Simulation and Modeling
Continuous System

Er. Narayan Sapkota, M.Sc.
narayan.sapkota@eemc.edu.np

Everest Engineering College (Senior Lecturer)
Kathmandu University (Visiting Faculty)

January 5, 2025

Syllabus

1. Continuous System Simulation and System Dynamics
2. Continuous System Models
3. Differential Equations
4. Analog Computers
5. Hybrid Computers
6. Digital Analog Simulators
7. Continuous System Simulation Language
8. CSMP III
9. Feedback System
10. Interactive System
11. Real Time Simulation
12. Predator-Prey Model

Table of Contents

System Dynamics (1)

- System dynamics is a methodology for studying and analyzing complex systems by focusing on the interactions between various components or subsystems over time.
- It involves constructing causal loop diagrams and stock-and-flow diagrams to represent feedback loops and accumulations within the system.
- Loop diagrams, also known as causal loop diagrams (CLDs), are a fundamental tool used in system dynamics to represent the feedback structures within a system. They visually depict how variables in a system are interconnected and how they influence one another over time. These diagrams show causal relationships between different elements of a system, capturing the essence of feedback loops that drive the system's behavior.

System Dynamics (2)

- Stock-and-flow diagrams visually depict the stocks (quantities that accumulate over time) and the flows (rates at which stocks change) within a system.Stocks are the accumulations or quantities that change over time. They represent the "state" of the system at any given moment and are typically shown as rectangles or boxes in a diagram.
  - Inventory (how much stock is currently on hand)
  - Population (total number of individuals in a population)
  - Bank account balance (total money in the account)
- Flows are the rates at which stocks change over time. They represent the movement into or out of a stock. Flows are typically shown as arrows connected to the stocks, with the arrows indicating the direction of the flow.
  - Production rate (rate at which new products are made)
  - Birth rate (rate at which new individuals are born)
  - Expenditure (rate at which money is spent)

System Dynamics (3)

- These diagrams allow analysts to understand how changes in one part of the system can influence other parts and how these effects evolve and propagate over time.
- System dynamics is widely used in fields such as business management, public policy, and environmental science to model and analyze dynamic systems, including supply chains, economic markets, and ecological systems.
- For example, in business management, system dynamics can be used to model the interactions between sales, production, and inventory levels in a manufacturing company.
- This approach helps identify potential bottlenecks and optimize operational efficiency by simulating the effects of various decisions over time.

Table of Contents

Continuous System Models (1)

Continuous system models are the systems where state variables change continuously over time. These systems can be represented by differential equations that describe the relationship between various components or states of the system. CSSLs are specifically designed to handle such systems.

- Nature of the System: Continuous systems model processes where variables change smoothly over time. Examples include fluid flow in pipes, electrical circuits, or population dynamics.
- Mathematical Representation: The models often involve differential equations or integral equations that describe how the system's state evolves over time.
- Execution: In continuous simulation, the state of the system is updated continuously and instantaneously based on the equations governing the system's dynamics.

Table of Contents

Differential Equations (1)

A differential equation is an equation that involves one or more functions and their derivatives. The derivatives represent the rates of change of these functions, and the differential equation expresses how the function behaves in relation to its rate of change. In simpler terms, a differential equation describes how a quantity changes with respect to another quantity, usually time or space.

Differential equations can be classified into two major categories:

1. Ordinary Differential Equations (ODEs): Involves functions of a single variable and its derivatives.
2. Partial Differential Equations (PDEs): Involves functions of multiple variables and their partial derivatives.

The general form of an ordinary differential equation is:

$$\frac{dy}{dx} = f(x, y)$$

Differential Equations (2)

Real-Time Examples of Differential Equations

1. Population Growth (Logistic Growth Model) The population of a species in an environment can be modeled using a differential equation that takes into account the rate of change of the population. The population grows at a rate proportional to the current population size, but as the population approaches a maximum limit (carrying capacity), growth slows down.

The logistic growth model is given by:

$$\frac{dP}{dt} = rP\left(1 - \frac{P}{K}\right)$$

Where:

- $P(t)$ is the population at time t,
- r is the growth rate,

- K is the carrying capacity of the environment.

2. Newton's Law of Cooling

This law describes the rate at which an object's temperature changes in response to the surrounding environment. The rate of change of the temperature of an object is proportional to the difference between its current temperature and the ambient temperature.

The differential equation is:

$$\frac{dT}{dt} = -k(T - T_{\text{ambient}})$$

Where:

- $T(t)$ is the temperature of the object at time t,
- T_{ambient} is the ambient temperature,

- k is a constant that depends on the properties of the object and the environment.

3. Motion of an Object (Simple Harmonic Motion)

The motion of an object attached to a spring (such as a mass oscillating on a spring) follows a differential equation based on Hooke's law. The acceleration of the object is proportional to its displacement from the equilibrium position, and this leads to simple harmonic motion.
The equation for this motion is:

$$\frac{d^2x}{dt^2} = -kx$$

Where:

- $x(t)$ is the displacement of the object at time t,

Differential Equations (5)

- k is a constant proportional to the stiffness of the spring.

4. Electric Circuit (RC Circuit)

In an electrical circuit with a resistor (R) and a capacitor (C), the voltage across the capacitor changes over time, and this rate of change can be described by a differential equation.

The equation governing the voltage $V(t)$ across the capacitor is:

$$\frac{dV}{dt} = -\frac{1}{RC}V$$

5. Spread of Disease (Epidemic Models)

Epidemiological models often use differential equations to describe how diseases spread through populations. One well-known model is the SIR (Susceptible-Infected-Recovered) model, which describes how individuals transition between these categories over time.

Differential Equations (6)

The system of differential equations for the SIR model is:

$$\frac{dS}{dt} = -\beta SI$$

$$\frac{dI}{dt} = \beta SI - \gamma I$$

$$\frac{dR}{dt} = \gamma I$$

Where:

- $S(t)$ is the number of susceptible individuals,
- $I(t)$ is the number of infected individuals,
- $R(t)$ is the number of recovered individuals,
- β is the infection rate,
- γ is the recovery rate.

