Industrial Control & Automation

Lecture 1: An Introduction

Outline

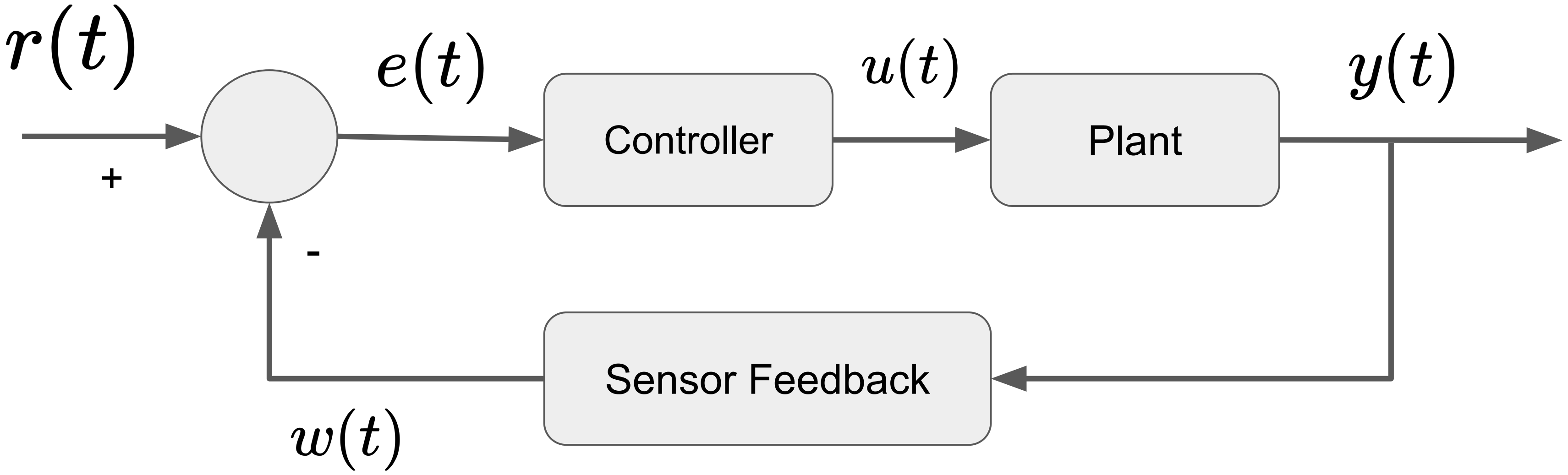
- Introduction to Industrial Automation
- History of Automation / Control Theory
- Four stages of Industrial Revolution
- Modeling of Dynamical Systems

Industrial Automation

Control Related Vocabulary

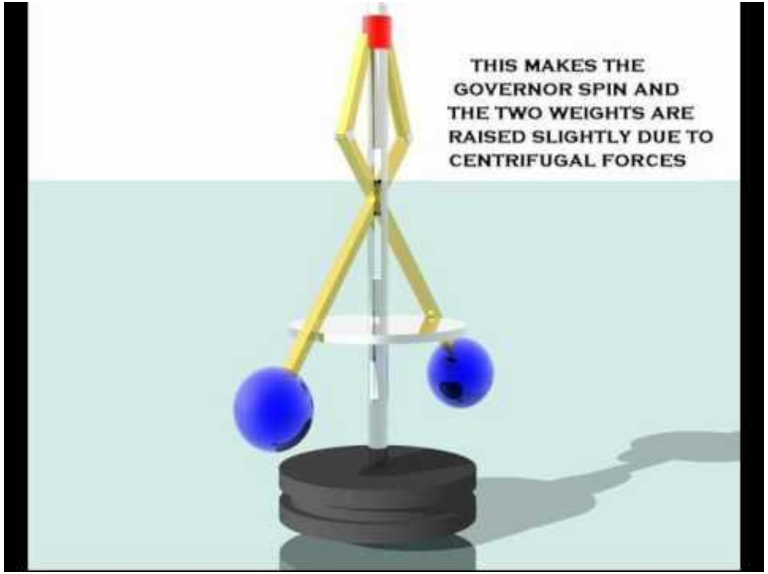
- **Plant** - A piece of equipment, or a set of machine parts performing together.
- **Process** - A natural, progressively continuing operation or development marked by series of gradual changes that succeed one another in a relatively fixed way and lead toward a particular result or end.
- **System** - A combination of components that act together and perform a certain objective - physical, biological, economic systems.
- **Disturbances** - A signal that tends to adversely affect the output or behaviour of a system. It could be an internal or external.
- **Feedback Control** - An operation that reduces the error between a given reference and system output so that a desired behaviour is obtained from a system.
- **Automation** - The control of an industrial process (manufacturing, production etc.) by automatic rather than manual means.

