Cyber-Physical Systems

Laura Nenzi

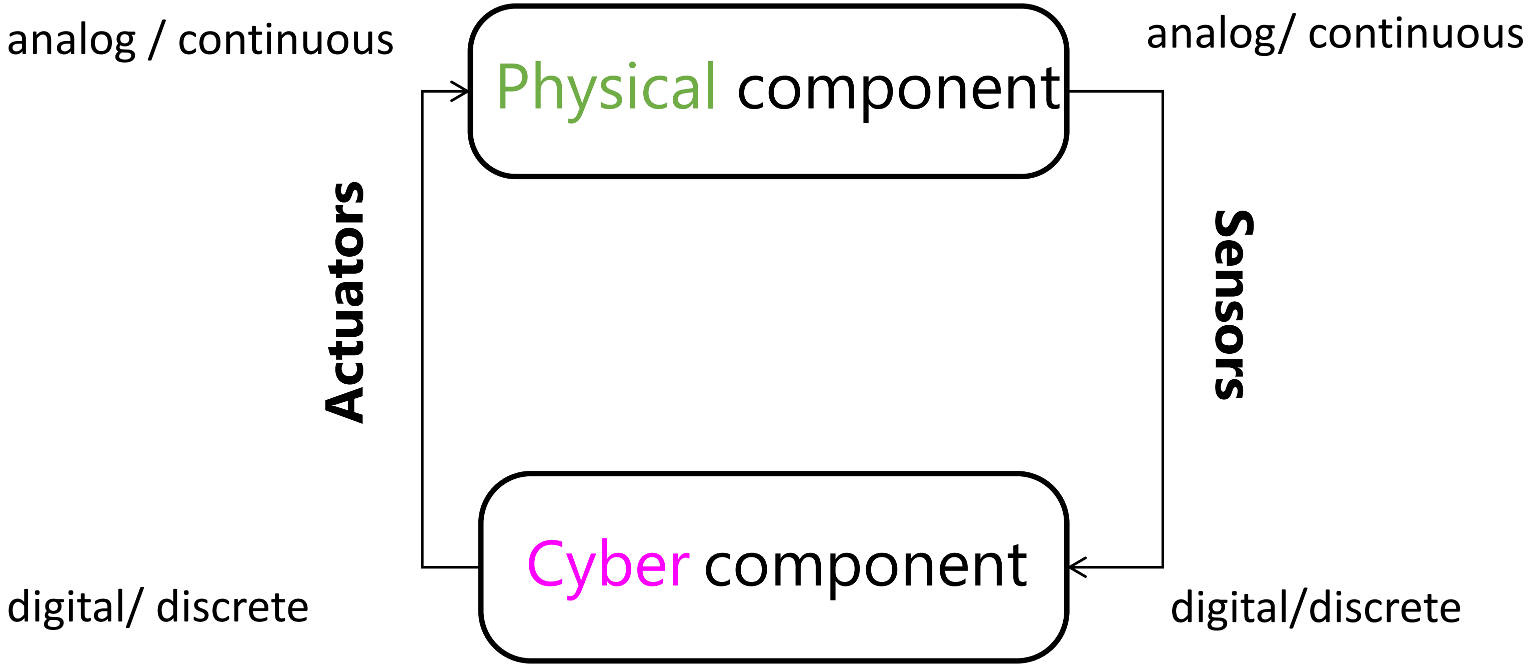
Università degli Studi di Trieste
I Semestre 2023

Lecture 2: Modeling (Introduction)
Dynamical Systems

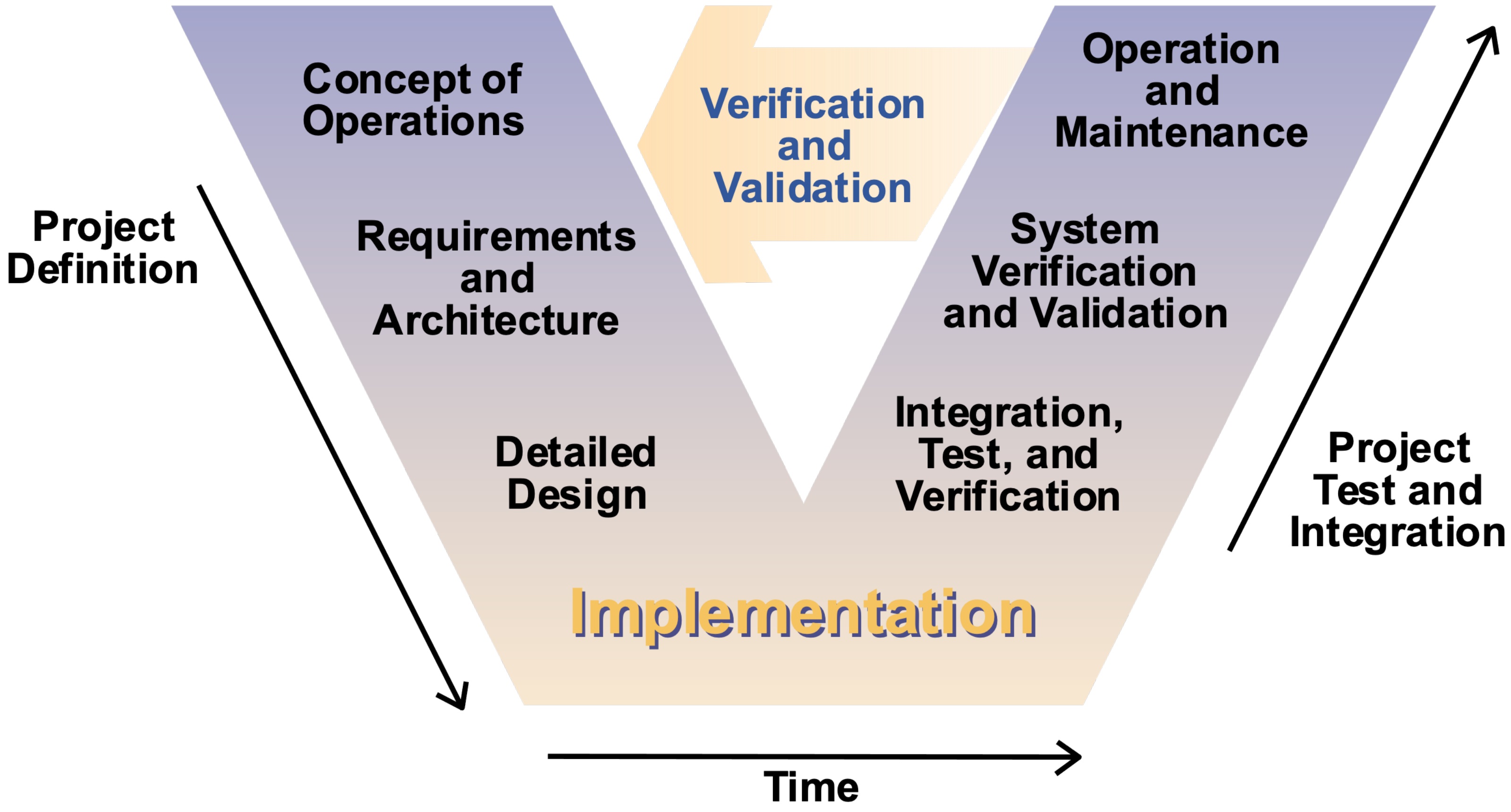
[Many Slides due to J. Deshmukh, Toyota]