Linear Differential Equation (1)

A linear differential equation is a differential equation in which the unknown function and its derivatives appear to the first power and are not multiplied by each other. In other words, a differential equation is linear if it does not involve nonlinear terms such as the square, cube, or other powers of the unknown function or its derivatives, nor products of the unknown function and its derivatives.

A general form of a linear ordinary differential equation (ODE) of order n is:

$$a_n(x)\frac{d^n y}{dx^n} + a_{n-1}(x)\frac{d^{n-1} y}{dx^{n-1}} + \cdots + a_1(x)\frac{dy}{dx} + a_0(x)y = f(x)$$

Where:

- y is the unknown function of x,
- $a_n(x), a_{n-1}(x), \ldots, a_1(x), a_0(x)$ are known functions of x,

Linear Differential Equation (2)

- $f(x)$ is a given function (may represent a non-homogeneous part of the equation).

The equation is linear because each term involves either the function y, or its derivatives, each raised to the first power, and no terms involve products or powers of y or its derivatives.

Classification of Linear Differential Equations

1. First-Order Linear Differential Equation

This type involves only the first derivative of the unknown function. The general form is:

$$\frac{dy}{dx} + p(x)y = q(x)$$

Where $p(x)$ and $q(x)$ are known functions of x.

Linear Differential Equation (3)

2. Higher-Order Linear Differential Equations

These involve higher derivatives of the unknown function. The general form is:

$$a_n(x)\frac{d^n y}{dx^n} + a_{n-1}(x)\frac{d^{n-1} y}{dx^{n-1}} + \cdots + a_1(x)\frac{dy}{dx} + a_0(x)y = f(x)$$

Non-Linear Differential Equation (1)

A non-linear differential equation is a differential equation in which the unknown function and its derivatives appear in non-linear forms. This means the equation involves terms where the unknown function or its derivatives are raised to a power greater than one, or appear in a product with each other.

In contrast to linear differential equations, non-linear differential equations can have terms like y^2, $\frac{dy}{dx} \cdot y$, or higher powers of $\frac{dy}{dx}$, etc. These non-linearities lead to more complex behavior and often make the solution methods more difficult.
A general form of a non-linear ordinary differential equation (ODE) is:

$$F\left(x, y, \frac{dy}{dx}, \frac{d^2y}{dx^2}, \dots\right) = 0$$

Where:

Non-Linear Differential Equation (2)

- y is the unknown function of x,
- The function F can include any terms that involve y, its derivatives, and possibly powers or products of these terms.

Classification of Non-Linear Differential Equations Non-linear differential equations can be classified into various types based on the order of the equation, the number of variables involved, and the form of non-linearity. Some common types include:

1. Non-linear first-order equations: These equations involve the first derivative of the unknown function, and the non-linearity may appear in the form of powers or products of the unknown function and its derivative.
2. Non-linear second-order equations: These equations involve the second derivative of the unknown function, and the non-linearity can appear in higher powers of the function or its derivatives.

Non-Linear Differential Equation (3)

3. Non-linear systems of equations: These involve more than one unknown function and may involve non-linear interactions between them.

Characteristics of Non-Linear Differential Equations

- Superposition principle does not apply: Unlike linear equations, the superposition principle (i.e., the sum of solutions is also a solution) does not hold for non-linear equations.
- Multiple solutions: Non-linear differential equations can have multiple, sometimes infinitely many, solutions, or no solution at all.
- Chaotic behavior: In some cases, non-linear differential equations can exhibit chaotic behavior, where small changes in initial conditions lead to dramatically different outcomes.
- Difficulty in analytical solutions: Non-linear equations are often more difficult to solve analytically and may require numerical methods or approximations.

Partial Differential Equation (1)

A Partial Differential Equation (PDE) is a differential equation that involves multiple independent variables and the partial derivatives of an unknown function with respect to those variables. The unknown function depends on two or more independent variables, and the equation expresses the relationship between the function and its partial derivatives.

In contrast to ordinary differential equations (ODEs), which involve derivatives with respect to a single variable, PDEs deal with multivariable functions. PDEs are fundamental in describing a wide variety of physical phenomena, such as heat conduction, wave propagation, and fluid dynamics.

The general form of a partial differential equation is:

$$F\left(x_1, x_2, \ldots, x_n, u(x_1, x_2, \ldots, x_n), \frac{\partial u}{\partial x_1}, \frac{\partial u}{\partial x_2}, \ldots, \frac{\partial^2 u}{\partial x_n}\right) = 0$$

Partial Differential Equation (2)

Where:

- $u(x_1, x_2, \ldots, x_n)$ is the unknown function of n independent variables $x_1, x_2, \ldots, x_n$,
- The partial derivatives represent how the function changes with respect to each independent variable.

Real-Time Examples of Partial Differential Equations

1. Heat Equation (Thermal Diffusion)

The heat equation models the distribution of heat (or temperature) in a given region over time. It is one of the most widely studied PDEs and is used in engineering, physics, and other fields.
The heat equation in one spatial dimension is given by:

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}$$

Partial Differential Equation (3)

Where:

- $u(x, t)$ is the temperature at position x and time t,
- α is the thermal diffusivity constant,
- $\frac{\partial^2 u}{\partial x^2}$ is the second partial derivative of temperature with respect to space, representing the rate at which heat is conducted.

This equation is used to model how heat flows through a solid object or how the temperature evolves in a medium over time.

2. Wave Equation (Vibrations of a String)

The wave equation describes the propagation of waves, such as sound waves, light waves, or vibrations in a string. It is a second-order PDE that can describe phenomena like musical instrument vibrations or seismic waves.

