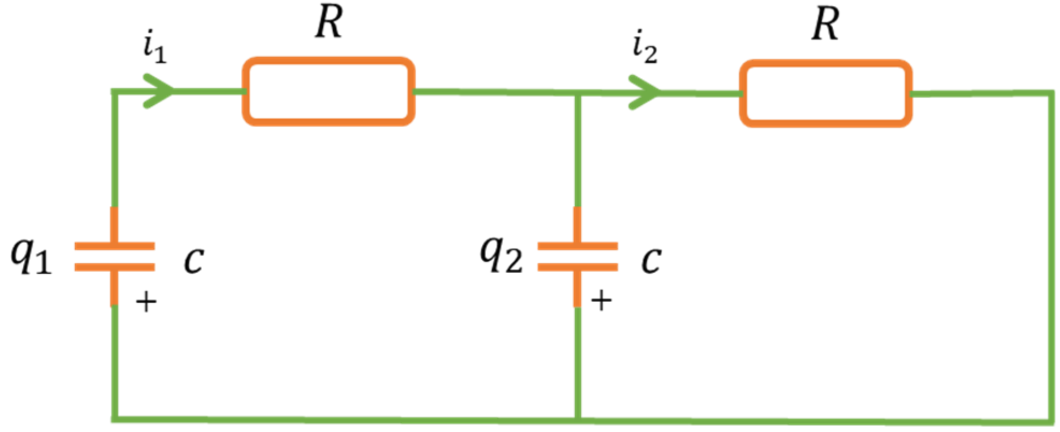


Figure 1: Electrical Circuit

Example

Worked example: Using charge q

- Moreover, both charges and currents are related as follows:

$$i_x = i_2 - i_1 \tag{4}$$

- Therefore,

$$\dot{q}_1 = i_1 = -\frac{1}{cR}q_1 + \frac{1}{cR}q_2 \tag{5}$$

$$\dot{q}_2 = i_x = i_2 - i_1 = -\frac{1}{cR}q_2 + \frac{1}{cR}q_1 - \frac{1}{cR}q_2 \tag{6}$$

- Or equivalently, the matrix form:

$$\dot{\boldsymbol{q}} = \begin{bmatrix} -\frac{1}{Rc} & \frac{1}{Rc} \\ \frac{1}{Rc} & -\frac{2}{Rc} \end{bmatrix} \boldsymbol{q} \tag{7}$$

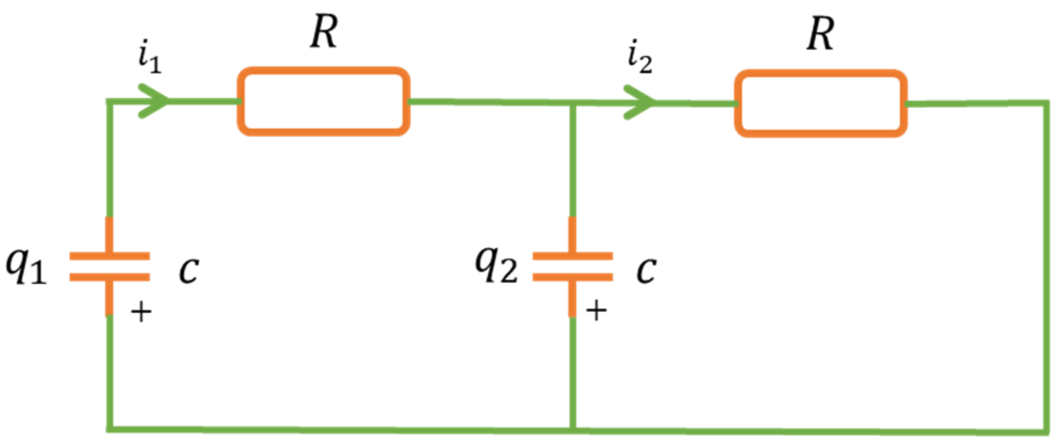


Figure 1: Electrical Circuit

Change of Coordinates

- Since the system is represented by a linear map of the coordinates A.
- Can we change the coordinates of the system?
- Let us assume that we have two different bases on $\mathbb{R}^n$, $\mathbf{x}$ and $\mathbf{z}$.
- Then there exists a non-singular square matrix T such that $\mathbf{z} = \mathrm{T}\mathbf{x}$.

- The dynamical system can also be represented in the basis $\mathbf{z}$ as follows

$$\dot{\mathbf{z}} = \mathrm{T}\dot{\mathbf{x}} = \mathrm{T}\mathrm{A}\mathbf{x} = \mathrm{T}\mathrm{A}\mathrm{T}^{-1}\mathbf{z}$$

- In the context of **dynamical systems**, a **change of coordinates** means **representing the system in a different set of variables** (a different "coordinate system") without altering the underlying dynamics.

Example

Worked example: Using current $\boldsymbol{i}$

2. Using the current $\boldsymbol{i}$:

- The time derivatives of (3) and (2) are given by:

$$\dot{i}_2 R + \frac{1}{c}\dot{q}_2 = 0 \Rightarrow \dot{q}_2 = -Rc\dot{i}_2 \tag{8}$$

$$\frac{1}{c}\dot{q}_1 + \dot{i}_1 R - \frac{1}{c}\dot{q}_2 = 0 \Rightarrow \dot{q}_1 = i_1 = -Rc\dot{i}_1 - Rc\dot{i}_2 \tag{9}$$

- The dynamical equations in the capacitors can be written as:

$$i_1 = \dot{q}_1 = -Rc\dot{i}_1 - Rc\dot{i}_2 \tag{10}$$

$$\dot{q}_2 = i_2 - i_1 = -\frac{1}{cR}q_2 + \frac{1}{cR}q_1 - \frac{1}{cR}q_2 \tag{11}$$

- Reordering the above equation, we get the result in the matrix form:

$$\dot{\boldsymbol{i}} = \begin{bmatrix} -\frac{2}{Rc} & \frac{1}{Rc} \\ \frac{1}{Rc} & -\frac{1}{Rc} \end{bmatrix} \boldsymbol{i} \tag{12}$$

- **Q:** *This electrical circuit is an autonomous systems or a non-autonomous system?*