Open-Loop System

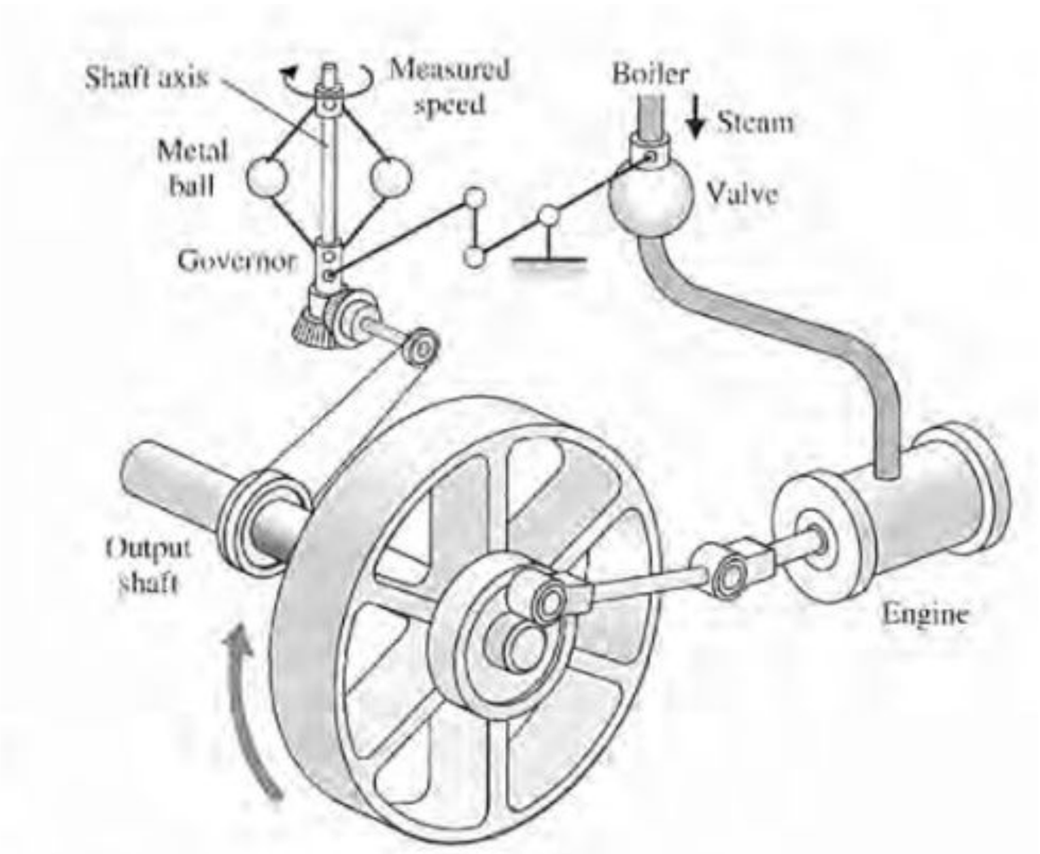


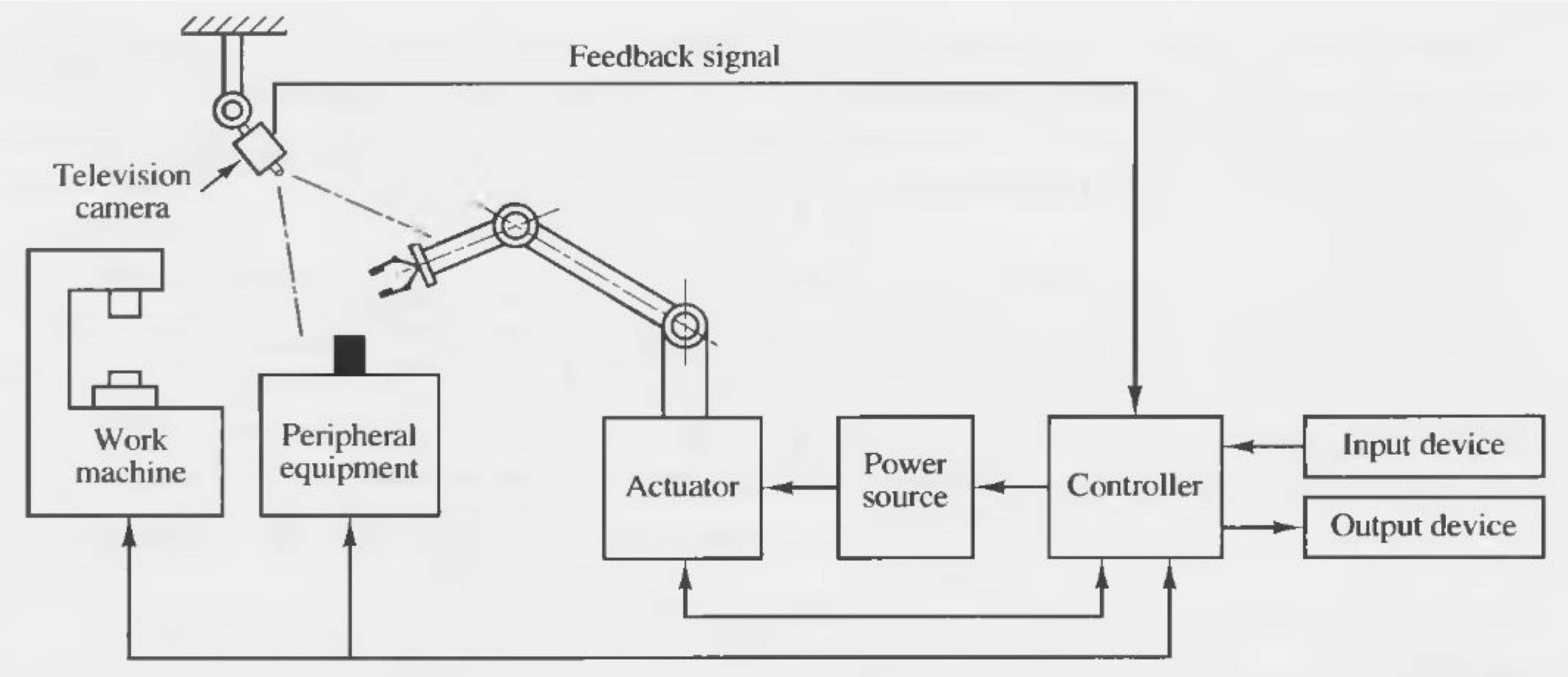
Closed-loop System

Examples of Feedback Control Systems

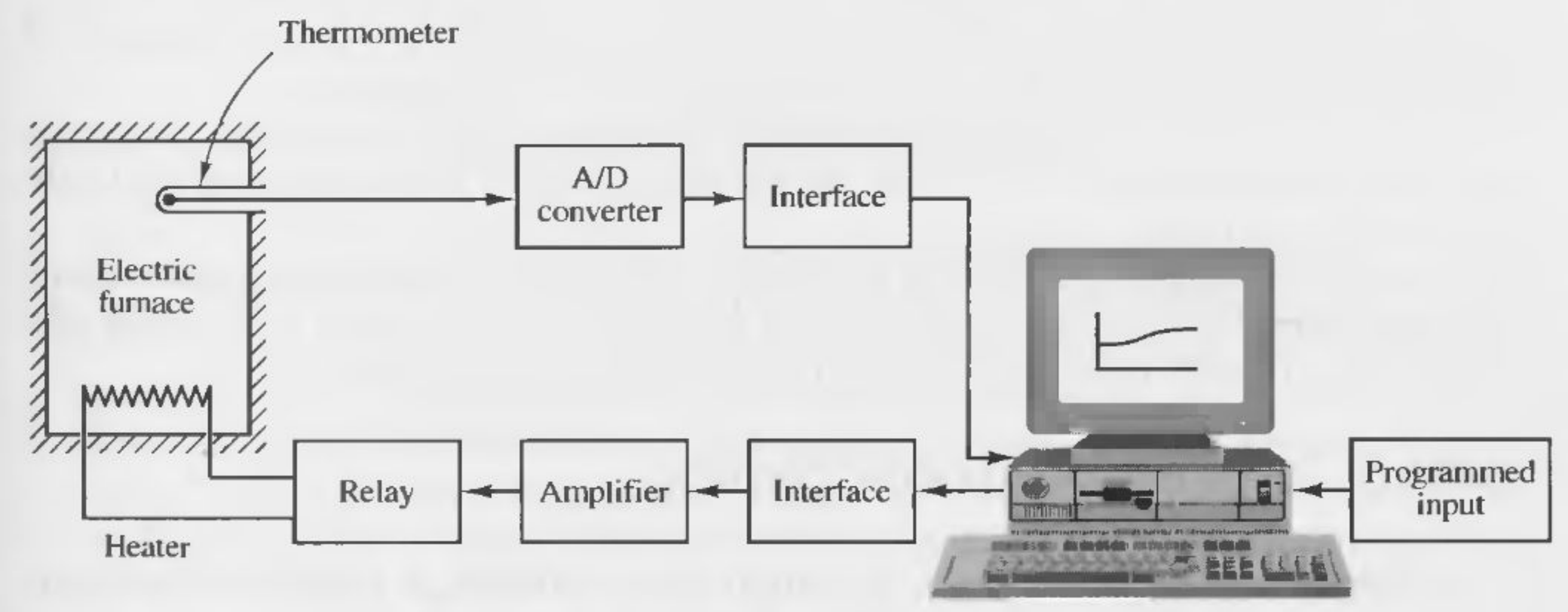


James Watt's Flyball Speed Governor
for Steam Engines





A vision-based robot Control System

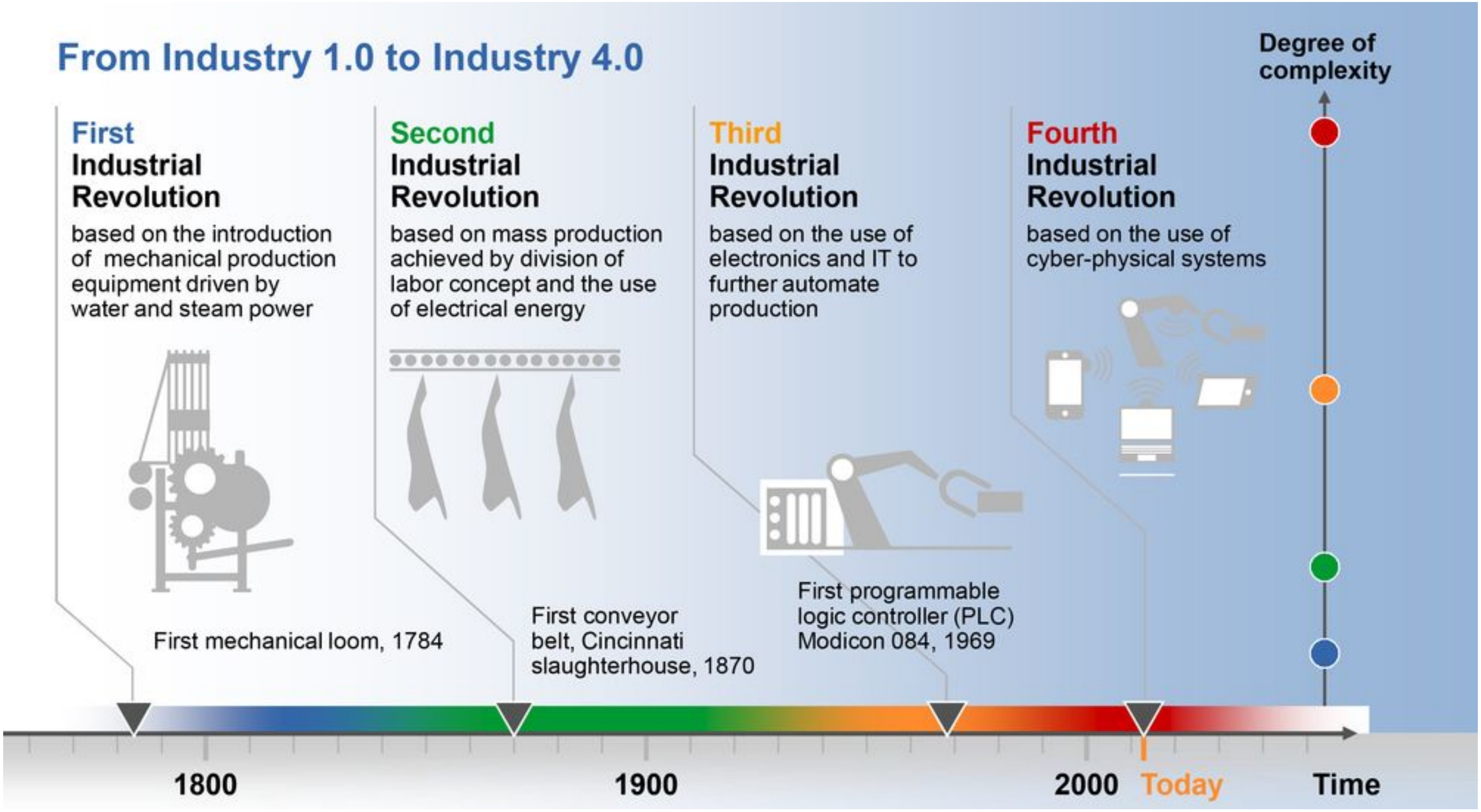


A Temperature Control System

Brief History of Control Theory / Automation

- 1788: James Watt's Speed Governor for controlling speed of steam engines.
- 1868 - J C Maxwell formulated mathematical theory of flyball governor by using differential equations.
- 1922: Automatic Control for steering Ships
- 1932: Nyquist developed a method to analyze the stability of closed-loop systems based on open-loop response to steady-state sinusoidal inputs.
- 1934: Hazen introduced the term servomechanism for position control systems.
- 1940-60: World War II given impetus to automatic control theory for missile-guidance system, aeronautical systems, radar, antennae, gun positioning system
- 1940: Frequency response methods made it possible for engineers to design linear closed-loop control systems.
- 1950: Root-locus methods were developed by Evans
- 1957: First artificial satellite, Sputnik, was launched by Russia leading to beginning of space age.
- Late 1950s: Optimal Control theory were developed
- 1960s: Multi-input Multi-output (MIMO) systems, modern control, state variable analysis were developed. Digital Computers were now available for more complex analysis.