Partial Differential Equation (4)

The wave equation in one spatial dimension is given by:

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2}$$

Where:

- $u(x, t)$ represents the displacement of the wave at position x and time t,
- c is the wave speed (such as the speed of sound or the speed of a vibrating string),
- $\frac{\partial^2 u}{\partial t^2}$ is the second partial derivative of displacement with respect to time,
- $\frac{\partial^2 u}{\partial x^2}$ is the second partial derivative of displacement with respect to space, describing how the wave propagates.

This equation is used to model the motion of a vibrating string, sound waves in air, or other wave-like phenomena.

3. Laplace's Equation (Electrostatics and Gravitational Fields)

Laplace's equation is a second-order PDE that arises in many areas of physics, particularly in electrostatics, gravitational fields, and fluid dynamics. It is used to describe the potential field in regions where there are no charges or masses (i.e., in free space).

The general form of Laplace's equation in two dimensions is:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$$

Where:

- $u(x, y)$ represents the potential at a point (x, y),

Partial Differential Equation (6)

- $\frac{\partial^2 u}{\partial x^2}$ and $\frac{\partial^2 u}{\partial y^2}$ are the second partial derivatives of the potential with respect to the spatial variables x and y, respectively.

Laplace's equation is widely used in electrostatics, for example, to calculate the electric potential in a region without charges, or in gravitational problems to describe the gravitational potential in free space.

Table of Contents

Analog Computers (1)

- Historically, continuous system simulation was in general use for studying complex systems long before discrete system simulation was similarly applied.
- The main reason was that, before the general availability of digital computers, needed for the numerical computation of discrete system simulation, there existed devices whose behavior is equivalent to a mathematical operation such as addition or integration. Putting together combinations of such devices in a manner specified by a mathematical model of a system allowed the system to be simulated. By their nature, the devices gave continuous outputs and so lent themselves readily to the simulation of continuous systems. Specific devices have been created for particular systems, but with so general a technique, it has been customary to refer to them as analog computers, or, when they are primarily used to solve differential equation models, as differential analyzers.

Analog Computers (2)

- Physical models based on analogies were described in Chap. I. The specific example discussed there described the analogy between electrical and mechanical systems. Simulation with an analog computer, however, is more properly described as being based on a mathematical model than as being a physical model.
- The most widely used form of analog computer is the electronic analog computer, based on the use of high gain dc (direct current) amplifiers, called operational amplifiers. Voltages in the computer are equated to mathematical variables, and the operational amplifiers can add and integrate the voltages. With appropriate circuits, an amplifier can be made to add several input voltages, each representing a variable of the model, to produce a voltage representing the sum of the input variables.

Analog Computers (3)

- Different scale factors can be used on the inputs to represent coefficients of the model equations. Such amplifiers are called summers. Another circuit arrangement produces an integrator for which the output is the integral with respect to time of a single input voltage or the sum of several input voltages.
- All voltages can be positive or negative to correspond to the sign of the variable represented. To satisfy the equations of the model, it is sometimes necessary to use a sign inverter, which is an amplifier designed to cause the output to reverse the sign of the input.
- Electronic analog computers are limited in accuracy for several reasons.
    - It is difficult to carry the accuracy of measuring a voltage beyond a certain point.

Analog Computers (4)

- Secondly, a number of assumptions are made in deriving the relationships for operational amplifiers, none of which is strictly true: so amplifiers do not solve the mathematical model with complete accuracy. A particularly troublesome assumption is that there should be zero output for zero input.
- Another type of difficulty is presented by the fact that the operational amplifiers have a limited dynamic range of output, so that scale factors must be introduced to keep within the range. As a consequence, it is difficult to maintain an accuracy better than 0.1% in an electronic analog computer.
- A digital computer is not subject to the same type of inaccuracies. Virtually any degree of accuracy can be programmed and, with the use of floating-point representation of numbers, an extremely wide range of variations can be tolerated.

Analog Computers (5)

- Integration of variables is not a natural capability of a digital computer, as it is in an analog computer, so that integration must be carried out by numerical approximations. However, methods have been developed which can maintain a very high degree of accuracy. A digital computer also has the advantage of being easily used for many different problems. An analog computer must usually be dedicated to one application at a time, although time-sharing sections of an analog computer has become possible.

Analog Computers (6)

- In spite of the widespread availability of digital computers, many users prefer to use analog computers. There are several considerations involved. The analog representation of a system is often more natural in the sense that it directly reflects the structure of the system, simplifying both the setting-up of a simulation and the interpretation of the results. Under certain circumstances, an analog computer is faster than a digital computer, principally because it can be solving many equations in a truly simultaneous manner; whereas a digital computer can be working only on one equation at a time, giving the appearance of simultaneity by interlacing the equations. On the other hand, the possible disadvantages of analog computers, such as limited accuracy and the need to dedicate the computer to one problem, may not be significant.

Components of Analog Computers (1)

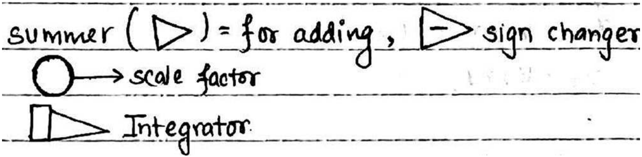
Analog computers consist of various components that perform specific mathematical operations. The main components include:

1. Integrator: An integrator is a component that computes the integral of an input signal with respect to time. It typically uses a feedback loop to continuously adjust the output based on the integral of the input. The output voltage is proportional to the time integral of the input voltage. This component is essential for solving differential equations, as it naturally integrates variables.
2. Summer: A summer is a circuit that adds several input voltages together to produce a single output voltage, which is the sum of the inputs. In an analog computer, the input voltages represent different variables of the model, and the summer provides the sum of these variables. Summers are typically implemented using operational amplifiers in an inverting or non-inverting configuration, and they can handle both positive and negative voltages.