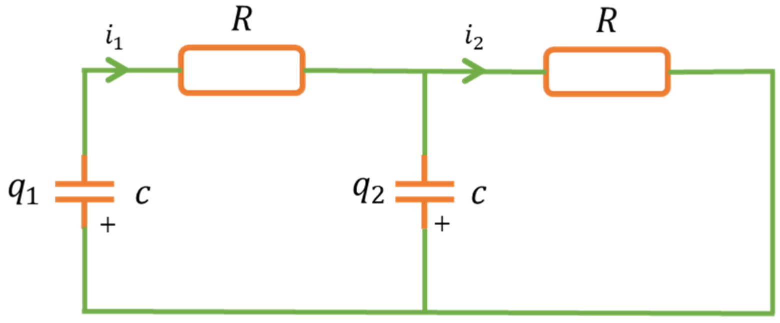
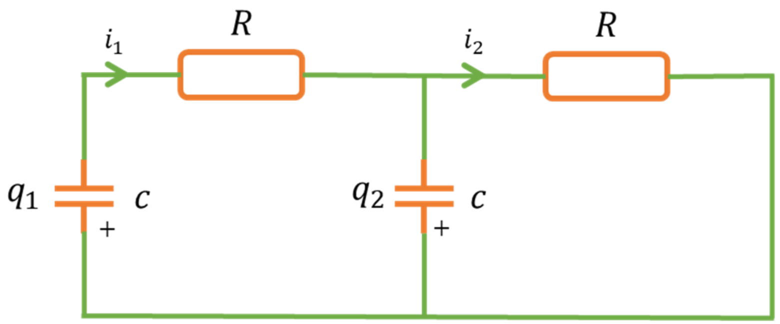


Figure 1: Electrical Circuit

Example

- We have two different sets of coordinates for the previous circuit.



$$\dot{\boldsymbol{i}} = \begin{bmatrix} -\frac{2}{Rc} & \frac{1}{Rc} \\ \frac{1}{Rc} & -\frac{1}{Rc} \end{bmatrix} \boldsymbol{i} \qquad (12)$$

$$\dot{\boldsymbol{q}} = \begin{bmatrix} -\frac{1}{Rc} & \frac{1}{Rc} \\ \frac{1}{Rc} & -\frac{2}{Rc} \end{bmatrix} \boldsymbol{q} \qquad (7)$$

- According to equations (2) and (3) in the previous slides

$$i_1 = -\frac{1}{cR} q_1 + \frac{1}{cR} q_2$$

$$i_2 = -\frac{1}{cR} q_2$$

$$\therefore \boldsymbol{i} = \begin{bmatrix} -\frac{1}{cR} & \frac{1}{cR} \\ 0 & -\frac{1}{cR} \end{bmatrix} \boldsymbol{q} \qquad (13)$$

For this case, the matrix in equation (13) represents the Transformation Matrix T.

Example

- Applying the transformation result to the system

$$\dot{\boldsymbol{q}} = \begin{bmatrix} -\frac{1}{Rc} & \frac{1}{Rc} \\ \frac{1}{Rc} & -\frac{2}{Rc} \end{bmatrix} \boldsymbol{q} = \mathrm{A}\,\boldsymbol{q}$$

$$\dot{\boldsymbol{i}} = \mathrm{TAT}^{-1}\boldsymbol{i}$$

$$\dot{\boldsymbol{i}} = \begin{bmatrix} -\frac{1}{cR} & \frac{1}{cR} \\ 0 & -\frac{1}{cR} \end{bmatrix} \begin{bmatrix} -\frac{1}{Rc} & \frac{1}{Rc} \\ \frac{1}{Rc} & -\frac{2}{Rc} \end{bmatrix} \begin{bmatrix} -\frac{1}{cR} & \frac{1}{cR} \\ 0 & -\frac{1}{cR} \end{bmatrix}^{-1} \boldsymbol{i}$$

- We recover the system in equation (12)

$$\dot{\boldsymbol{i}} = \begin{bmatrix} -\frac{2}{Rc} & \frac{1}{Rc} \\ \frac{1}{Rc} & -\frac{1}{Rc} \end{bmatrix} \boldsymbol{i}$$

- Hence, performing the basis transformation we obtain the same result as in (12).
- This example has demonstrated the relationship between two representations of the same system.
- Any transfer function can be represented by infinite state-space representations

State-space representation of a linear system

Dynamical Systems Models

- We would like to model our dynamical system, including explicitly input u and output y:

$$\dot{x} = f(x,u) \qquad x \in \mathbb{R}^{n_x}, \qquad u \in \mathbb{R}^{n_u} \tag{13}$$

$$y = h(x,u) \qquad y \in \mathbb{R}^{n_y} \tag{14}$$

where n_x is the number of state coordinates, n_u is the number of inputs, n_y is the number of outputs.

- This representation is called state-space representation.
- Is a very general, and most real systems can be modelled by (13) and (14).
- The equations (13) and (14) are referred to as the system equation and the output equation, respectively.
- In contrast with the transfer function representation of a system, the state-space representation is not limited to linear systems.

State-space representation of a linear system

Linear Dynamical Systems Models

- A linear time invariant system can be represented in state space as follows:

$$\dot{x} = Ax + Bu \qquad x \in \mathbb{R}^{n_x}, \qquad u \in \mathbb{R}^{n_u} \tag{15}$$

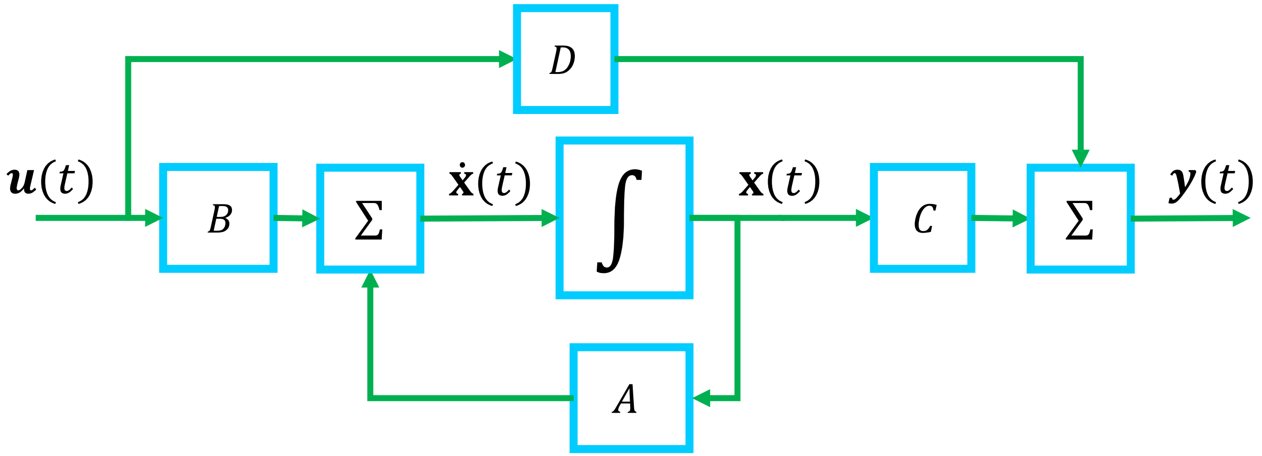
$$y = Cx + Du \qquad y \in \mathbb{R}^{n_y} \tag{16}$$

where $A \in \mathbb{R}^{n_x \times n_x}$, $B \in \mathbb{R}^{n_x \times n_u}$, $C \in \mathbb{R}^{n_y \times n_x}$, and $D \in \mathbb{R}^{n_y \times n_u}$.

