

Computational Models for Embedded Systems

Vescan Andreea, PHD, Assoc. Prof.

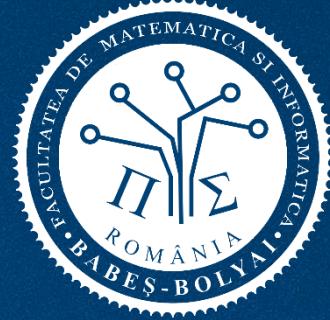


Faculty of Mathematics and Computer Science
Babeş-Bolyai University

Cluj-Napoca
2025-2026



Lecture 13b: Dynamical systems



Software Systems Verification and Validation

“Tell me and I forget, teach me and I may remember, involve me and I learn.”

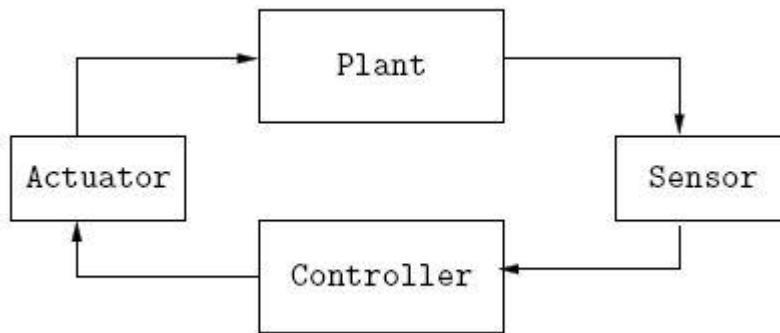
(Benjamin Franklin)

Outline

- Dynamical Systems
- Continuous-time Models
 - Continuously evolving inputs/outputs
 - Models with: State variables;
Disturbance
 - Linear Systems
 - Composing systems
 - Stability
- Designing Controllers
 - Open-loop Controller
 - Feedback Controller
- Analysis Techniques
 - Numerical Simulation
 - Computing reachable states

Continuous-time Models

Continuously Evolving Inputs and Outputs



Typical architecture of
a control system

- plant → models the physical world that is to be controlled
- controller → influences the evolution of the plant using actuators
- sensor → provides measurements that help the controller to make decisions
 - Example:
 - thermostat design:
 - plant=room, sensor=thermometer, controller=regulate the temperature (adjusting the heat flow from the furnace)
 - → plant model needs to capture how the temperature of the room changes as a function of the heat-flow from the furnace and the difference between the room temperature and outside temperature.

Continuous-time Models (cont.)

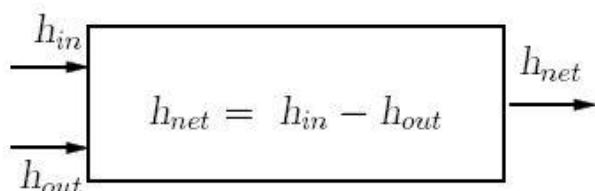
Continuously Evolving Inputs and Outputs

- **Models of dynamical systems** - described as components with inputs and outputs that are connected to one another using block diagrams.
 - model of computation is synchronous
 - essential difference:
 - synchronous model
 - the values of variables - are updated in a sequence of discrete logical rounds
 - dynamical model
 - the values of variables are updated continuously with passage of time = **continuous-time components.**
 - **continuous-time component** - the variables range over a compact set (e.g. an interval of the set of **reals** with specified units of measurement)
 - **models of dynamical systems**
 - every variable implicitly has the type real (e.g. an explicitly associated interval range)
 - The values a variable takes over time can then be described as a function from the time domain (**time**) to real. Functions from the time domain to the set of reals are called **signals**.

For a variable x , a signal over x is a function that assigns a real value $\rho(t)$ to the variable x for every time t in time.

Continuous-time Models (cont.)