Components of Analog Computers (2)

3. Scale Factor: The scale factor is a multiplier used to adjust the magnitude of the input voltages to match the coefficients of the mathematical model. For example, if a model requires a specific coefficient for a variable, the input voltage representing that variable is multiplied by the appropriate scale factor. This allows the analog computer to accurately represent the coefficients in the mathematical model. The scale factor can be adjusted to scale the inputs to desired magnitudes for the equation being solved.
4. Inverter: An inverter is an amplifier that reverses the polarity of the input voltage. It is used when the mathematical model requires a variable with the opposite sign. The inverter ensures that the output voltage has the correct sign to match the model's equation. It is typically implemented as a simple operational amplifier in an inverting configuration. The inverter is crucial for solving systems where certain terms have negative coefficients or when a variable's sign must be changed.

Components of Analog Computers (3)



$$M\frac{d^2x}{dt^2} + D\frac{dx}{dt} + Kx = F(t) \tag{1}$$

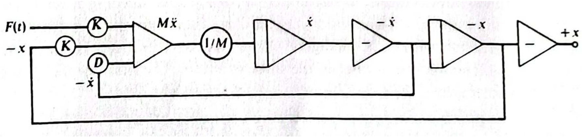
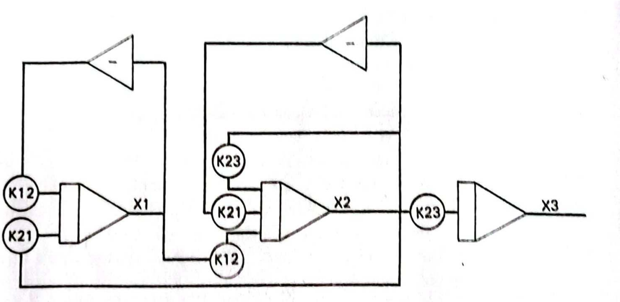


Figure 1

Analog Methods (2)

$$\begin{aligned}\frac{dx_1}{dt} &= -k_{12}x_1 + k_{21}x_2,\\ \frac{dx_3}{dt} &= k_{23}x_2,\\ \frac{dx_2}{dt} &= k_{12}x_1 - (k_{21} + k_{23})x_2.\end{aligned} \tag{2}$$



Analog Methods (3)

Figure 2

Table of Contents

Hybrid Computers (1)

- Initially, analog computers were limited in terms of nonlinear operations, only using a few specialized components such as multipliers and function generators. These devices were costly to produce and limited in their scope.
- However, advances in solid-logic technology have made these nonlinear devices cheaper, more reliable, and more readily available. This progress has also improved the performance of operational amplifiers and expanded the range of devices used in analog computing.
- A hybrid computer is a system that combines the continuous output capabilities of traditional analog computers with the nonlinear, discrete operations of digital computers.
- Digital elements in a hybrid computer are responsible for logical operations (Boolean algebra), storing values for later use, comparing values, and controlling switches.

Hybrid Computers (2)

- This combination of analog and digital elements allows for a more flexible system capable of handling both continuous and discrete elements.
- Originally, the term "hybrid computer" referred to an extension of analog computers by adding special-purpose, often custom-built digital devices. However, in modern computing, very few purely analog computers are built.
- Instead, hybrid computers equipped with numerous standard digital elements are commonly used. These devices can simulate systems that are predominantly continuous but also involve digital elements, such as artificial satellites, where both continuous equations (like motion) and digital control signals must be simulated.
- Hybrid computers are particularly valuable when simulating systems that require both continuous and discrete components.

Hybrid Computers (3)

- For example, in the case of an artificial satellite simulation, the continuous equations describing the motion of the satellite can be handled by the analog part of the hybrid computer, while the digital part can simulate the control signals that adjust the satellite's orientation or behavior. This combination enables the simulation of more complex systems with both continuous and digital components.
- Furthermore, hybrid computers are highly effective when a system, which could be adequately modeled using an analog computer, is the subject of a repetitive study.
- For instance, if the goal is to search for a maximum value in a system, the digital elements of the hybrid computer can store values, modify initial conditions, and manage switches to control multiple runs of the simulation. By doing so, hybrid computers can carry out large portions of a study without requiring constant human intervention, making them efficient tools for such tasks.

Table of Contents

Digital Analog Simulators (1)

- To avoid the disadvantages of analog computers, many digital computer programming languages have been developed to produce digital-analog simulators. These simulators allow a continuous model to be programmed on a digital computer in essentially the same way as it is solved on an analog computer.
- The primary advantage of digital-analog simulators is that they combine the benefits of digital computers (such as higher accuracy, flexibility, and general-purpose capabilities) with the continuous nature of analog simulations. This allows for greater precision and more complex models to be handled, while still simulating the behavior of continuous systems.
- The programming languages used for these simulators contain *macro-instructions* that perform the essential actions needed for the simulation, such as *adders, integrators,* and *sign changers*. These macro-instructions replicate the functions of operational amplifiers in analog computers.

Digital Analog Simulators (2)

- For example, adders are used to sum input values representing different system variables, integrators calculate the accumulation of these variables over time, and sign changers modify the sign of input signals to satisfy the mathematical model of the system.
- In a digital-analog simulator, a program is written to link together these macro-instructions in a manner similar to how operational amplifiers are interconnected in analog computers. This structure allows digital computers to mimic the behavior of an analog system with greater control over parameters such as precision, speed, and flexibility.
- However, the use of such simulators still requires the programmer to have a solid understanding of both the mathematical model and the computational resources available.

Digital Analog Simulators (3)

- Over time, more advanced and powerful techniques have been developed for applying digital computers to the simulation of continuous systems. These methods allow for the handling of large, complex systems with high accuracy and greater computational efficiency.
- The development of digital-analog simulators has paved the way for more sophisticated simulation tools that integrate digital processing power with continuous modeling. As a result, digital-analog simulators are now less commonly used in practice, being largely replaced by more integrated and specialized approaches.
- Despite this shift, digital-analog simulators were an important step in the evolution of computer-based simulations, enabling the application of digital computing to fields traditionally dominated by analog methods.