- Equations (15) and (16) are said to be the state-space representation of a linear system.

- For systems with single input and output (SISO), i.e., $n_u = n_y = 1$, B is a column vector, C is a row vector and D is a number.

- Systems with several inputs and several outputs, i.e., $n_u > 1$, $n_y > 1$, are referred to as Multiple-Input Multiple-Output (MIMO).



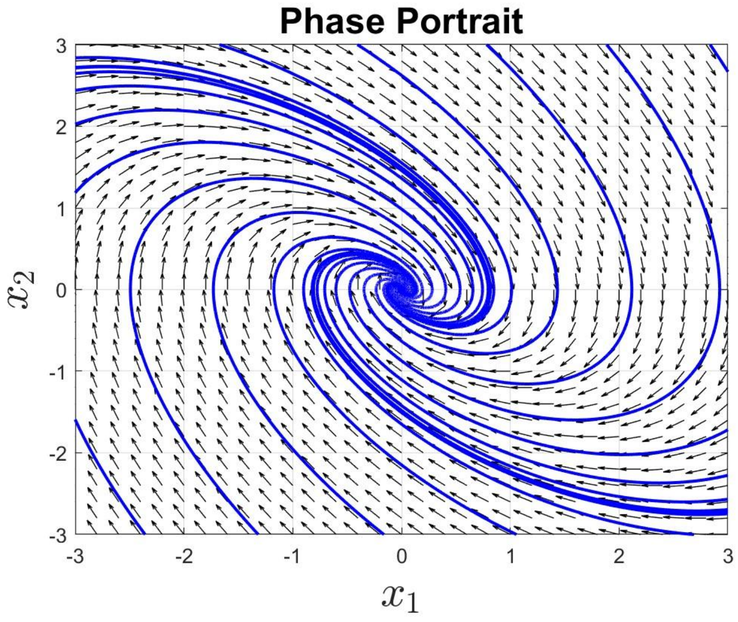
State-space representation of a linear system

- For this class, we will restrict our attention to SISO systems.
- Any ordinary differential equation in the form:

$$\frac{d^n y}{dt^n} + a_{n-1}\frac{d^{n-1} y}{dt^{n-1}} + \cdots + a_1 \frac{dy}{dt} + a_0 y = b_m \frac{d^m u}{dt^m} + b_{m-1}\frac{d^{m-1} u}{dt^{m-1}} + \cdots + b_1 \frac{du}{dt} + b_0 u \tag{17}$$

with $m < n$ has an equivalent state-space representation.

- Among all possible state-space representations of a system, three are very important: controller canonical form, observer canonical form, and modal form



Controller Canonical Form

- Consider a system described by

$$\frac{d^n y}{dt^n} + a_{n-1}\frac{d^{n-1}y}{dt^{n-1}} + \cdots + a_1\frac{dy}{dt} + a_0 y$$

$$= b_m\frac{d^m u}{dt^m} + b_{m-1}\frac{d^{m-1}u}{dt^{m-1}} + \cdots + b_1\frac{du}{dt} + b_0 u$$

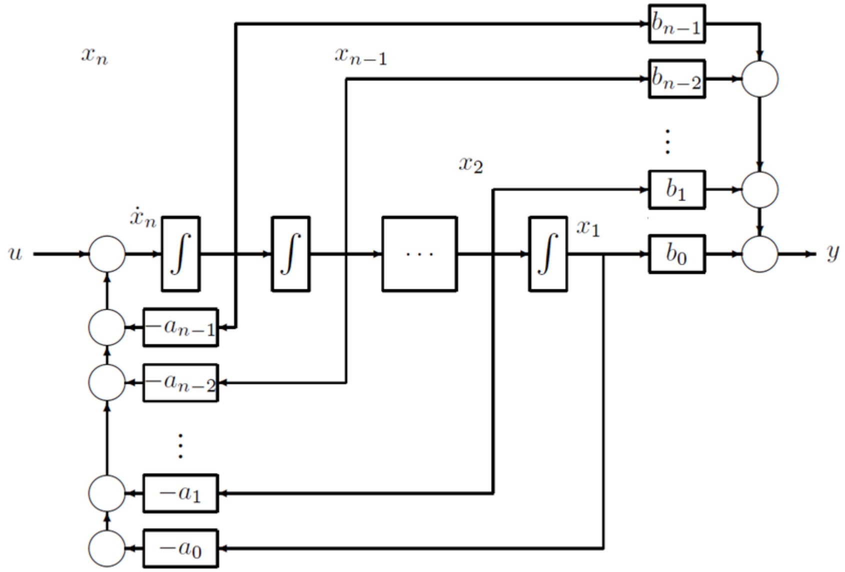
- Then its observer canonical form is given by

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 \\ -a_0 & -a_1 & -a_2 & -a_3 & \dots & -a_{n-1} \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u(t)$$

$$y = \begin{bmatrix} b_0 & b_1 & b_2 & \dots & b_{n-1} \end{bmatrix} x(t)$$

- The input only directly affects one state, and the output is a linear combination of the state coordinates.

- Other versions of this form can be found in the literature by renaming the state in opposite order.