Cyber-Physical System (CPS)

Combination of physical (environment / plant / process / system) with a cyber (computation / software / code) components **potentially networked** and **tightly interconnected**



Model-based Design Approach

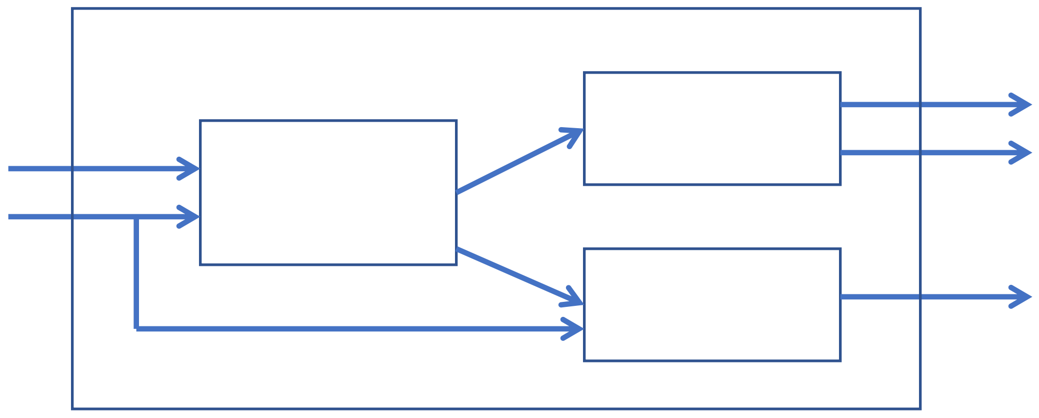


Validation : "Are you building the right thing?"

Verification : "Are you building it right?"

Model-based Design Approach

MBD languages are often visual and block-diagram based, e.g. Simulink



Functional Components

1. *Classical* model of computation: Functional or Transformational Programs

   - Start from a given input,
   - Produce a certain output and then **terminate**
   - Desired functionality can be described by a mathematical function
   - Emphasis is on data computation

Reactive Components

2. Reactive Programs:

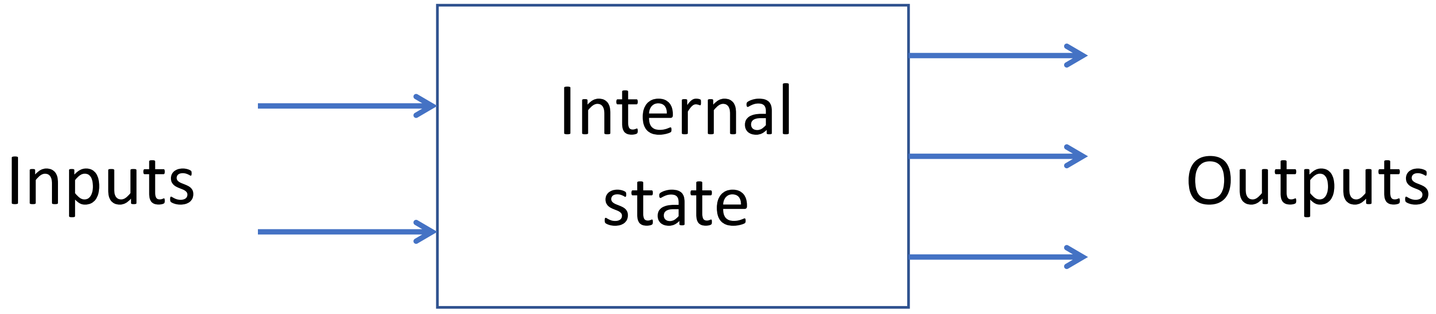
- It maintains and internal state
- Continuously interact with the environment at a rate decided by the environment
- Emphasis is on system-environment interaction; e.g. airline autopilot, mail-servers, etc.

Models of Computation: Timing

- What's the notion of time in the model?
  - Real-time or Logical time-steps of execution?
- What time do different components in the model use?
  - Single global clock for full synchronization?
  - Different clocks in each component?
- What level of granularity do we need in time?
  - Discrete time-steps or Continuous dense time?

Reactive Component

Most convenient model of computation for an (Autonomous) CPS is a ***reactive and concurrent model of computation.***



An autonomous CPS can be viewed as a ***network of components*** *that communicate either* ***synchronously*** *or* ***asynchronously.***

Models: abstractions of CPS

Examples of type of modeling for CPS components:

- Modeling physical phenomena (dynamical systems) – differential equation
- Feedback control systems – time-domain modeling
- Modeling modal behavior – FSMs, hybrid automata, …
- Modeling sensors and actuators – models that help with calibration, noise elimination,
- Modeling hardware and software – capture concurrency, timing, …
- Modeling networks – latencies, error rates, packet loss,

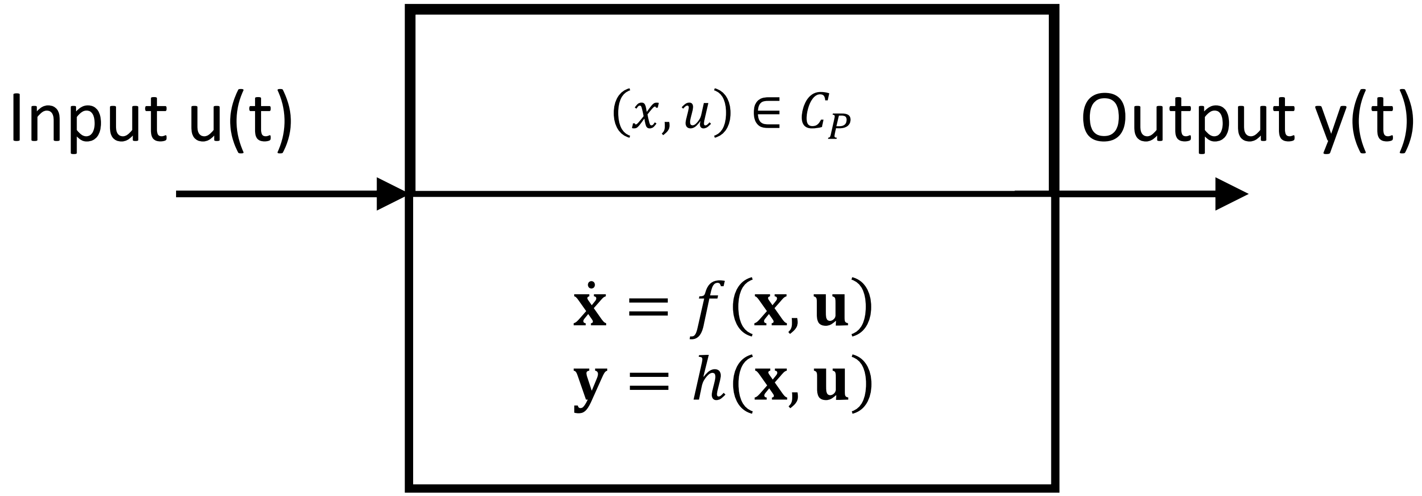
Models of Computation

- Continuous-time models/Dynamical system models
  - Like Synchronous, but time evolves continuously
- Synchronous Model of Computation
- Asynchronous Model of Computation
- Timed Models
  - Like Asynchronous models, but with explicit time information
  - Can make use of global time for coordination
- Hybrid Dynamical Models