Digital Analog Simulators (4)

A detailed history of their development can be found in Brennan, Robert D., and Robert N. Linebarger. "A survey of digital simulation: digital analog simulator programs." Simulation 3.6 (1964): 22-36.

Table of Contents

Continuous System Simulation Language (1)

- Confining a digital computer to routines that represent just the functions of an analog or hybrid computer, as done in digital-analog simulators, is clearly a restriction in terms of flexibility and functionality.
- To remove this limitation, a number of Continuous System Simulation Languages (abbreviated as CSSLs) have been developed. These languages allow for the direct programming of continuous systems, providing a more natural and flexible approach compared to the constrained structure of digital-analog simulators.
- Unlike digital-analog simulators, which require the user to break down equations into functional elements (such as adders and integrators), a CSSL uses a familiar statement-based input for digital computers.

Continuous System Simulation Language (2)

- This means that a problem can be programmed directly from the equations of a mathematical model, making it easier to describe the system without the need for extensive translation into specific operations. The resulting approach is more intuitive and accessible for modeling complex systems.
- Of course, a CSSL can still include macros or subroutines that replicate the functions of specific analog elements. This makes it possible to incorporate the convenience of a digital-analog simulator if desired, providing an additional layer of flexibility.
- In fact, most CSSL implementations not only provide a subset of standard analog elements (such as addition and integration) but also allow users to define custom, special-purpose elements.
- These elements correspond to operations particularly important in specific types of applications, enabling tailored solutions to a broad range of continuous system simulation problems.

Continuous System Simulation Language (3)

- One of the key strengths of CSSLs is their ability to describe complex relations between variables. Beyond just providing simple analog functions, CSSLs include a variety of built-in mechanisms for expressing mathematical relationships in the form of differential equations, algebraic expressions, and logical operations. This capability extends the power of CSSLs beyond the limitations of analog methods, allowing for the modeling of dynamic systems that involve nonlinearities, time-varying coefficients, and other complex behaviors that are essential in advanced simulations.

Continuous System Simulation Language (4)

- The core of a CSSL is its ability to represent continuous systems using mathematical expressions, making it possible to simulate systems described by linear and nonlinear differential equations, which are typical in fields such as engineering, physics, and economics. These languages thus allow for a more direct translation from theoretical models to practical simulation programs, facilitating faster model development and iteration.
- A general specification of the requirements for a CSSL has been published by a user group, detailing the essential components and capabilities that a CSSL should provide to users.

Continuous System Simulation Language (5)

- Over the years, several implementations of CSSLs have been produced, each designed to meet the specific needs of different industries and applications. A particularly notable implementation is the Continuous System Modeling Program, Version III (CSMI-III), which serves as a powerful tool for continuous system simulation across various disciplines.

Table of Contents

- CSMP III, or Continuous System Modeling Program III, is a software tool developed for the numerical simulation and modeling of continuous systems.
- Primarily used in the 1970s and 1980s, it allows users to define and solve differential equations to simulate real-world dynamic systems.
- The software employs FORTRAN-like statements for structural definitions, data assignments, and control operations
- CSMI-III extends beyond the traditional role of CSSLs by providing a platform that allows users to model and simulate a wide range of physical and engineering systems.
- It provides a versatile environment where mathematical models of systems described by differential equations can be directly translated into computer simulations.

CSMP III (2)

- This is achieved through a user-friendly interface that supports the input of equations and system specifications in a manner that closely resembles the structure of the original mathematical model.

Features of CSMI-III

- Direct Input of Mathematical Models: Users can directly input mathematical equations that describe the behavior of the system being modeled, eliminating the need for breaking them down into simpler functional operations like those required in digital-analog simulators.
- Specialized Element Definitions: While the program provides a core set of analog functions such as adders and integrators, CSMI-III also allows users to define custom elements suited to their specific application, making it a highly adaptable tool for a wide variety of simulations.

CSMP III (3)

- Support for Complex Systems: CSMI-III can handle not just linear systems but also nonlinear systems with time-varying coefficients, offering greater power in modeling real-world phenomena that involve such complexities.
- Integration of Multiple Domains: CSMI-III facilitates the simulation of systems that involve both continuous dynamics and discrete components, making it suitable for hybrid systems where the interaction between analog and digital elements is critical.
- It is also used in research settings where the primary goal is to optimize system behavior or perform sensitivity analysis across a range of conditions.

CSMP III consist of:

CSMP III (4)

1. Structural Statements: Define model structure with programming language syntax and functional blocks for repeated operations.
2. Data Statements: Assign numerical values to parameters, constants, and initial conditions.
3. Control Statements: Specify options for program assembly, execution, and output of calculation results.

Structure Statements (1)

Structural statements represent the system's dynamics using mathematical operations like addition, subtraction, multiplication, division, and exponentiation. For example, the equation:

$$X = 6\frac{Y}{W} + X + (Z - 2)^2$$

would be represented in CSMP III as:

$$X = 6.0 * Y/W + (Z - 2.0) * *2$$

Variables and Constants

- Variable names can be up to six characters.
- Constants are defined using data statements, with real constants in decimal notation.
- Real constants are specified in decimal notation.

- Exponent notation may also be used; for example, 1.2E-4 represents 0.00012.
- Fixed value constants may also be declared.

Functional Blocks

CSMP III includes functional blocks that perform operations similar to FORTRAN's mathematical functions. These include exponential functions, trigonometric functions, and others such as maximum or minimum values. Example of an integration function block:

$$Y = \mathrm{INTGRL}(IC, X)$$

This calculates the integral of X with respect to time and assigns it to the variable Y, starting from an initial value IC.