Observer Canonical Form

- Consider a system described by

$$\frac{d^n y}{dt^n} + a_{n-1}\frac{d^{n-1}y}{dt^{n-1}} + \cdots + a_1\frac{dy}{dt} + a_0 y$$

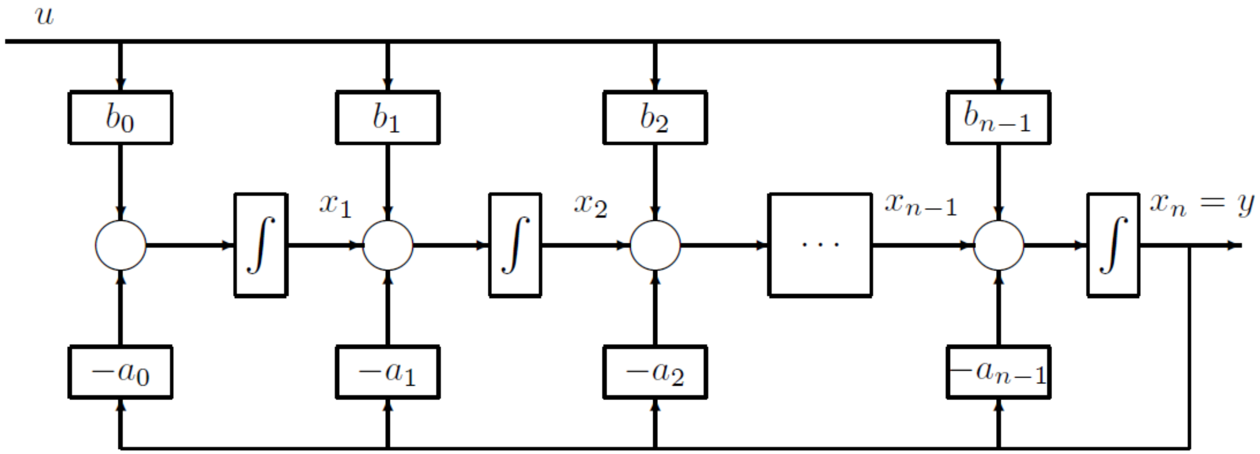
$$= b_m\frac{d^m u}{dt^m} + b_{m-1}\frac{d^{m-1}u}{dt^{m-1}} + \cdots + b_1\frac{du}{dt} + b_0 u$$

- Then its observer canonical form is given by

$$\dot{x}(t) = \begin{bmatrix} 0 & 0 & 0 & 0 & \dots & 0 & -a_0 \\ 1 & 0 & 0 & 0 & \dots & 0 & -a_1 \\ 0 & 1 & 0 & 0 & \dots & 0 & -a_2 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 0 & -a_{n-2} \\ 0 & 0 & 0 & 0 & \dots & 1 & -a_{n-1} \end{bmatrix} x(t) + \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_{n-2} \\ b_{n-1} \end{bmatrix} u(t)$$

$$y = [0 \quad 0 \quad 0 \quad \dots \quad 0 \quad 1]\, x(t)$$

- The output is one of the state coordinates and the input may directly affect the dynamic of all states.
- Other versions of this form can be found in the literature by renaming the state in opposite order.



Modal Form

- The modal form of a system is obtained when the matrix A is represented by its diagonal form.
- This makes the modes of the system (its natural exponential terms) explicit.

Definition: The matrix $A \in \mathbb{R}^{n\times n}$ is said to be diagonalizable if there exist a diagonal matrix $\Lambda \in \mathbb{C}^{n\times n}$ and a non-singular matrix $V \in \mathbb{C}^{n\times n}$ such that:

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

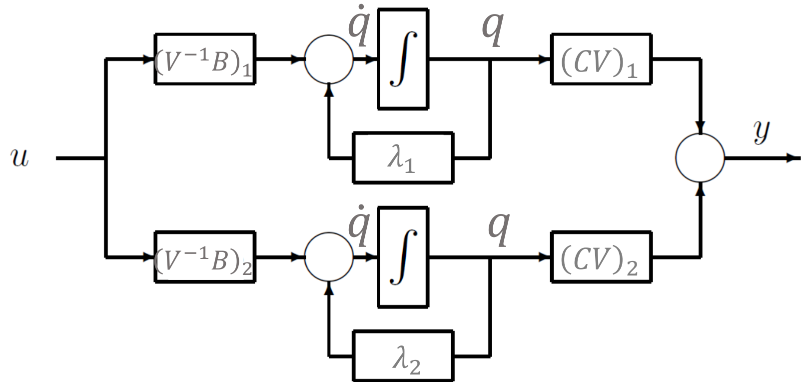
- The diagonal elements of Λ, λ_i, are called eigenvalues of the matrix A and they satisfy $\det(A - \lambda_i I) = 0 \; for \; i = 1,2,\ldots,n$
- The column vectors of V are the eigenvectors of the matrix A.

- Consider a SS representation, where matrix A has n different eigenvalues.

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du$$

- Applying the change of variable $x = Vq$, we obtain

$$\dot{q} = \Lambda q + V^{-1}Bu$$
$$y = CVq + Du$$



Solution of a SS representation

Autonomous System

- Let a system to be defined as

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t), \qquad \mathbf{x}(0) = \mathbf{x}_0$$

- Has a unique solution given by

$$\mathbf{x}(t) = e^{At}\mathbf{x}_0, \qquad \forall t > 0$$

Proof: Let's assume that the solution of the system is $\mathbf{x}(t) = e^{At}\mathbf{x}_0$, knowing that $\mathbf{x}(0) = e^{0}\mathbf{x}_0 = \mathbf{x}_0$, then

$$\dot{\mathbf{x}}(t) = \frac{d(e^{At})}{dt}\mathbf{x}_0 = Ae^{At}\mathbf{x}_0 = A\mathbf{x}(t)$$

- (The state-transition matrix). From the solution of the matrix ODE the matrix e^{At} is referred to as the state-transition matrix.

- From any instant t0 up to the instant $t_0 + t$, the states are related by

$$\mathbf{x}(t_0 + t) = e^{A(t-t_0)}\mathbf{x}(t_0)$$

Solution of a SS representation

Autonomous System

- The solution

$$\mathbf{x}(t) = e^{At}\mathbf{x}_0, \qquad \forall t > 0$$

- If A has n linearly independent eigenvectors $v_1, \dots, v_n$ with eigenvalues $\lambda_1, \dots, \lambda_n$ then the general solution is a **linear combination**:

$$x(t) = c_1 e^{\lambda_1 t} v_1 + c_2 e^{\lambda_2 t} v_2 + \cdots + c_n e^{\lambda_n} v_n$$

- where constants c_i are determined by the initial condition $x(0) = x_0$.

- If λ is real: motion is exponential growth/decay along eigenvector v.