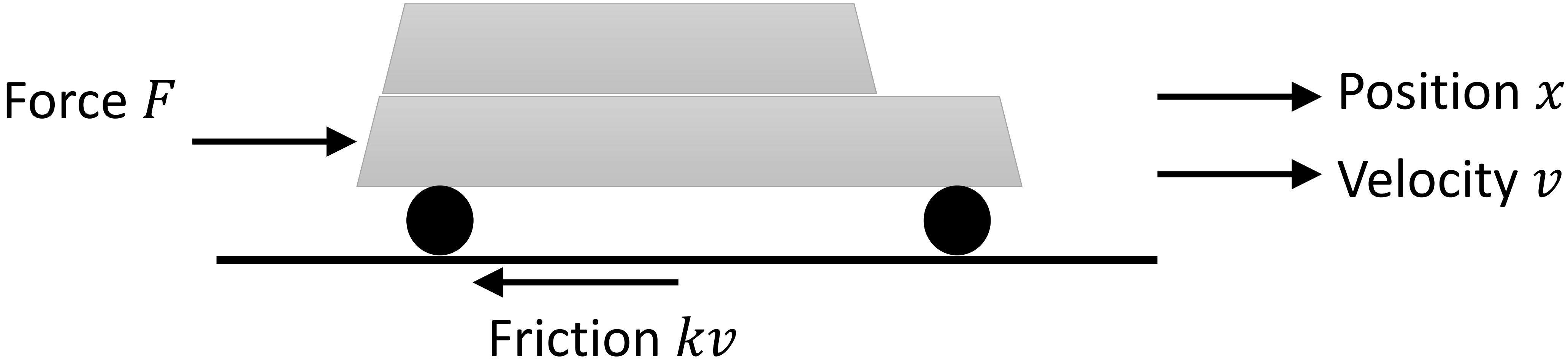
Dynamical Systems

- Most natural model for describing most physical systems
- Continuous/discrete systems that continuously evolve over time
- It is represented by differential equations that involve the rates of change of quantities
- Quantities describe the state of the phenomena, modeled as state variables
  - Pressure, Temperature, Velocity, Acceleration, Current, Voltage, etc.
- Could include algebraic relations between state variables

Continuous-time component (differential)



Model of a simple car



Newton's law of motion: $F = m\frac{d^2x}{dt^2} + kv$; $v = \frac{dx}{dt}$

State-Space representation

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$$
$$\mathbf{y} = h(\mathbf{x}, \mathbf{u})$$

Example:

Convert

$$\dot{x} = v(t)$$
$$\dot{v} = \frac{F(t) - kv(t)}{m}$$

- It is numerically efficient to solve
- It can handle complex systems
- It allows for a more geometric understanding of dynamic systems
- It forms the basis for much of modern control theory

State-Space representation

All derivatives are with respect to single independent variable, often representing time.

Order of ODE is determined by highest-order derivative of state variable function appearing in ODE

ODE with higher-order derivatives can be transformed into equivalent first-order system.

$$x^{(k)} = f(x, \dots, x^{(k-1)})$$

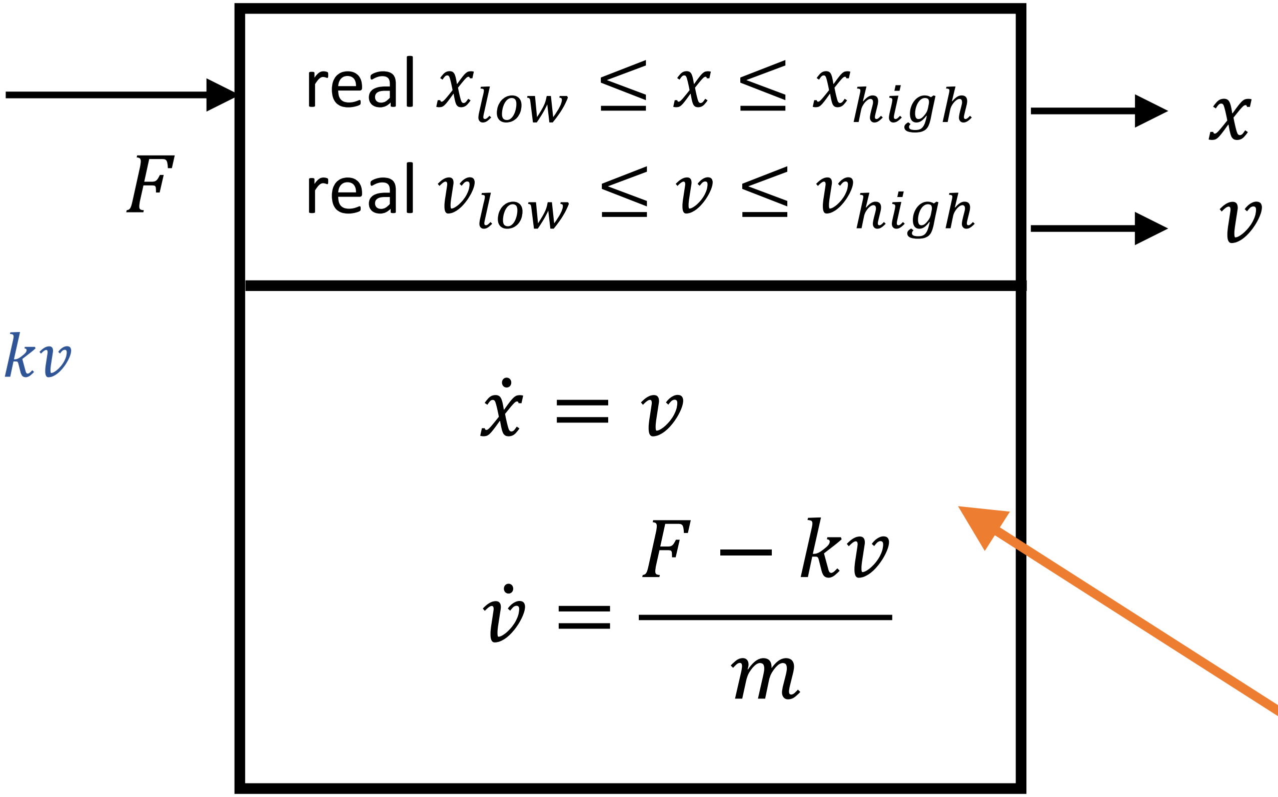
$$z_1 = x,\ z_2 = \dot{x}, \dots, z_k = x^{(k-1)}$$