Continuously Evolving Inputs and Outputs



- Continuous-time component **NetHeat**

- two input variables: h_{in} (heat inflow) and h_{out} (heat outflow)
- a single output variable h_{net} (net heatflow)
- mapping from two input signals to an output signal
- dynamics is expressed by the equation $h_{net} = h_{in} - h_{out}$
(at every time t , the value of h_{net} equals the expression)
- Given a signal ρ_{in} for the input h_{in} , and a signal ρ_{out} for the input h_{out} , the output signal ρ_{net} for the output variable h_{net} is defined by $\rho_{net}(t) = \rho_{in}(t) - \rho_{out}(t)$ ρ_{net} for all times t in time.
- This unique output signal is called the response of the component to input signals ρ_{in} and .

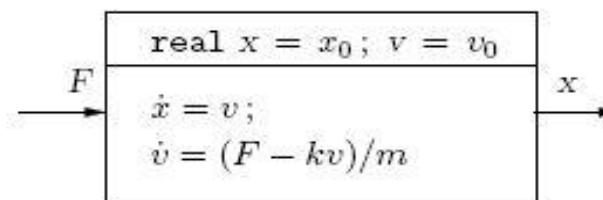
Continuous-time Models

Models with: State variables; Disturbance

- Example of a stateful continuous-time component – how the speed of a car changes as a result of the force applied to it by the engine

$$F - k \dot{x} = m \ddot{x}.$$

- F =the force applied to the car
- x = the position of the car
- k is the coefficient of the frictional force
- m denotes the mass of the car
- \dot{x} denotes the first-order time derivative of the signal assigning values to the variable x , and thus captures the velocity of the car.
- \ddot{x} denotes the second-order derivative of this signal, that is, the acceleration of the car.



- x = models the position of the car
- v = models the velocity of the car
- For every state variable, the dynamics is given by specifying the first-order time-derivative of the value of the state variable as a function of the input variables and state variables
 - $\dot{x} = v$ the rate of change of the state variable x at time t equals the value of the state variable v at time t
 - the rate of change of the state variable v at time t equals the value of the expression $(F - kv)/m$ at time t
- The two equations together are equivalent to the original equation.
- The output of the car is its position

Continuous-time Models

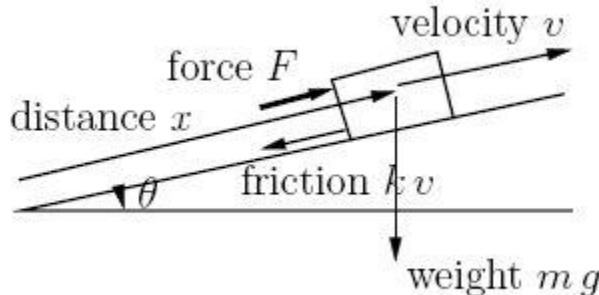
Models with: State variables; Disturbance

- The dynamics of the component is specified by
 - a real-valued expression h_y for every output variable y
 - a real-valued expression f_x for every state variable x
- Each of these expressions is an expression over the **input** and **state** variables.
- The value of the output variable y at time t is obtained by:
 - evaluating the expression h_y using values of state and input variables at time t
 - the signal for a state variable x should be such that its rate of change at time t should equal the value of the expression f_x evaluated using values of state and input variables at time t
- For given input signals, the signals for state and output variables are uniquely determined.
- The **execution of a continuous-time component** is similar to the **execution of a deterministic synchronous reactive component**, except the notion of a round is now infinitesimal: at every time t , the outputs at time t are determined as a function of the inputs at time t and the state of the component at time t , and then the state is updated using the rate of change specified by the derivative evaluated using the inputs and state at time t .

Continuous-time Models

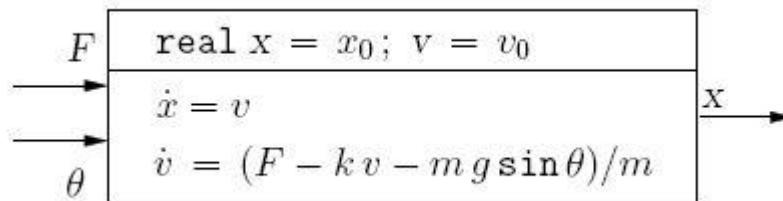
Models with Disturbance

- we now want to account for the grade of the road:
 - on an up-hill, the weight of the car works against the force applied by the engine,
 - on a down-hill, the weight of the car adds to this force
- The cruise controller needs to adjust the force F to keep the net velocity v in the direction along the road constant.
- the grade of the road by an additional input that captures the angle of the road with the horizontal
- The forces acting on the car in the direction along the road are:
 - F in the forward direction controlled by the engine,
 - $k v$ in the backward direction modeling the friction,



Car motion on a graded road

onal force along the road in the backward direction.