- 1969: Apollo 11 - Neil Armstrong and Buzz Aldrin landed on Moon
- 1960-80: Adaptive and Robust control methods.
- 1983 - Introduction of Personal Computers (PC) - allowing engineers to develop control design software and simulation tools.
- 1990 - Export oriented manufacturing companies emphasizing automation.
- 1994 - Feedback control systems were widely used in automobiles.
- 1995 - Global Positioning System (GPS)
- 1997 - Mars Rover named ‘Sojourner’ was launched to explore martian surface.
- 1998-2003: Advances in micro/nano technologies
- 1998: International Space Station was launched with first long-term residents arriving in 2000.
- 2010-20: Self-driving Cars, Electrical Vehicles (EV)
- 2020: SpaceX Falcon 9 reusable launch vehicles.


From Industry 1.0 to Industry 4.0
Degree of complexity
First Industrial Revolution
based on the introduction of mechanical production equipment driven by water and steam power
Second Industrial Revolution
based on mass production achieved by division of labor concept and the use of electrical energy
Third Industrial Revolution
based on the use of electronics and IT to further automate production
Fourth Industrial Revolution
based on the use of cyber-physical systems
First mechanical loom, 1784
First conveyor belt, Cincinnati slaughterhouse, 1870
First programmable logic controller (PLC) Modicon 084, 1969
1800
1900
2000
Today
Time

What this course is all about?

- We will study methods for modeling and analyzing the systems.
- Then we will study methods to design controller so that we can obtain desired response from these systems.
- In the process, we will study, analyze and examine various real-world systems.
- Hopefully, this will give you a better understanding of what goes behind creating an automated system which when scaled may lead to the creation of an automated factory (as we saw in the first picture in this session).

Modeling of Dynamical Systems

- Most of the physical systems are ***dynamical systems*** which are represented by differential Equations.

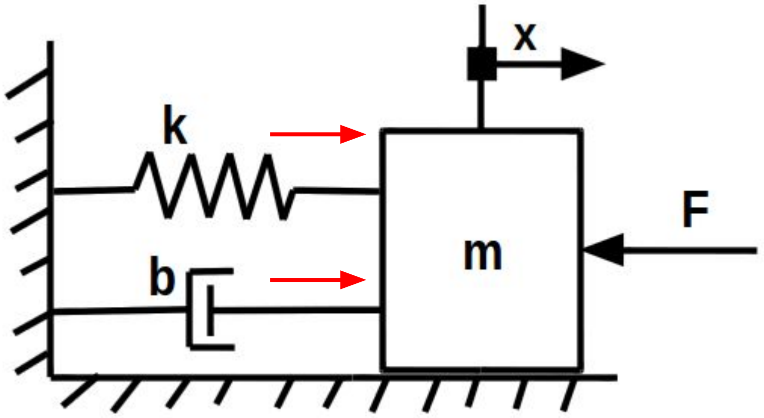
- System modeling primarily involves the following steps:
    - The first step is to define various components of a system and assumptions.
    - Derive mathematical relationships for these components based on basic principles.
    - Obtain differential equations representing the mathematical model of the complete system.
    - Solve the differential equations to obtain desired output variables.
    - Analyze the system behaviour and if required, revisit the assumptions, modify the components and its parameter to improve the output behaviour.

- We will study some examples:
    - Spring-mass-damper system
    - Electrical RLC Circuit

Spring-Mass-Damper System

$$Net\ Force = mass \times acceleration$$

$$\begin{aligned} \text{position} &= x(t) \\ \text{velocity} &= \dot{x}(t) \\ \text{acceleration} &= \ddot{x}(t) \end{aligned}$$