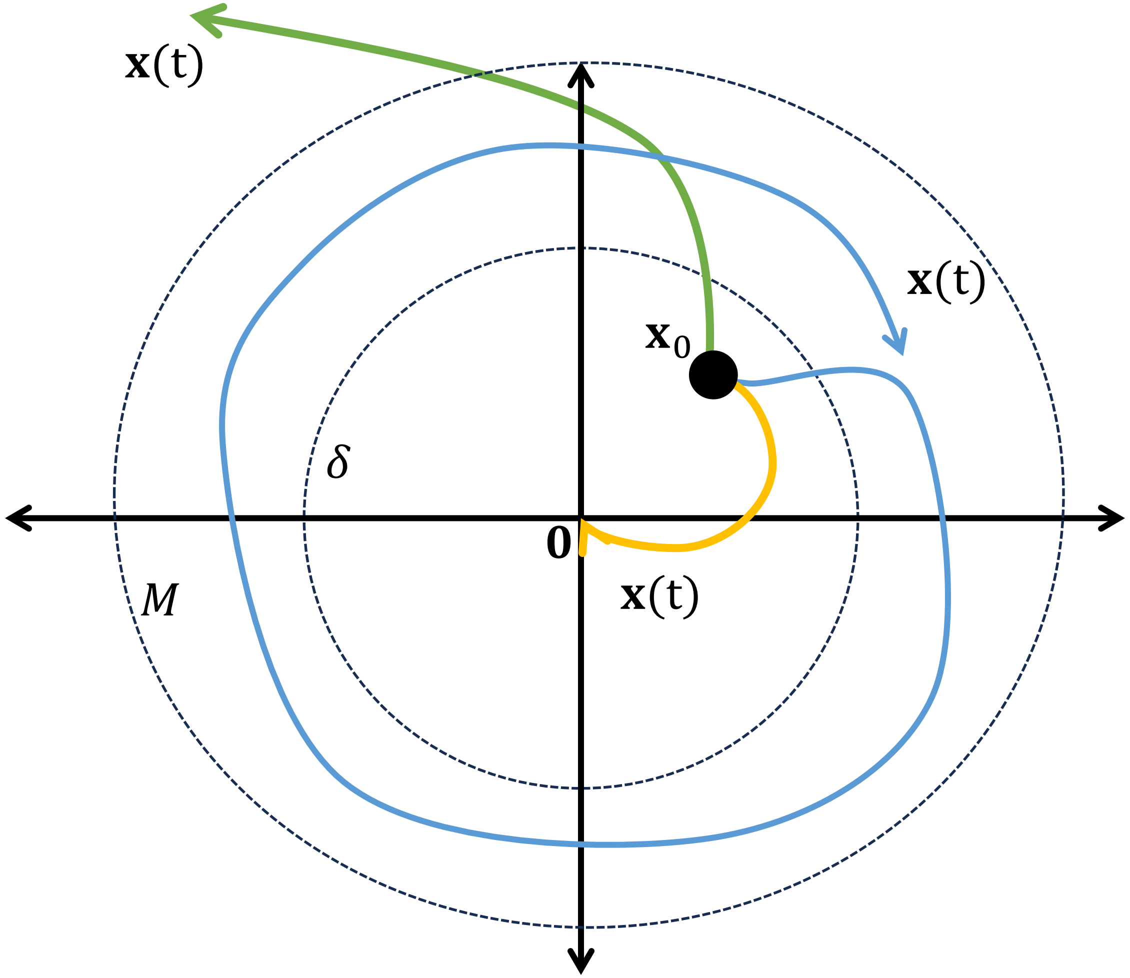
- If $\lambda = \alpha \pm j\beta$ is complex: eigenvectors are complex, and give a real solution:

$$x(t) \sim e^{\alpha t}\,(c_1 \cos(\beta t) + c_2 \sin(\beta t))$$

Analysis of a SS representation

Stability

- An autonomous system $\dot{\mathbf{x}} = A\mathbf{x}$ is said to be **asymptotically stable** if $\lim_{t\to\infty} \mathbf{x}(t) = 0\,, \forall \mathbf{x}_0 \in \mathbb{R}^n$.
- It is said to be **marginally stable** if for any initial condition $\mathbf{x}_0 \in \mathbb{R}^n$ such that $\|\mathbf{x}_0\| < \delta$, there exists $M \in \mathbb{R}$ such that $\|\mathbf{x}_0\| < M$.
- Finally, if the system is neither asymptotically stable nor marginal stable, it said to be **unstable**.



Analysis of a SS representation

- From the solution it can be seen that for an autonomous system $\dot{\mathbf{x}} = A\mathbf{x}$ is **asymptotically stable if all the eigenvalues of A have strictly negative real part**.
- The system is **marginally stable** if it has one or more distinct poles on the imaginary axis, and any remaining poles have negative real part.
- Finally, the system is **unstable** either if any pole has a positive real part, or any repeated poles on the imaginary axis.

Solution of a SS representation

Non-Autonomous System

- By analogy, the solution of a non-homogeneous ODE is given by the addition of the solution of the homogeneous case, i.e. the autonomous case, plus the particular solution.
- Let a system to be defined as

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\boldsymbol{u}(t); \;\; \mathbf{x}(0) = \mathbf{x}_0$$

- Has a unique solution given by

$$\mathbf{x}(t) = e^{\mathrm{A}t}\mathbf{x}_0 + \int_0^t e^{\mathrm{A}(t-\tau)}\mathbf{B}\boldsymbol{u}(\tau)d\tau\,, \qquad \forall t > 0$$

- The output then is

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x} + \mathbf{D}\boldsymbol{u}(t)$$

Proof: Rewrite the system as $\dot{\mathbf{x}}(t) - \mathbf{A}\mathbf{x}(t) = \boldsymbol{B}\boldsymbol{u}(t)$.

Multiply both sides by $e^{-t\mathbf{A}}$

$$e^{-t\mathrm{A}}[\dot{\mathbf{x}}(t) - \mathbf{A}\mathbf{x}(t)] = \mathrm{e}^{-t\mathrm{A}}\mathbf{B}\boldsymbol{u}(t)$$

Which is equivalent to

$$\frac{d}{dt}\left[e^{-t\mathbf{A}}\mathbf{x}(t)\right] = e^{-t\mathbf{A}}\mathbf{B}\boldsymbol{u}(t)$$

Finally integrating over $[0, t]$

$$e^{-t\mathbf{A}}\mathbf{x}(t) = e^{-0\mathbf{A}}\mathbf{x}_0 + \int_0^t e^{-\tau\mathbf{A}}\mathbf{B}\boldsymbol{u}(\tau)d\tau$$

Multiplying by $e^{t\mathrm{A}}$ the result is obtained.