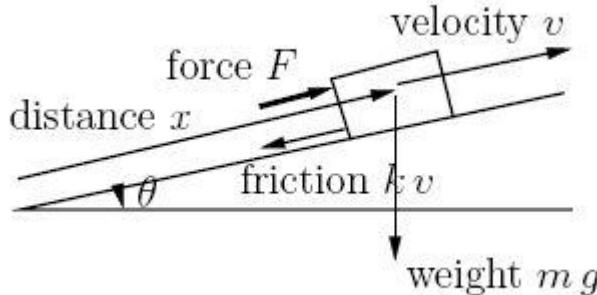


Continuous-time component
modeling car motion on a graded road

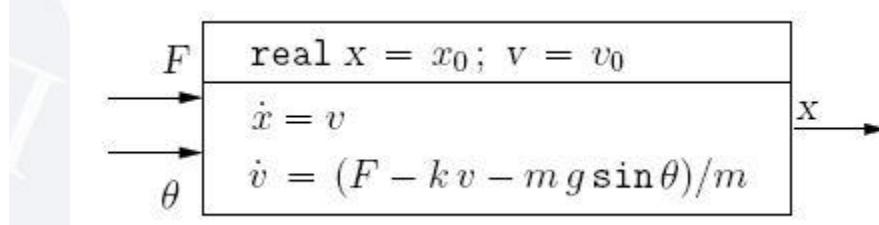
Continuous-time Models

Models with Disturbance

- The control design problem for the revised model is different in a crucial way:
 - the input signal θ modeling the grade of the road is not controlled by the controller and is not known in advance.
 - Such an uncontrolled input is sometimes called a **disturbance**.
 - The controller should be designed to produce the controlled input signal F so that it works for a reasonable range of F values (for instance, all values in the range $[-30, +30]$).



Car motion on a graded road



Continuous-time component
modeling car motion on a graded road

Continuous-time Models

Linear Systems

- Consider a continuous-time component C with an input variable x and an output variable y .
- Given a input signal $\rho : time \rightarrow real$ there is a unique output signal $\rho' : time \rightarrow real$ corresponding to the execution of the component C corresponding to the input signal ρ .
- Thus, a continuous-time component is a function from input signals to output signals. If this function is linear, the corresponding dynamical system is called **linear**.
- Linearity – 2 properties:
 - **Scaling** = If the input signal is scaled by a constant factor, then the output signal also gets scaled by the same factor
 - **Additivity**= If an input signal can be expressed as a sum of two input signals, then the corresponding output signal is also the sum of the output signals corresponding to the component input signals.

Continuous-time Models

Linear Systems

- The conventional form for expressing the dynamics for linear systems uses matrices.

- Consider a linear component with:

- m input variables I={u₁,...,u_m}
- n state variables S = {x₁,...,x_n}
- k output variables O={y₁,...,y_k}

- We can view:

- the input as a vector of dimension m x 1
- state as a vector of dimension n x 1
- output as a vector of dimension k x 1

- The dynamics is expressed by four matrices each with real-valued coefficients:

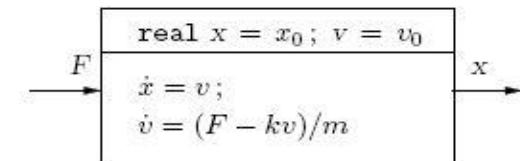
- matrix A =n x n, matrix B=n x m, matrix C=k x n, matrix D= k x m.