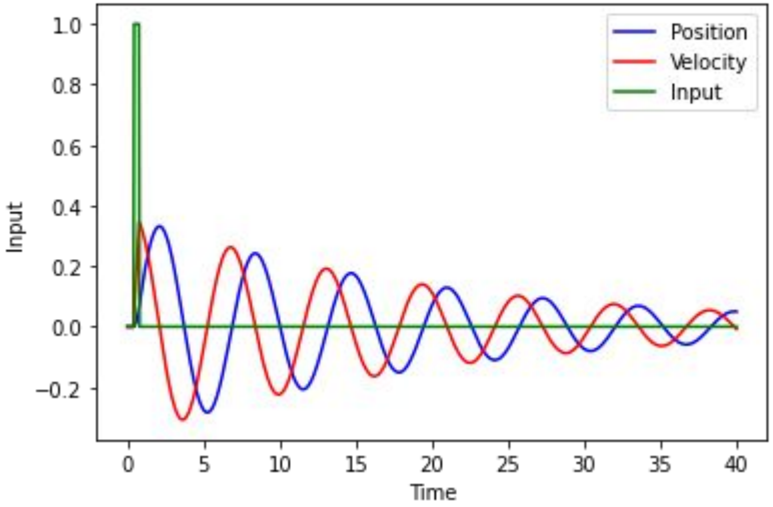


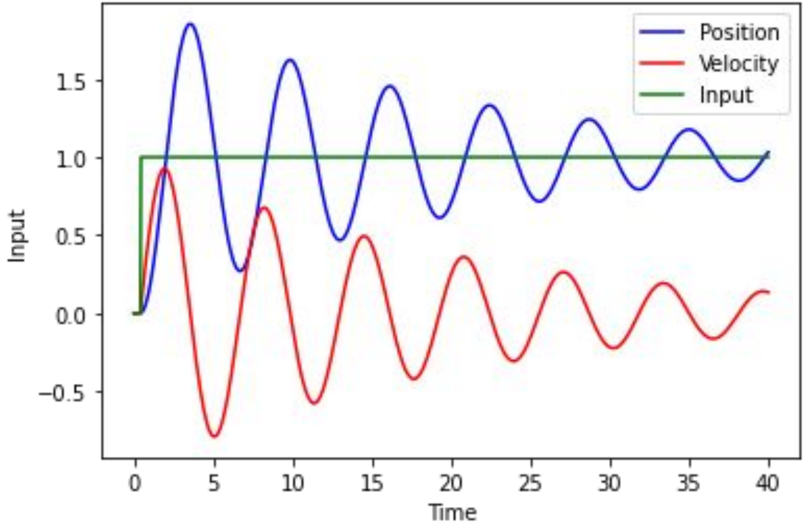
$$\boxed{\begin{aligned} m\frac{d^2x(t)}{dt^2} + b\frac{dx(t)}{dt} + kx(t) = F(t) \\ m\ddot{x} + b\dot{x} + kx = F \end{aligned}}$$

Where b is the damping coefficient and k is spring constant.

Obtaining System response by solving ODEs

Mass = 1Kg
B = 0.1
K = 1.0
Input1 = Pulse of width 1 second = u(t) - u(t-1)
Input2 =Unit Step

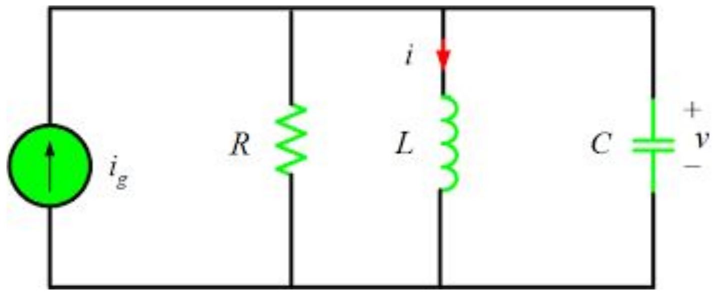




$$
\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= -bx_2 - kx_1 + F(t)
\end{aligned}
$$

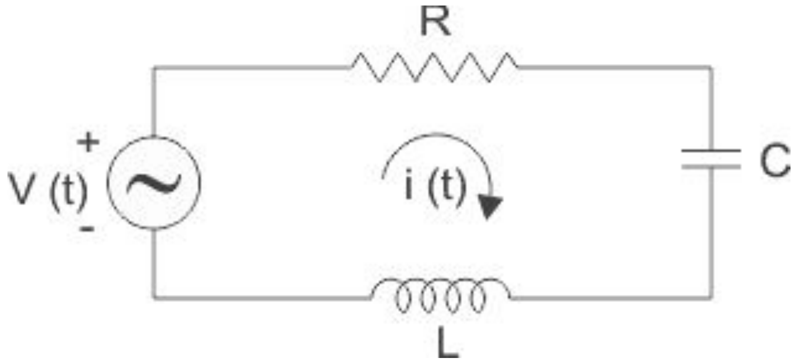
Position and velocity oscillate before attaining the steady state.