Solution of a SS representation

- There are many ways to solve the SS representations numerically and algebraically.
- Some of the most used ones are Laplace transforms as follows
- For **homogeneous systems** the laplace solution can be computed as

$$\boldsymbol{X}(s) = (s\boldsymbol{I} - \mathbf{A})^{-1}\boldsymbol{X}(0)$$

- where $\mathrm{e}^{\mathrm{A}t} = \mathcal{L}^{-1}[(s\boldsymbol{I} - \mathbf{A})^{-1}]$ the demonstration is out of the scope of this lecture.

- For **non-homogeneous systems**

$$\boldsymbol{X}(s) = (s\boldsymbol{I} - \mathbf{A})^{-1}[\boldsymbol{X}(0) + \boldsymbol{B}\boldsymbol{U}(s)]$$

The demonstration of this result is out of the scope of this lecture.

- Another type of solution are numerical simulations, to be seen in the next section.

Example

Linear system
Mass Spring in SS

{Learn, Create, Innovate};

Manchester **Robotics**

State-space representation of a linear system

Worked example

- Let us define the set of states as:

$$\begin{aligned} x_1 &= y \\ x_2 &= \dot{y} \end{aligned} \tag{21}$$

- Then we can find a state-space representation of this system. From the definition of both coordinates, it is trivial that $\dot{x}_1 = x_2$, then (20) can be rewritten in term of x_1, x_2, and $\dot{x}_2$.

$$\dot{x}_2 + \frac{\beta}{m}\, x_2 + \frac{k}{m} x_1 = \frac{F}{m} \tag{22}$$

$$\ddot{y} + \frac{\beta}{m}\, \dot{y} + \frac{k}{m} y = \frac{F}{m} \tag{23}$$

- As a result, the system is described by two first order differential equations:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -\dfrac{\beta}{m}\, x_2 - \dfrac{k}{m} x_1 + \dfrac{1}{m} F \end{cases} \tag{24}$$

State-space representation of a linear system

Worked example

- We rewrite these two equations using matrices and the state $\boldsymbol{x} = (x_1, x_2)$.

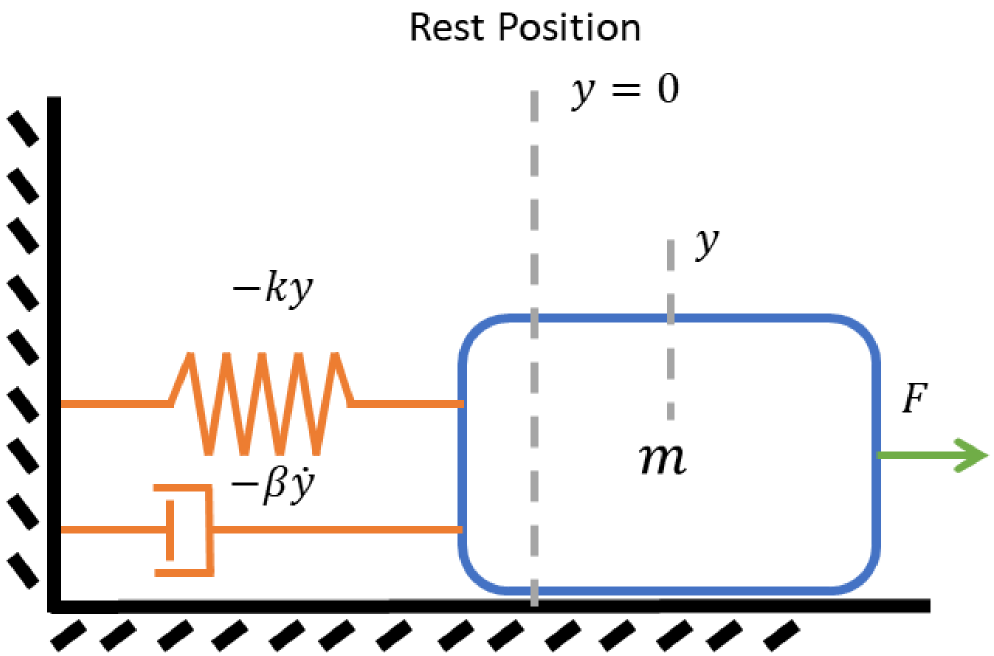
$$\dot{x}_1 = 0x_1 + x_2 + 0F \tag{25}$$

$$\dot{x}_2 = -\frac{\beta}{m}x_2 - \frac{k}{m}x_1 + \frac{1}{m}F \tag{26}$$

$$\dot{\boldsymbol{x}} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{\beta}{m} \end{bmatrix}\boldsymbol{x} + \begin{bmatrix} 0 \\ 1/m \end{bmatrix}F \tag{27}$$

- Using (21), the output equation is given by:

$$y = x_1 + 0x_2 + 0F = \begin{bmatrix} 1 & 0 \end{bmatrix}\boldsymbol{x} + 0F \tag{28}$$



State-space representation of a linear system

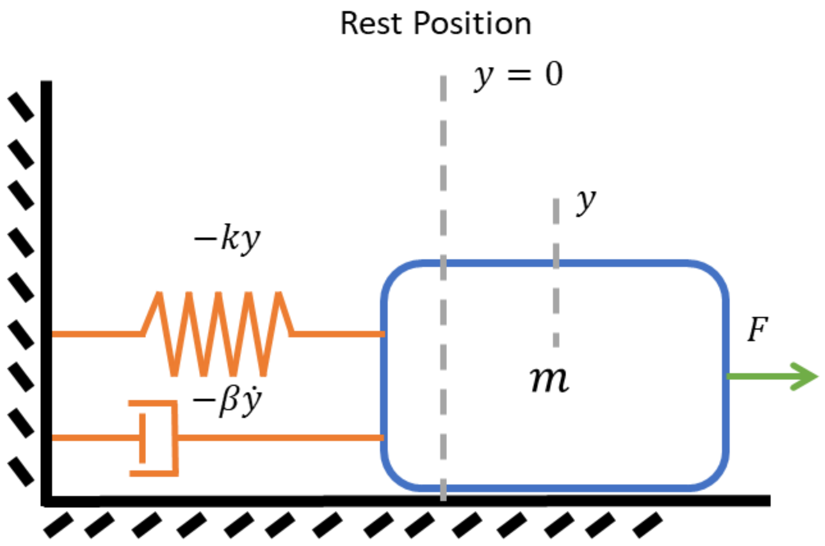
Worked example

In summary, the state-space representation of an ideal mass-spring-damper is given by:

$$A = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{\beta}{m} \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 1/m \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 \end{bmatrix} \qquad D = 0$$

$$\boxed{\begin{aligned} \dot{\boldsymbol{x}} &= \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{\beta}{m} \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} 0 \\ 1/m \end{bmatrix} F \\ y &= \begin{bmatrix} 1 & 0 \end{bmatrix} \boldsymbol{x} \end{aligned}} \qquad (29)$$



{Learn, Create, Innovate};

Discrete-time dynamic models

Introduction

Manchester Robotics

Discrete-time dynamic models

- A digital computer by its very nature, deals internally with discrete-time data or numerical values of functions at equally spaced intervals determined by the sampling period.
- Thus, discrete-time models such as difference equations are widely used in computer control applications.
- One way a continuous-time dynamic model can be converted to discrete-time form is by employing a finite difference approximation.