- The dynamics is given

$$\dot{S} = AS + BI, \quad O = CS + DI.$$

- for the car motion example: m=2 and n=k=1

$$A = \begin{bmatrix} 0 & 1 \\ 0 & -k/m \end{bmatrix}; \quad B = [0 \ 1/m]; \quad C = [1 \ 0]; \quad D = [0]$$



Continuous-time Models

Composing systems

- Continuous-time components can be composed using block diagrams in a way similar to synchronous components.
- Operations
 - variable renaming
 - output hiding
 - parallel composition
- Need to establish absence of cyclic awaits dependencies – when composing
 - an output variable y of a continuous time component C awaits an input variable x if the value of y at time t depends on the value of x at time t
 - the await dependencies can be determined by a simple syntactic check (because the evolution of each output variable y of a continuous-time component is described by an expression hy over the state and input variables):
 - if the set of input variables that occur in the expression hy is J , then the output y awaits the input variables in J , but does not await the remaining input variables
 - the output variable yj awaits the input variable ui precisely when the matrix entry D_{ji} is non-zero.

Continuous-time Models

Stability

- Control systems – example a cruise controller
 - **a safety requirement** - the speed of the car should always be below some maximum speed
 - **liveness requirement** - the difference between the actual speed and the desired speed of the car should eventually be close to zero
 - **a new kind of requirement** - that changes in inputs, such as the grade of the road, should not cause disproportionately large changes in the speed of the car = stability. (relevant only for continuous-time systems)
- A signal ρ assigning values to a real-valued variable as a function of time, is said to be bounded if there exists a constant Δ such that $|\rho(t)| \leq \Delta$ for all time t .

Continuous-time Models

Stability

- A signal over a set V of variables is bounded if the component of ρ corresponding to each variable x in V is bounded.
- **Stable system** - whenever the input signal is bounded then so is the output signal produced by the component in response. The bound on the output signal can be different from the bound on the input signal. This particular formalization of stability is known as

Bounded Input Bounded Output (BIBO) stability

STABLE SYSTEM

A continuous-time component C with input variables I and output variables O is *Bounded-Input-Bounded-Output stable* if for every bounded input signal ρ , the output signal ρ' produced by C in response to the input ρ , is also bounded.

Designing Controllers

- Given a dynamical system model of the plant
 - the controller is designed to provide the controlled input signals to maintain the output of the system close to the desired output irrespective of changes in uncontrolled input signals corresponding to disturbances.
- Next
 - some basic terminology in control design
 - get familiar with the most commonly used class of controllers in industrial practice
 - Open-loop Controller
 - Feedback Controller

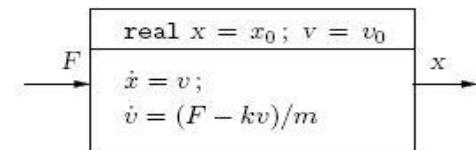
Designing Controllers

Open-loop Controller

- open-loop controller = does not use measurements of the state or outputs of the plant to make its decisions
 - relies on the model of the plant to decide on the controlled input for the plant
 - its implementation does not require sensors
- the controller's objective is to maintain constant velocity = we want
$$v = 0 \quad \text{at all times}$$

if v_0 is the initial velocity, the desired value of the input force F equals kv_0 : the controller can simply apply this constant force to the car to maintain the velocity at v_0

- In practice, operation of such a controller is acceptable, provided there is a possibility of manual intervention. If the driver finds the speed of the car unacceptable, she would simply increment or decrement the desired speed triggering a recalculation of the force applied by the open-loop controller.



Designing Controllers

Feedback Controller

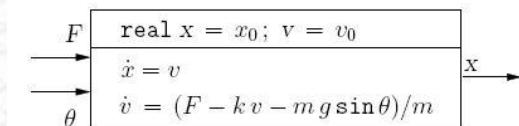
- A feedback controller - uses sensors to measure the output, and thus, indirectly the current state of the plant, to update the values of the controlled input variables.

- The model accounts for the change in the grade of the road. Suppose that the controller is applying the correct amount of force to maintain the velocity of the car at the desired cruising speed.

- The speed of the car, as measured by the actuators, is an input to the controller. It notices the change in speed and adjusts the force to make the velocity again equal to the desired cruising speed.

- Feedback controller not only can cope with disturbances (such as the grade) whose variation with time is not predictable in advance, but can work well even when the mathematical model of the plant is only a rough approximation of the real-world dynamics.

Implementation of a feedback controller requires sensors, and its performance is related to the accuracy of measurements by these sensors.