Electrical RLC Circuit



$$\frac{v(t)}{R} + \frac{1}{L}\int_0^t v(t)dt + C\frac{dv(t)}{dt} = i_s(t) = r(t)$$

$$C\ddot{v}(t) + \frac{1}{R}\dot{v}(t) + \frac{1}{L}v(t) = \dot{r}(t)$$

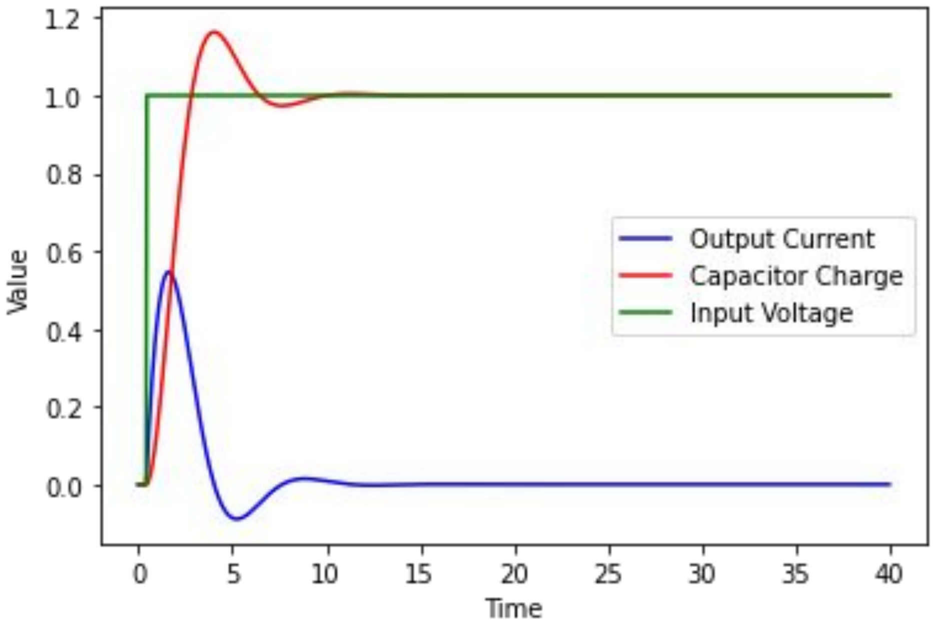


$$i(t)R + L\frac{di(t)}{dt} + \frac{1}{C}\int_0^t i(t)dt = v(t)$$

$$L\ddot{q}(t) + R\dot{q}(t) + \frac{1}{C}q(t) = v(t)$$

Series RLC Circuit

Parameters:
R = 1 Ohm
L = 1 Henry
C = 1 Farad
Input = Unit step function = u(t-1)



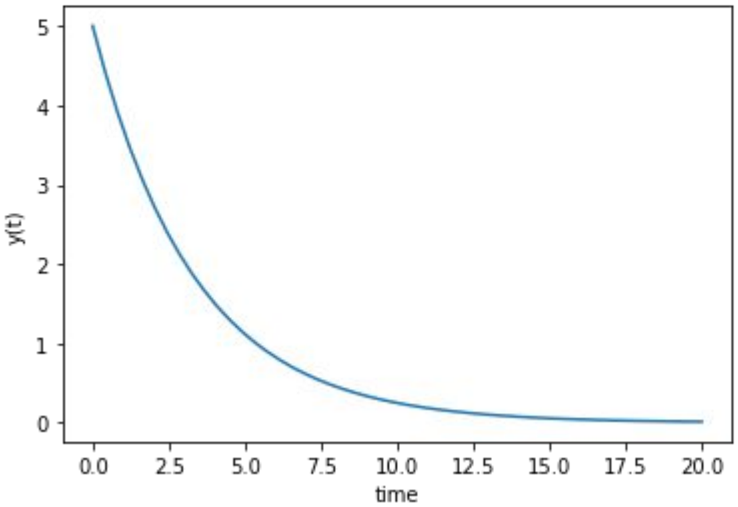
The Current in the circuit falls to zero and the capacitor gets charged to 1.0.

$$
\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= -\frac{R}{L}x_2 - \frac{1}{LC}x_1 + \frac{v(t)}{L}
\end{aligned}
$$

ODESolvers in Python

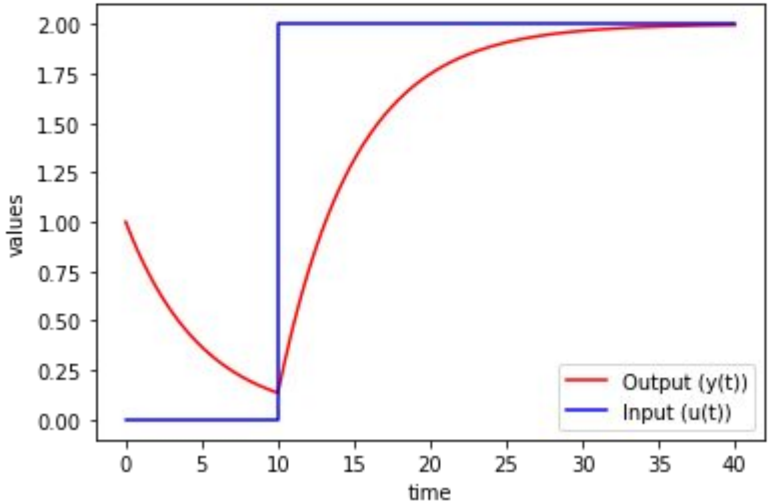
Example 1

```python
import numpy as np
from scipy.integrate import odeint
import matplotlib.pyplot as plt

# function that returns dy/dt
def model(y,t):
    k = 0.3
    dydt = -k * y
    return dydt

# initial condition
y0 = 5

# time points
t = np.linspace(0,20)

# solve ODE
y = odeint(model,y0,t)

# plot results
plt.plot(t,y)
plt.xlabel('time')
plt.ylabel('y(t)')
plt.show()
```