Data Statements (1)

- In CSMP III, data statements are used to initialize variables and define constants within the model. These data statements provide the initial values for simulation variables and define other parameters used throughout the model.
- Of the data statements, one called INCON can be used to set the initial values of the integration-function block. Other parameters can be given a value for a specific run with the CONST control statement.

There are several types of data statements in CSMP III:

Data Statements (2)

- INCON (Initial Conditions): This statement sets the initial values for variables. It is commonly used for specifying the starting values of integration variables.

  $$\text{INCON } Y = 0.0, \quad X = 1.0$$

  In this example, the initial values for variables Y and X are set to 0.0 and 1.0, respectively.

- CONST (Constants): This statement is used to define fixed values that do not change during the simulation. Constants are usually parameters or other scalar values.

  $$\text{CONST } A = 2.0, B = 5.0$$

Data Statements (3)

- PARAM (Parameter values): The `PARAM` statement defines a set of values for a specific parameter. When using this statement, the model is run for each of the specified values in succession. This is similar to the list in Python.

$$\text{PARAM } D = (0.25, 0.50, 0.75, 1.0)$$

This statement defines a series of values for the parameter D and runs the simulation four times, with D taking each of the values in turn.

Control Statements

Control statements specify how the model behaves, including defining initial values, parameter values, time intervals, and how results are printed or plotted.

- INCON: Sets initial values.
- CONST: Assigns constant values for the simulation.
- PARAM: Defines a series of values for parameters to vary during the simulation.
- TIMER: Specifies time intervals and simulation times.
- DELT: Integration Interval
- FINTIM: Finish Time
- PRDEL: Interval at which to print results
- OUTDEL: Interval at which to print-plot
- PRINT and PRTPLT: Used for generating printed output and graphical plots.

Two other control statements with the words TITLE and LABEL can be used to put headings on the printed and print-plotted outputs, respectively. Whatever comment is written after the words becomes the heading.

Hybrid Statements

The set of structural, data, and control statements for a problem can be assert] bled in any order but they must end with an END control statement. However control statements which define another run of the same model can follow an ENE statement if they also are terminated by another END statement. This can hs repeated many times, until an ENDJOB statement signals end of all runs. A completely separate model can then follow the ENDJOB statement.

CSMP III Program (1)

$$M\frac{d^2x}{dt^2} + D\frac{dx}{dt} + Kx = F(t)$$

```
TITLE AUTOMOBIL SUSPENSION SYSTEM
*
PARAM D = (5.656, 16.968, 39.592, 56.55, 113.12)
*
X2DOT = (1.0/M)*(K*F -K*X - D*XOT)
XDOT = INTGRL(0.0, X2DOT)
X = INTOPL(0.0, XDT)
*
CONST M = 2.0, F = 1.0, K = 400.0
TIMER DELT = 0.005, FINTIM = 1.5, PRDEL = 0.05, OUTDEL =
     0.05
PRINT X, XDOT, X2DOT
LABEL DISPLACEMENT VERSUS TIME
END
STOP
```

CSMP III Program (2)

$$L\frac{d^2q}{dt^2} + R\frac{dq}{dt} + \frac{q}{C} = E(t) \quad (3)$$

$$\int_0^4 \sqrt{16 - x^2}\, dx$$

Simulation of Predator-Prey Model

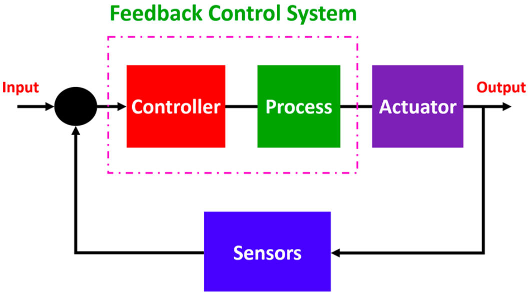
$$\frac{dx}{dt} = \alpha x - \beta xy$$

$$\frac{dy}{dt} = \delta xy - \gamma y$$

Table of Contents

Feedback System (1)

- A key factor influencing the performance of many systems is the coupling between the input and output. This phenomenon is referred to as feedback.
- A feedback system is a closed-loop system where the output influences the input, allowing the system to self-regulate, either amplifying changes (positive feedback) or diminishing them (negative feedback).
- These mechanisms are essential for system stability and responsiveness.
- It is used in various fields such as control systems, biology, economics, and engineering
- In feedback system, the system's output is fed back into the system as input to regulate its further output. Feedback can be positive (amplifying changes) or negative (reducing changes).



1. Positive Feedback
   - Mechanism: The feedback signal is added to the input signal, amplifying the output. This results in a runaway effect or exponential growth, which can lead to system instability if not controlled.

Feedback System (3)

- Use Case: It is less commonly used because it can lead to uncontrollable situations. However, it is used in processes where amplification is desired, such as in the human birth process (contractions leading to further contractions) and electronic oscillators.
- Characteristics:
  - Amplifies output.
  - Can lead to instability.
  - Rarely used in control systems due to the risk of uncontrollable outcomes.

2. Negative Feedback

- Mechanism: In negative feedback, the feedback signal is subtracted from the input, which helps the system return to equilibrium. It stabilizes the system by reducing errors or deviations from the desired output.

Feedback System (4)

- Use Case: Negative feedback is the most common type, especially in control systems, such as in temperature regulation (like in thermostats), **audio systems**, and automated industrial processes.
- Characteristics:
  - Reduces errors or deviations.
  - Enhances stability and accuracy.
  - Common in most control systems for maintaining performance.

Feedback System (5)

Components of a Feedback System

1. **Sensor** The sensor measures the actual output of the system. It can take various forms, such as temperature sensors, pressure sensors, or position sensors, depending on the application. Example: In a thermostat, a temperature sensor detects the room temperature.

2. **Comparator** The comparator compares the measured output (from the sensor) to a predetermined setpoint (desired value). This difference is often called the *error signal*.

Example: In a temperature control system, the comparator compares the room temperature to the desired temperature setting.