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \\ . \\ . \\ . \\ \dot{z}_k \end{bmatrix} = \begin{bmatrix} z_2 \\ z_3 \\ z_4 \\ . \\ . \\ . \\ f(x, \dots, x^{(k-1)}) \end{bmatrix}$$

Model of a simple car

$$m\frac{d^2x}{dt^2} = F - kv$$

$$v = \frac{dx}{dt}$$



- Rate of change of each state variable and output variables defined using expressions over inputs and states

Expressions, not assignments!

Executions of Car

- Let $\mathbb{T}$ represent a set representing time instants, i.e. $\mathbb{T} \subseteq \mathbb{R}^{\geq 0}$
- Input Signal: Function F from $\mathbb{T} \rightarrow \mathbb{R}$
  - Input signal is assumed to be continuous or piecewise-continuous
- Given an initial state (x_0, v_0) and an input signal $F(t)$, the execution of the system is defined by ***state-trajectories*** $x(t)$ and $v(t)$ (from $\mathbb{T}$ to $\mathbb{R}$) that satisfy the ***initial-value problem***:
  - $x(0) = x_0; v(0) = v_0$
  - $\dot{x} = v(t); \dot{v} = \frac{F(t) - kv(t)}{m}$

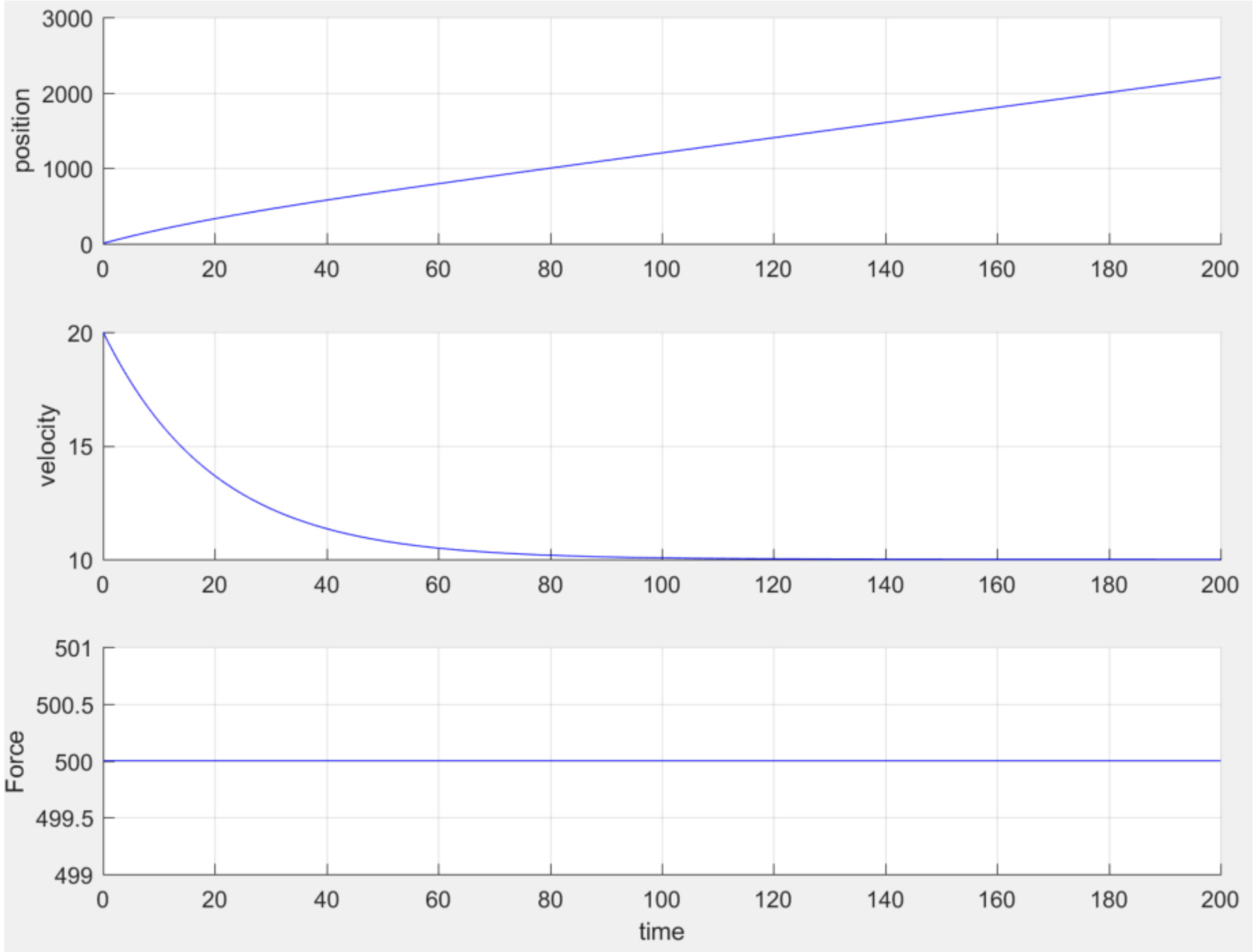
Sample Execution of Car

Suppose $\forall t$: $F(t) = 0, x_0 = 5\text{ m}, v_0 = 20\text{ m/s}, m = 1000\text{kg}, k = 50Ns/m$