- Consider a nonlinear differential equation

$$\frac{dy(t)}{dt} = f(y,u) \tag{30}$$

where y is the output variable and u is the input variable.

Discrete-time dynamic models

- A This equation can be numerically integrated (for instance using Euler method) by introducing a finite difference approximation for the derivative.
- For example, the frst-order, backward difference approximation to the derivative at $t = k\Delta t$ is:

$$\frac{dy(t)}{dt} \cong \frac{y(k) - y(k-1)}{\Delta t} \qquad (31)$$

where Δt is the integration interval (the control engineers name it sampling time) specified by the user and $y(k)$ denotes the values of $y(k)$ at $t = k\Delta t$.

- So,

$$\frac{y(k) - y(k-1)}{\Delta t} \cong f(y(k-1), u(k-1)) \qquad (32)$$

- or:

$$y(k) = y(k-1) + \Delta t\, f(y(k-1), u(k-1)) \qquad (33)$$

- This is a first-order difference equation that can be used to predict $y(k)$ based on information at the previous time step $(k-1)$. This type of expression is called a recurrence relation.

Discrete-time dynamic models

- For higher-order ODEs, we can use a generalisation of the Euler method that we used for solving first-order ODEs. To illustrate the method, let us consider a 2nd order ODE:

$$\frac{d^2\, y(t)}{dt^2} = f(t, y, \frac{dy(t)}{dt}) \qquad (34)$$

- Or:

$$\ddot{y} = f(t, y, \dot{y}) \qquad (35)$$

- For discretization, the idea is to write the second order system (ODE) as a system of two first order systems (ODEs) and then apply Euler's method to the first order equations.
- So, as we did before, we'll define a new variable:

$$\begin{cases} y = x_1 \\ \dot{y} = x_2 = \dot{x}_1 \end{cases} \qquad (36)$$

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = f(t, x_1, x_2) \end{cases} \qquad (37)$$

- We need initial conditions:

For Newton the initial conditions are the initial position and initial velocity.

$$\begin{cases} x_1(t_0) = 0 \\ x_2(t_0) = 0 \end{cases} \qquad (38)$$

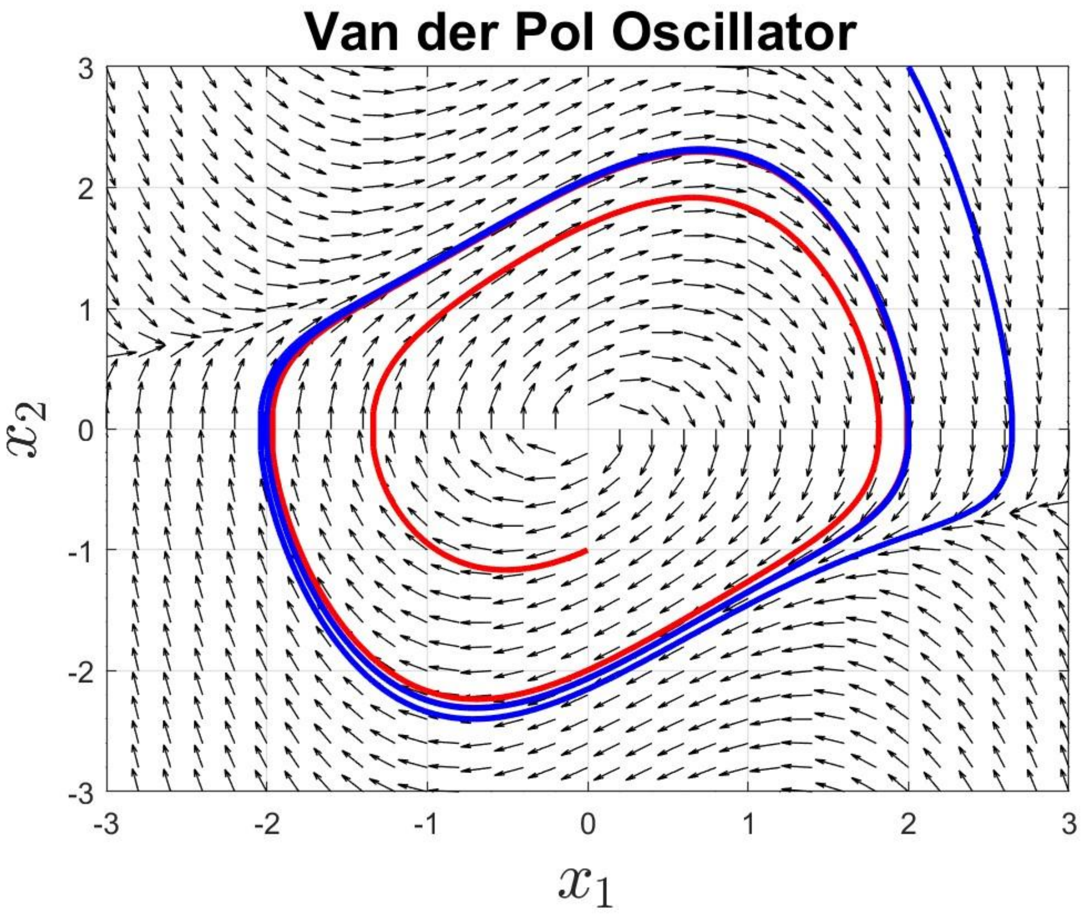
Discrete-time dynamic models

- Now, the idea is to solve both x_1 and x_2 simultaneously using Euler's method for both first order ODEs:

$$\begin{cases} \dfrac{x_1(k) - x_1(k-1)}{\Delta t} = x_2(k-1) \\ \dfrac{x_2(k) - x_2(k-1)}{\Delta t} = f(t, x_1, x_2) \end{cases} \tag{39}$$

$$\begin{cases} x_1(k) = x_1(k-1) + \Delta t\, x_2(k-1) \\ x_2(k) = x_2(k-1) + \Delta t\, f((k-1), x_1, x_2) \end{cases} \tag{40}$$

- This can be generalized to third order ODEs, or fourth order ODEs, as well as n order ODEs.