Feedback System (6)

3. Controller The controller processes the error signal and determines the necessary corrective action to bring the output closer to the setpoint. It adjusts system parameters to minimize the error.
Example: In a heating system, the controller decides whether to turn the heater on or off based on the temperature difference.

4. Actuator The actuator is the device that performs the actual adjustments to the system based on the controller's commands. It translates the controller's decisions into physical actions.
Example: In a heating system, the actuator could be a valve or a heating element that adjusts the temperature.

Simulation of an Autopilot (1)

Recommendation

For detail theory, please see Section 4-11, page 74 of our Textbook.

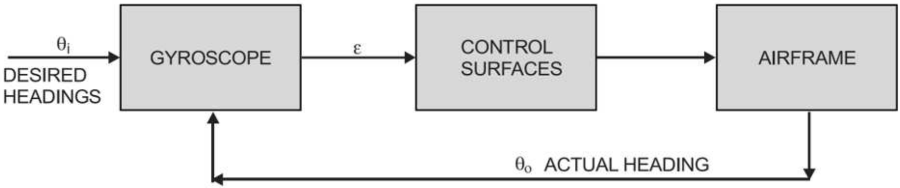


Fig. 0.2: *Model of an autopilot aircraft.*

The error signal, $\epsilon =$ desired heading, or input,θ_i - actual heading, or output, θ_0

$$\epsilon = \theta_i - \theta_0 \tag{1}$$

The rudder is assumed to generate a force (torque) proportional to this error signal. The rudder's turning force is proportional to the magnitude of the error signal.

$$\text{Force} \propto \epsilon$$

Instead of moving the aircraft sideways, this force applies a torque to rotate the aircraft.

$$\text{torque} \propto \epsilon$$

The torque acting on the aircraft is due to two factors:

Simulation of an Autopilot (3)

- The torque produced by the rudder itself, which depends on the error signal.
- A resisting force (viscous drag) that is proportional to the angular velocity of the aircraft.

$$\text{torque} = K\epsilon - D\dot{\theta}_0 \tag{2}$$

Where K and D are constants. The first term on the right-hand side is the torque produced by the rudder, and the second is the viscous drag.

According to the fundamental law of mechanics, the acceleration of a body is proportional to the applied force. For rotational motion, this law translates into:

$$\text{torque} = I\ddot{\theta}_0 \tag{3}$$

where:

Simulation of an Autopilot (4)

- I is the moment of inertia of the aircraft.
- $\ddot{\theta}_0$ is the angular acceleration (second derivative of the heading).

Combining Eq. (1), (2), and (3),

$$I\dot{\theta}_0 + D\dot{\theta}_0 + K\theta_0 = K\theta_i\dot{\theta}_0 \tag{4}$$

If we divide both sides of the equation by I, and making the following substitutions:

$$2\zeta\omega = \frac{D}{T}, \quad \omega^2 = \frac{K}{T}$$

The equation of motion relating output to input then takes the following form:

$$\ddot{\theta}_0 + 2\zeta\omega\dot{\theta} + \omega^2\theta_0 = \omega^2\theta_i \tag{5}$$

Simulation of an Autopilot (5)

```
TITLE AIRCRAFT WITH RATE CONTROL
*
PARAM D = (5.656, 16.968, 39.592, 56.56, 113.12)
INPUT = A*TIME
ERROR = INPUT - HEAD
TORQUE = K*ERROR - D*ANGVEL
ANGACC = TORQUE/I
HEAD = INTGRL(0.0, ANGVEL)
ANGVEL = INTGRL(0.0, ANGACC)
*
CONST I = 2.0, K = 400.0, A = 0.0175
TIMER DELT = 0.005, FINTIM = 1.5, PRDEL = 0.05
PRINT HEAD
LABEL HEADING VERSUS TIME
END
```

Table of Contents

Interactive System (1)

- An interactive system is a computing system that allows direct communication between the user and the system in real-time.
- It enables users to input commands, data, or gestures and receive immediate feedback, fostering a dynamic and responsive user experience.
- Such systems are designed to support a variety of interactions, such as graphical user interfaces (GUIs), voice recognition, and touch interfaces.
- Common examples of interactive systems include video games, multimedia applications, and computer-aided design (CAD) tools.
- These systems are characterized by their ability to process user input and adapt to user needs, enhancing usability and functionality.
- Interactivity is essential in modern computing as it promotes user engagement and increases efficiency by enabling continuous user-system communication.

Interactive System (2)

- In addition, interactive systems often incorporate real-time data processing, ensuring that responses and outputs are generated without noticeable delays.

Table of Contents

Real Time Simulation (1)

- Real-time simulation involves modeling and simulating a system where the computational time is synchronized with the actual time, ensuring that the simulation runs in parallel with real-world processes.
- The key characteristic of real-time simulations is that the results and feedback are produced without significant delays, enabling immediate responses.
- These simulations are crucial in areas like control systems, robotics, automotive engineering, and flight simulations, where actions need to be executed instantaneously.
- Real-time simulation ensures that the virtual model behaves like its real-world counterpart, allowing for testing, training, and validation of systems without physical prototypes.

Real Time Simulation (2)

- The computational environment is carefully managed to meet the strict timing constraints, as even small delays can lead to incorrect results or unsafe system behaviors.
- By maintaining this real-time synchronization, it becomes possible to interact with, test, and optimize systems in dynamic, time-sensitive scenarios.

Table of Contents

Predator-Prey Model (1)

The Lotka–Volterra equations are a pair of first-order nonlinear differential equations frequently used to describe the dynamics of biological systems, such as predator-prey or competition models. The general form of the Lotka–Volterra equations is given by:

$$\frac{dx}{dt} = \alpha x - \beta xy \tag{6}$$

$$\frac{dy}{dt} = \delta xy - \gamma y \tag{7}$$

where:

- x represents the population density of prey (for example, the number of rabbits per square kilometre),

Predator-Prey Model (2)

- y represents the population density of predators(for example, the number of foxes per square kilometre),
- α is the growth rate of prey,
- β is the predation rate,
- γ is the death rate of predators,
- δ is the rate at which predators increase by consuming prey.