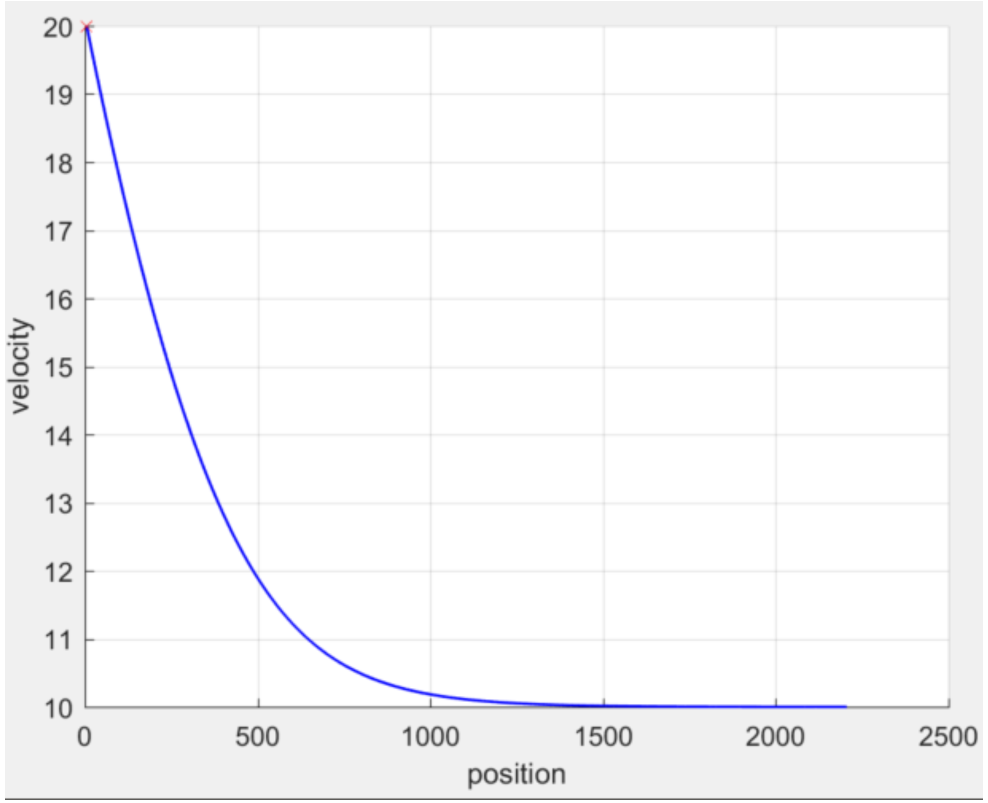
- Then, we need to solve:
  - $x(0) = 5; v(0) = 20$
  - $\dot{x} = v; \dot{v} = -\frac{kv}{m}$

- Solution to above differential equation (solve for v first, then x):
- $v(t) = v_0 e^{-\frac{kt}{m}}; x(t) = \frac{mv_0}{k}\left(1 - e^{-\frac{kt}{m}}\right)$
- Note, as $t \to \infty$, $v(t) \to 0$, and $x(t) \to \frac{mv_0}{k}$

Plots



position
3000
2000
1000
0
velocity
20
15
10
Force
501
500.5
500
499.5
499
0
20
40
60
80
100
120
140
160
180
200
time



velocity
20
19
18
17
16
15
14
13
12
11
10
0
500
1000
1500
2000
2500
position

Differential Equation

The state of the system is characterized by state variables, which describe the system. The rate of change is (usually) expressed with respect to time

Simple Example: Temperature equations

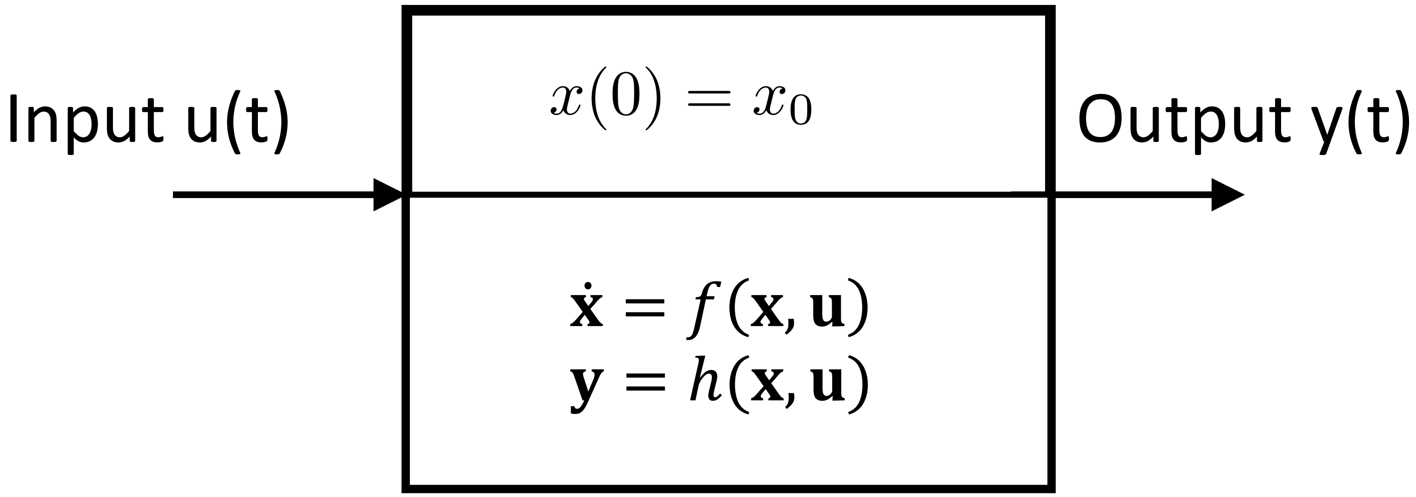
$$\frac{dT}{dt} = -aT + T_{ext} + K_H u$$