Example

Linear system
Mass Spring in SS

{Learn, Create, Innovate};

State-space representation of a linear system

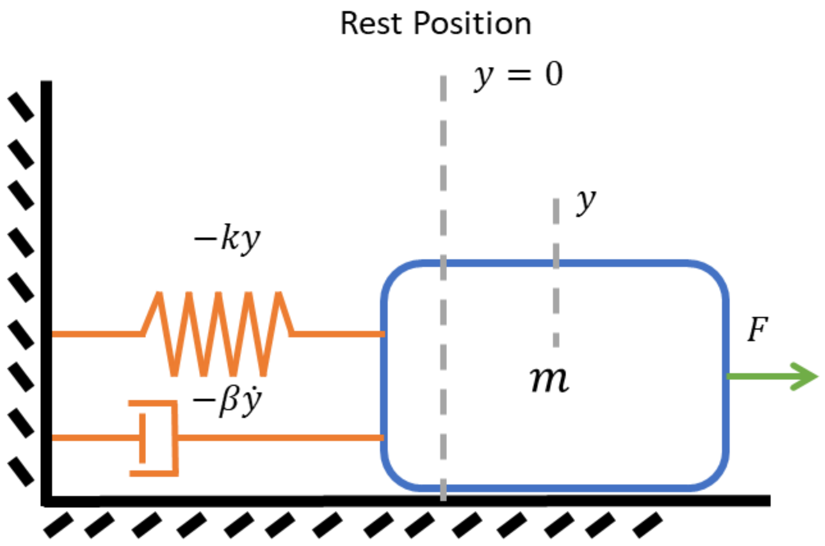
Worked example

In summary, the state-space representation of an ideal mass-spring-damper is given by:

$$A = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{\beta}{m} \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 1/m \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 \end{bmatrix} \qquad D = 0$$

$$\boxed{\begin{aligned} \dot{\boldsymbol{x}} &= \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{\beta}{m} \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} 0 \\ 1/m \end{bmatrix} F \\ y &= \begin{bmatrix} 1 & 0 \end{bmatrix} \boldsymbol{x} \end{aligned}} \qquad (29)$$



Solving ODEs in MATLAB

a) Making a simple ODE solver based on a numerical method such as Euler's method

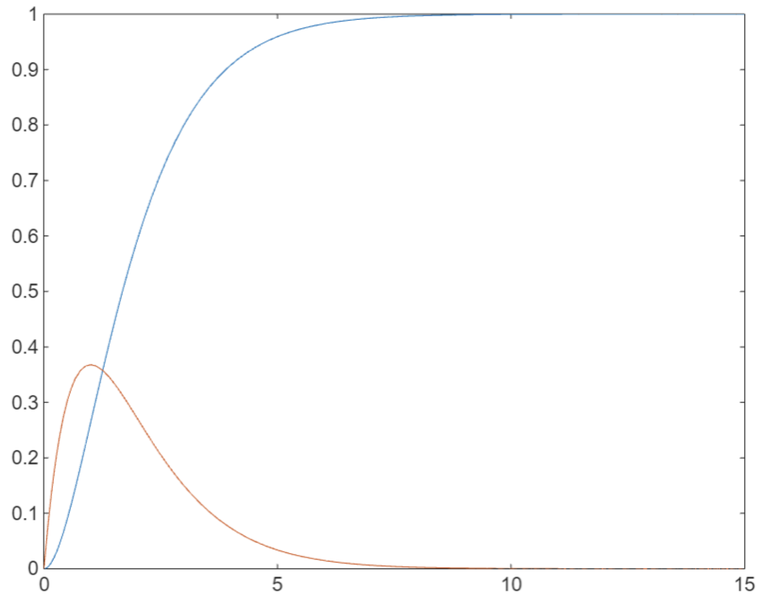
```
%% Parameters
b_damper=2.0;k_spring=1.0;mass=1.0;force=1.0;

%% Simulation Parameters
x1_0=0;x2_0=0;
dt=0.001;
tf=15

%% Euler Approximation configuration
t=0:dt:tf;
x1= zeros(length(t),1);
x2= zeros(length(t),1);
x1(1)=x1_0;x2(1)=x2_0;

%% Euler Approximation of the Solution
for k=1:length(t)-1
x1(k+1)=x1(k)+dt*(x2(k));
x2(k+1)=x2(k)+dt*(-(b_damper/mass)*x2(k)-(k_spring/mass)*x1(k)+(1/mass)*force);
end

%% Plotting
plot(t,x1,'LineWidth',3,'color','b')
hold
plot(t,x2,'LineWidth',3,'color','r')
```



Thank you

{Learn, Create, Innovate};

T&C

Terms and conditions

{Learn, Create, Innovate};

Manchester Robotics

Terms and conditions

- *THE PIECES, IMAGES, VIDEOS, DOCUMENTATION, ETC. SHOWN HERE ARE FOR INFORMATIVE PURPOSES ONLY. THE DESIGN IS PROPRIETARY AND CONFIDENTIAL TO MANCHESTER ROBOTICS LTD. (MCR2). THE INFORMATION, CODE, SIMULATORS, DRAWINGS, VIDEOS PRESENTATIONS ETC. CONTAINED IN THIS PRESENTATION IS THE SOLE PROPERTY OF MANCHESTER ROBOTICS LTD. ANY REPRODUCTION, RESELL, REDISTRIBUTION OR USAGE IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF MANCHESTER ROBOTICS LTD. IS STRICTLY PROHIBITED.*
- *THIS PRESENTATION MAY CONTAIN LINKS TO OTHER WEBSITES OR CONTENT BELONGING TO OR ORIGINATING FROM THIRD PARTIES OR LINKS TO WEBSITES AND FEATURES IN BANNERS OR OTHER ADVERTISING. SUCH EXTERNAL LINKS ARE NOT INVESTIGATED, MONITORED, OR CHECKED FOR ACCURACY, ADEQUACY, VALIDITY, RELIABILITY, AVAILABILITY OR COMPLETENESS BY US.*
- *WE DO NOT WARRANT, ENDORSE, GUARANTEE, OR ASSUME RESPONSIBILITY FOR THE ACCURACY OR RELIABILITY OF ANY INFORMATION OFFERED BY THIRD-PARTY WEBSITES LINKED THROUGH THE SITE OR ANY WEBSITE OR FEATURE LINKED IN ANY BANNER OR OTHER ADVERTISING.*