The Lotka–Volterra predator-prey model makes a number of assumptions about the environment and biology of the predator and prey populations:

1. The prey population finds ample food at all times.
2. The food supply of the predator population depends entirely on the size of the prey population.
3. The rate of change of population is proportional to its size.

Predator-Prey Model (3)

4. During the process, the environment does not change in favour of one species, and genetic adaptation is inconsequential.
5. Predators have limitless appetite.
6. Both populations can be described by a single variable. This amounts to assuming that the populations do not have a spatial or age distribution that contributes to the dynamics.

Biological Interpretation and Model Assumptions

Prey Growth:

- Prey have unlimited food supply and reproduce exponentially, represented by the term αx.
- Prey population growth is affected by predation, represented by βxy, where the predation rate depends on the interaction between predators and prey.

Predator-Prey Model (4)

- If either prey (x) or predator (y) is zero, no predation occurs.
- The prey equation is:

$$\frac{dx}{dt} = \alpha x - \beta xy$$

Predator Growth:

- Predator population growth is represented by the term δxy, indicating that it increases with the consumption of prey.
- Predator growth rate is different from predation rate, so a separate constant δ is used.
- The term γy represents the loss of predators due to natural death or emigration.
- The predator equation is:

$$\frac{dy}{dt} = \delta xy - \gamma y$$

Predator-Prey Model (5)

Suppose there are two species of animals, a rabbit (prey) and a fox (predator). If the initial densities are 10 rabbits and 10 foxes per square kilometre, one can plot the progression of the two species over time; given the parameters that the growth and death rates of rabbits are 1.1 and 0.4 while that of foxes are 0.1 and 0.4 respectively. The choice of time interval is arbitrary.

The dynamics of the rabbit population ($x(t)$) and the fox population ($y(t)$) can be described by the following system of differential equations:

$$\frac{dx}{dt} = \alpha x - \beta xy$$
$$\frac{dy}{dt} = \delta xy - \gamma y$$

Where:

- $x(t)$ is the rabbit population density at time t,

Predator-Prey Model (6)

- $y(t)$ is the fox population density at time t,
- $\alpha = 1.1$ is the natural growth rate of the rabbits,
- $\beta = 0.4$ is the predation rate of foxes on rabbits,
- $\delta = 0.1$ is the rate at which foxes reproduce from consuming rabbits,
- $\gamma = 0.4$ is the natural death rate of the foxes.

The initial conditions:$x(0) = 10$(rabbits per square kilometre), $y(0) = 10$(foxes per square kilometre)

Finally,

$$\frac{dx}{dt} = 1.1x - 0.4xy, x(0) = 10$$

$$\frac{dy}{dt} = 0.1xy - 0.4y, y(0) = 10$$

Predator-Prey Model (7)

Solution using Runge-Kutta 4th-order

At each time step t_n, we calculate the values of k_1, k_2, k_3, k_4 for both x and y. The procedure for updating x_n and y_n is as follows:

For x:

$$
\begin{aligned}
k_{1x} &= h \cdot f_1(x_n, y_n) \\
k_{2x} &= h \cdot f_1\left(x_n + \frac{k_{1x}}{2}, y_n + \frac{k_{1y}}{2}\right) \\
k_{3x} &= h \cdot f_1\left(x_n + \frac{k_{2x}}{2}, y_n + \frac{k_{2y}}{2}\right) \\
k_{4x} &= h \cdot f_1(x_n + k_{3x}, y_n + k_{3y})
\end{aligned}
$$

For y:

$$
\begin{aligned}
k_{1y} &= h \cdot f_2(x_n, y_n) \\
k_{2y} &= h \cdot f_2\left(x_n + \frac{k_{1x}}{2}, y_n + \frac{k_{1y}}{2}\right) \\
k_{3y} &= h \cdot f_2\left(x_n + \frac{k_{2x}}{2}, y_n + \frac{k_{2y}}{2}\right) \\
k_{4y} &= h \cdot f_2(x_n + k_{3x}, y_n + k_{3y})
\end{aligned}
$$

Finally, we update the values of x_{n+1} and y_{n+1} as follows:

$$
\begin{aligned}
x_{n+1} &= x_n + \frac{1}{6}\left(k_{1x} + 2k_{2x} + 2k_{3x} + k_{4x}\right) \\
y_{n+1} &= y_n + \frac{1}{6}\left(k_{1y} + 2k_{2y} + 2k_{3y} + k_{4y}\right)
\end{aligned}
$$

Predator-Prey Model (8)

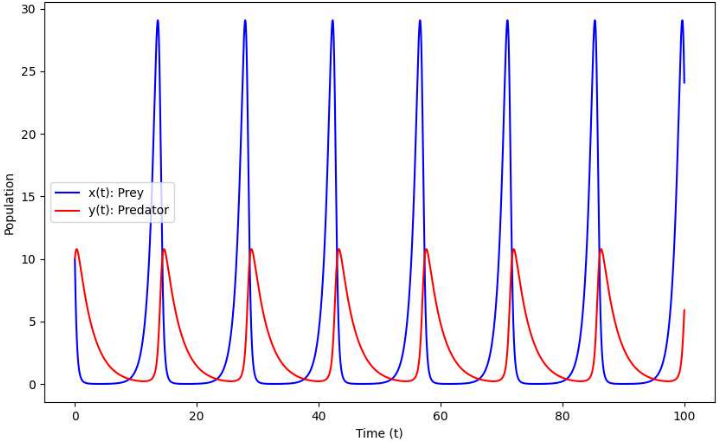


Figure 3: Population dynamics for rabbit and fox problem using Runge Kutta