$$\dot{y} = -0.3y;\ y(0) = 5$$

```python
import numpy as np
from scipy.integrate import odeint
import matplotlib.pyplot as plt

# function that returns dy/dt
def model(y,t):
    # u steps from 0 to 2 at t=10
    if t<10.0:
        u = 0
    else:
        u = 2
    dydt = (-y + u)/5.0
    return dydt

# initial condition
y0 = 1

# time points
t = np.linspace(0,40,1000)

# solve ODE
y = odeint(model,y0,t)

# plot results
plt.plot(t,y,'r-',label='Output (y(t))')
plt.plot([0,10,10,40],[0,0,2,2],'b-',label='Input (u(t))')
plt.ylabel('values')
plt.xlabel('time')
plt.legend(loc='best')
plt.show()
```



$$\dot{y} = \frac{dy}{dt} = \frac{(-y+u)}{5};\ y(0) = 1$$

$$u(t) = 2u(t-10)$$

Example 2

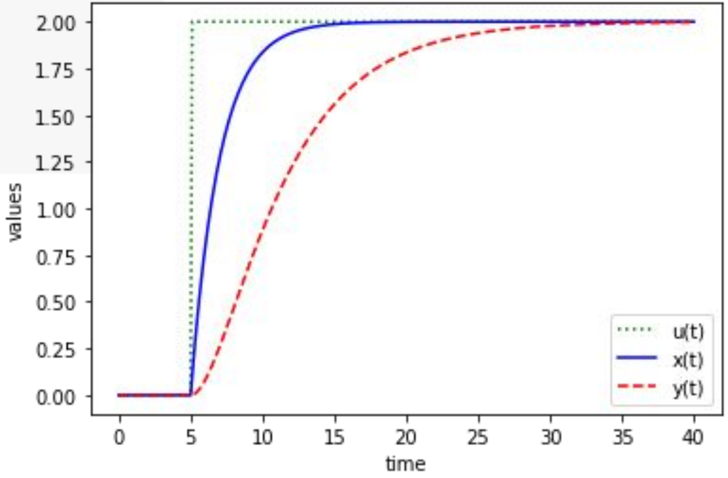
```python
import numpy as np
from scipy.integrate import odeint
import matplotlib.pyplot as plt

# function that returns dz/dt
def model(z,t,u):
    x = z[0]
    y = z[1]
    dxdt = (-x + u)/2.0
    dydt = (-y + x)/5.0
    dzdt = [dxdt,dydt]
    return dzdt

# initial condition
z0 = [0,0]

# number of time points
n = 401

# time points
t = np.linspace(0,40,n)

# step input
u = np.zeros(n)
# change to 2.0 at time = 5.0
u[51:] = 2.0

# store solution
x = np.empty_like(t)
y = np.empty_like(t)
# record initial conditions
x[0] = z0[0]
y[0] = z0[1]

# solve ODE
for i in range(1,n):
    # span for next time step
    tspan = [t[i-1],t[i]]
    # solve for next step
    z = odeint(model,z0,tspan,args=(u[i],))
    # store solution for plotting
    x[i] = z[1][0]
    y[i] = z[1][1]
    # next initial condition
    z0 = z[1]

# plot results
plt.plot(t,u,'g:',label='u(t)')
plt.plot(t,x,'b-',label='x(t)')
plt.plot(t,y,'r--',label='y(t)')
plt.ylabel('values')
plt.xlabel('time')
plt.legend(loc='best')
plt.show()
```

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} (-x+u)/2 \\ (-y+x)/5 \end{bmatrix}$$

$$\{x(0), y(0)\} = (0,0)$$

$$u(t) = 2u(t-5)$$

Example 3



Summary

- Industrial Automation is all about automating various processes in Industries by using machines thereby reducing human effort.

- In order to build such automatic machines, we need to understand how various machines or “Systems” work and how to “Control” them to achieve desired goals.

- Control Systems is about studying methods for modeling, analyzing and controlling physical systems.

- Most of the physical systems of interest are dynamical in nature represented by differential equations.