Analysis Techniques

- Given a model of a continuous-time component – includes:
 - the plant model
 - the feedback controller
 - constraints on initial state and disturbances
- → we want to analyze the behavior of the system

Analysis Techniques

Numerical Simulation

- a continuous-time component
 - state is captured by the variable x which may be a vector of variables
 - input is described by a variable u which also can be a vector of variables.
 - The function f gives the rate of change of the value of x as a function of state x and input u .
 - Given an input signal that assigns values to u as a function of time and an initial state x_0 , the evolution of the state of the system, then, can be computed by solving the differential equation with $x(0) = x_0$.
$$x = f(x, u)$$
- a closed-form solution for the state response $x(t)$ at time t can be computed – for specific form of f
- a general method for computing this signal - is to employ numerical simulation

the user provides a discretization step parameter Δ and the simulator attempts to compute the values of the state at times $\Delta, 2\Delta, 3\Delta, \dots$ that closely approximate the values of the desired response signal $x(t)$ as closely as possible

Analysis Techniques

Computing reachable states

- **safety verification problem** =

- given a dynamical system C with state variable x and dynamics described by $\dot{x} = f(x)$.
 - given a set $Init$ of initial states and a state property φ
 - → we want to know if for every state signal $x(t)$ with $x(0) \in Init$, is it the case that the state $x(t)$ satisfies φ for every t .
-
- That is, if the system is initialized to start in a state satisfying the constraint described by $Init$, we want to check if the state always stays within the region described by φ .
 - The set $Init$ describes the set of possible initial states, and violation of φ denotes an error.
 - For example, the property φ can assert that the magnitude of the error e between the reference signal and the plant output is less than some constant; a violation of this property would indicate an unacceptable overshoot.

References Sources

[1] Rajeev Alur, Principles of Cyber-Physical Systems, MIT Press, 2015

(Biblioteca de Matematica si Informatica, UBB,

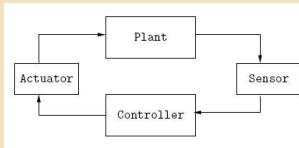
- Rajeev Alur, Principles of Cyber-Physical Systems, 2015
COTA:9491
- <https://www.bcucluj.ro/ro/despre-noi/filiala/biblioteca-de-matematic%C4%83-%C5%9Fi-informatic%C4%83>

CMES – Today

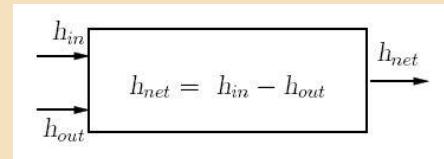
Bring it All Together

Dynamical systems

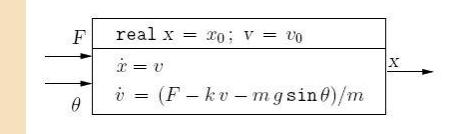
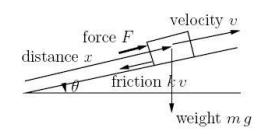
- Continuous-time Models
- Continuously Evolving Inputs and Outputs



- Continuous-time component NetHeat

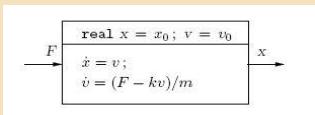


- Models with Disturbance

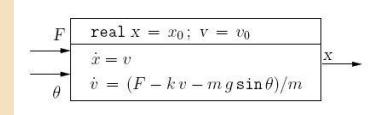


Designing Controllers

Open-loop Controller



Feedback Controller



- Analysis Techniques
- Numerical Simulation
- Computing reachable states

Thank You For Your Attention!

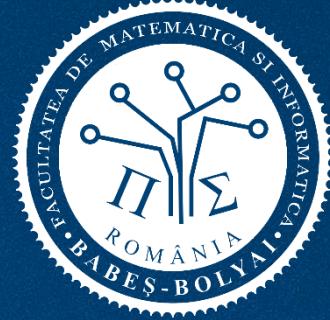
- ExitTicket
- Mentimeter
 - menti.com
 - Code:



Next Lecture

- Discussion about Exam





Software Systems Verification and Validation

“Tell me and I forget, teach me and I may remember, involve me and I learn.”

(Benjamin Franklin)