Continuous-Time Component Definition

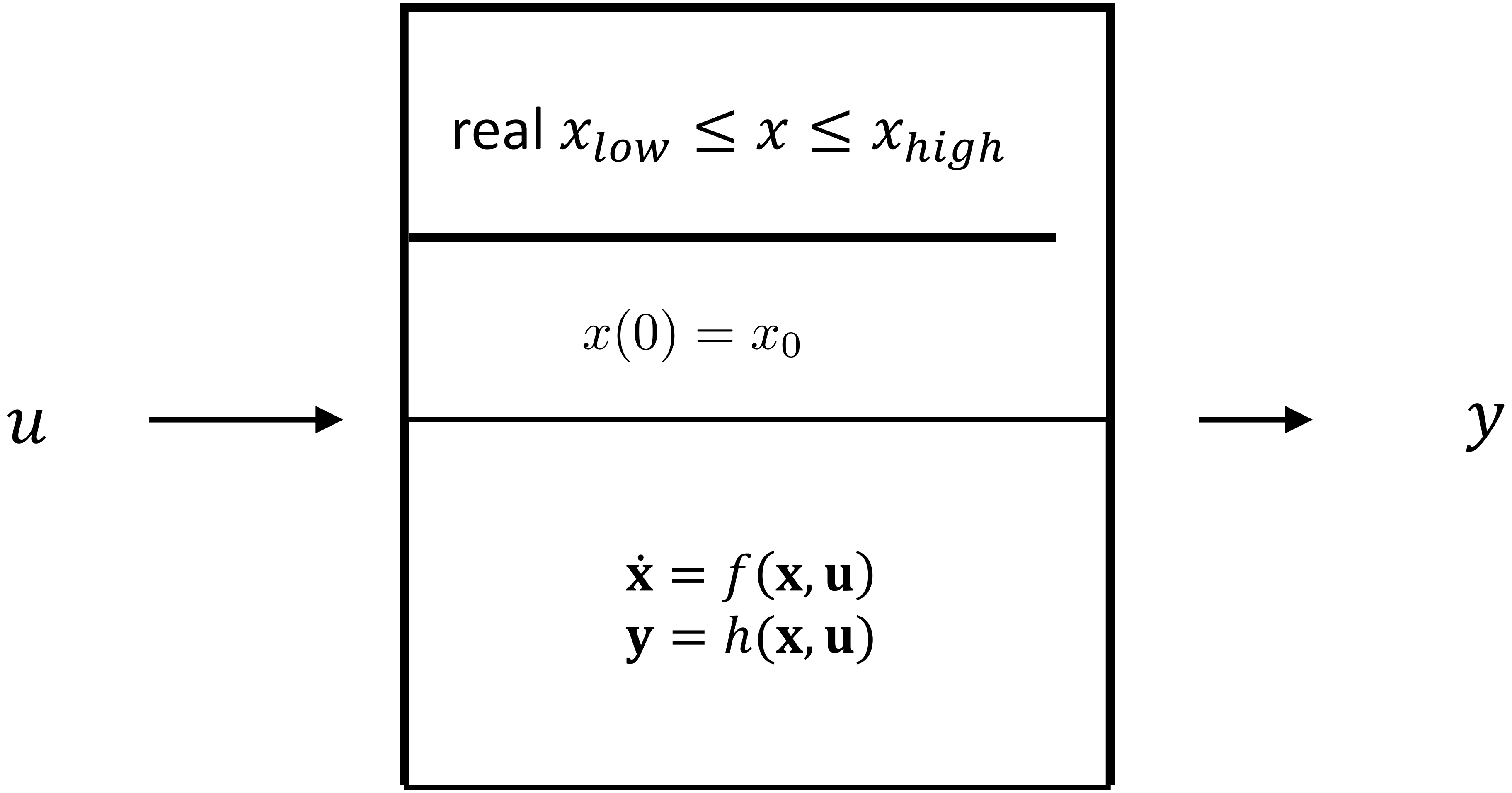
- Set I of real-valued input variables
- Set O or real-valued output variables
- Set X of real-valued (continuous) state variables
- Initialization $Init$ specifying a set $X_0$of initial values for states
- Dynamics: for each state variable, x, a real valued expression f over I and X
- Output Function: for each output variable, y, a real valued expression h over I and X.

Execution Definition

- Convention: $\mathbf{x} = (x_1, x_2, \dots x_n), \mathbf{y} = (y_1, y_2, \dots, y_m)$
- Given an input signal $u\colon \mathbb{T} \to \mathbb{R}$, an execution consists of a *differentiable* state signal $\mathbf{x}(\mathrm{t})$, and an output signal $\mathbf{y}(t)$, such that:
  1. $\mathbf{x}(0) \in X_0$
  2. For each output variable y and time t, $y(t) = h\big(u(t), x(t)\big)$
  3. For each state variable x, $\frac{d}{dt}x(t) = f(u(t), x(t))$



Order Differential Equation



Existence and Uniqueness of Solutions

- Given an input signal $u(t)$, when are we guaranteed that the system has at least one execution? Is there nondeterminism in continuous-time components?
- Input signal should be piecewise-continuous, and additional conditions need to be imposed on the RHS of dynamics (f) and output functions (h)
- Related to solutions for the initial value problem in the classical theory of ODEs

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$$
$$\mathbf{y} = h(\mathbf{x}, \mathbf{u})$$

Existence

- There exists at least one solution $\mathbf{x}(t)$ if the function f is continuous

- Definition of continuity uses notion of distance between points
  - Euclidean distance: $d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_2 = \sqrt{(x_1 - y_1)^2 + \cdots + (x_n - y_n)^2}$

- f is continuous if for all $\mathbf{x} \in \mathbb{R}^n$, for all $\epsilon > 0$, there exists a $\delta > 0$, such that for all $\mathbf{y} \in \mathbb{R}^n$, if $\|\mathbf{x} - \mathbf{y}\|_2 < \delta$, then $\|f(\mathbf{x}) - f(\mathbf{y})\|_2 < \epsilon$.

- Example when solution does not *globally* exist:
  - $\frac{dx}{dt} = 1/t$

Uniqueness

- Solution to initial value problem is unique if f is Lipschitz continuous
- Lipschitz-continuity is a stronger version of continuity: upper bounds how fast a function can change
- Function f is **Lipschitz-continuous** if there exists a constant L (called the Lipschitz constant) such that:

$$\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n: \|f(\mathbf{x}) - f(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|$$

- Examples:
  - Linear functions (e.g. $x_1 - 3x_2$) are Lipschitz continuous
  - Functions: x^2, $\sqrt{x}$ are not Lipschitz continuous over $\mathbb{R}^n$

- Can restrict $\mathbb{T}$ and X to some bounded and closed set such that f is piecewise-continuous and Lipschitz to get unique solutions over such compact domains

What do numeric solvers/simulators do?

- Allow modeling arbitrarily complex functions: even functions with unbounded discontinuities
- May not be even possible to check for Lipschitz conditions for what's implemented in a Matlab function/Simulink model
- Rely on numerical integration schemes/solvers to obtain solutions
  - ode45, ode23, ode15, etc.

Linear Components

- Special kind of dynamical system

$$\dot{\mathbf{x}} = \boldsymbol{f}(\mathbf{x}, \mathbf{u})$$
$$\boldsymbol{y} = \boldsymbol{h}(\mathbf{x}, \mathbf{u})$$

  - f is of the form $a_1x_1 + \cdots + a_nx_n + b_1u_1 + \cdots + b_mu_m$ or compactly, $f = A\mathbf{x} + B\mathbf{u}$
  - h is of the form $c_1x_1 + \cdots + c_nx_n + d_1u_1 + \cdots + d_mu_m$ or compactly, $h = C\mathbf{x} + D\mathbf{u}$
- Linear algebra was invented to reason about linear systems!
- Linear systems have many nice properties:
  - Many analysis methods in the frequency domain (using Fourier/Laplace transform methods)
  - Superposition principle (net response to two or more stimuli is the sum of responses to each stimulus)

Linear Systems

- Equation of simple car dynamics can be written compactly as:

$$\begin{bmatrix} \dot{x} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -k/m \end{bmatrix} \begin{bmatrix} x \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} [F]$$

- Letting $A = \begin{bmatrix} 0 & 1 \\ 0 & -k/m \end{bmatrix}$, $B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, we can re-write above equation in the form:

- $\dot{\mathbf{x}} = A\mathbf{x} + \mathrm{B}\mathbf{u}$, where $\mathbf{x} = [x \quad v]$, and $\mathbf{u} = [F]$

Solutions to Linear Systems

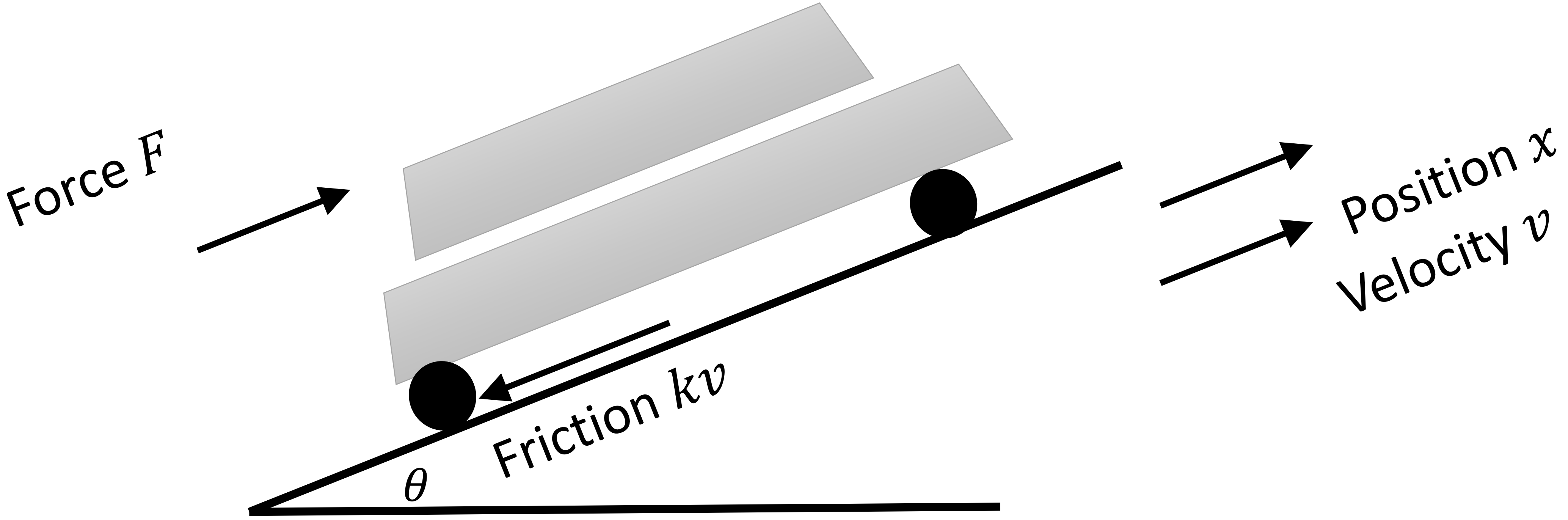
- ***Autonomous*** linear system has no inputs: $\dot{\mathbf{x}} = A\mathbf{x}$

- Solution of autonomous linear system can be fully characterized:
  - $\mathbf{x}(t) = e^{At}\mathbf{x}_0$
  - Computing e^A is easy if A is a diagonal matrix (non-zero elements are only on the diagonal)

- For a linear system with ***exogenous*** inputs?
  - $x(t) = e^{At}x_0 + \int_0^t e^{A(t-\tau)}Bu(\tau)d\tau$

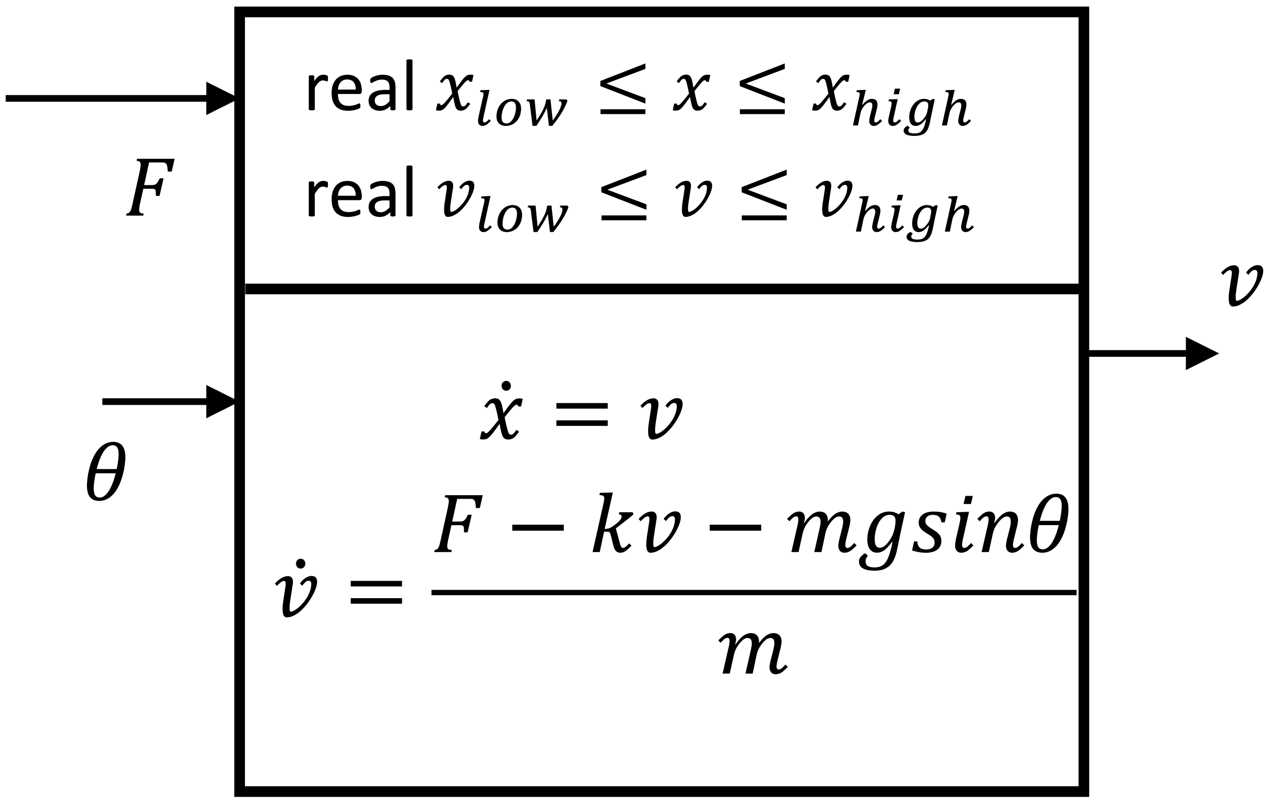
- In practice, numerical integration methods outperform matrix exponential

Model with disturbance



Newton's law of motion: $F = m\frac{d^2x}{dt^2} + kv + mg\sin(\theta)$

Model with disturbance



Time Invariant System

The system is time invariant because the output does not depend on the particular time the input is applied.

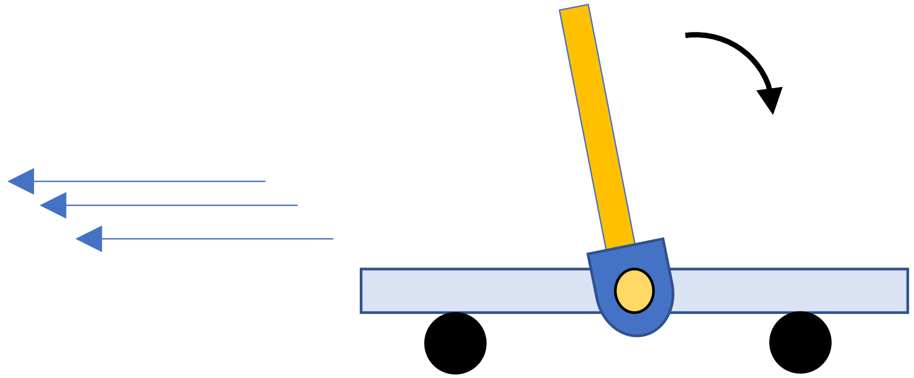
f does not depend on time

$$\frac{dx}{dt} = \dot{x} = f(x, u)$$

The underlying physical laws themselves do not typically depend on time.

Stability of Systems

- Property capturing the ability of a system to return to a quiescent state after perturbation
  - Stable systems recover after disturbances, unstable systems may not
  - Almost always a desirable property for a system design
- Fundamental problem in control: design controllers to *stabilize* a system

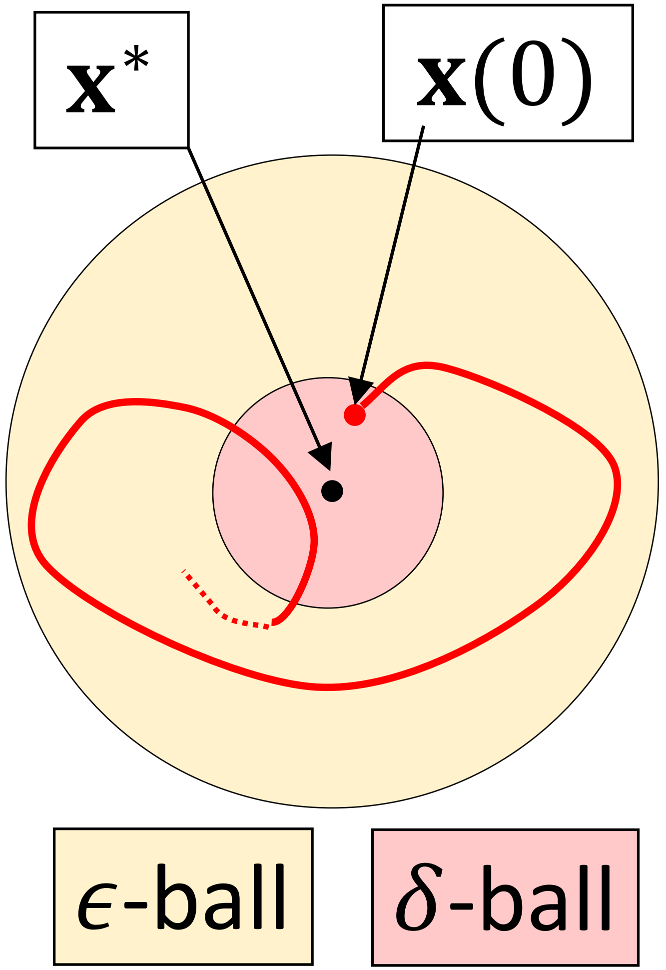


- Problem: Cart-pole is inherently unstable, aim: keep it upright
- Solution Strategy: Move cart in direction in the same direction as the pendulum's falling direction
- Design problem: Design a controller to stabilize the system by computing velocity and direction for cart travel

Lyapunov stability

Solutions starting δ close from equilibrium point must remain close (within ϵ) forever

- System $\dot{\mathbf{x}} = f(\mathbf{x})$ with f Lipschitz continuous
- Equilibrium point when $f(\mathbf{x})$ is zero (say $\mathbf{x}^*$)
- Equilibrium point $\mathbf{x}^*$ is Lyapunov-stable if:
  - For every $\epsilon > 0$,
    - There exists a $\delta > 0$, such that
      - if $\|\mathbf{x}(0) - \mathbf{x}^*\| < \delta$, then,
      - for every $t \geq 0$, we have $\|\mathbf{x}(t) - \mathbf{x}^*\| < \epsilon$



Asymptotic Stability

Solutions not only remain close, but also converge to the equilibrium

- System $\dot{\mathbf{x}} = f(\mathbf{x})$
- Equilibrium point $\mathbf{x}^*$ is asymptotically-stable if:
  - $\mathbf{x}^*$ is Lyapunov-stable +
  - There exists $\delta > 0$ s.t. if $\|\mathbf{x}(0) - \mathbf{x}^*\| < \delta$, then $\lim_{t\to\infty} \|\mathbf{x}(t) - \mathbf{x}^*\| = 0$

Exponential Stability

Solutions not only converge to the equilibrium, but in fact converge at least as fast as a known exponential rate

- All stable linear systems are exponentially stable
- This need not be true for nonlinear systems!

- System $\dot{\mathbf{x}} = f(\mathbf{x})$
- Equilibrium point $\mathbf{x}^*$ is exponentially-stable if:
  - $\mathbf{x}^*$is asymptotically stable +
  - There exist $\alpha > 0, \beta > 0$ s.t. if $\|\mathbf{x}(0) - \mathbf{x}^*\| < \delta$, then for all $t \geq 0$:

$$\|\mathbf{x}(t) - \mathbf{x}^*\| \leq \alpha\|\mathbf{x}(0) - \mathbf{x}^*\|e^{-\beta t}$$

Analyzing stability for linear systems

- Eigenvalues of a matrix A:
  - Value λ satisfying the equation $A\mathbf{v} = \lambda\mathbf{v}$. $\mathbf{v}$ is called the eigenvector
  - Equivalent to saying: values satisfying $|A - \lambda I| = 0$, where I is the identity matrix
- Interesting result for linear systems: System $\dot{\mathbf{x}} = A\mathbf{x}$ is asymptotically stable if and only if every eigenvalue of A has a negative real part
- Lyapunov stable if and only if every eigenvalue has non-positive real part
- Nonlinear systems: no simple analysis technique exists
  - Lyapunov's methods allow reasoning about stability of nonlinear systems

Stability analysis example for linear systems

- Manual way: solve the characteristic equation of the matrix A
- $A = \begin{bmatrix} 1 & -1 \\ 3 & 2 \end{bmatrix}$
- Characteristic equation: $|A \ - \lambda I| = 0$, i.e.
- $\begin{vmatrix} 1-\lambda & -1 \\ 3 & 2-\lambda \end{vmatrix} = 0$, or $(1-\lambda)(2-\lambda)+3=0$
- $(\lambda^2 - 3\lambda + 2 + 3) = 0$
- i.e., $\lambda = \frac{\left(3 \pm \sqrt{9-4\times 5}\right)}{2} = 1.5 \pm 1.65i$
- Real part is positive $\Rightarrow$ A represents an unstable linear system

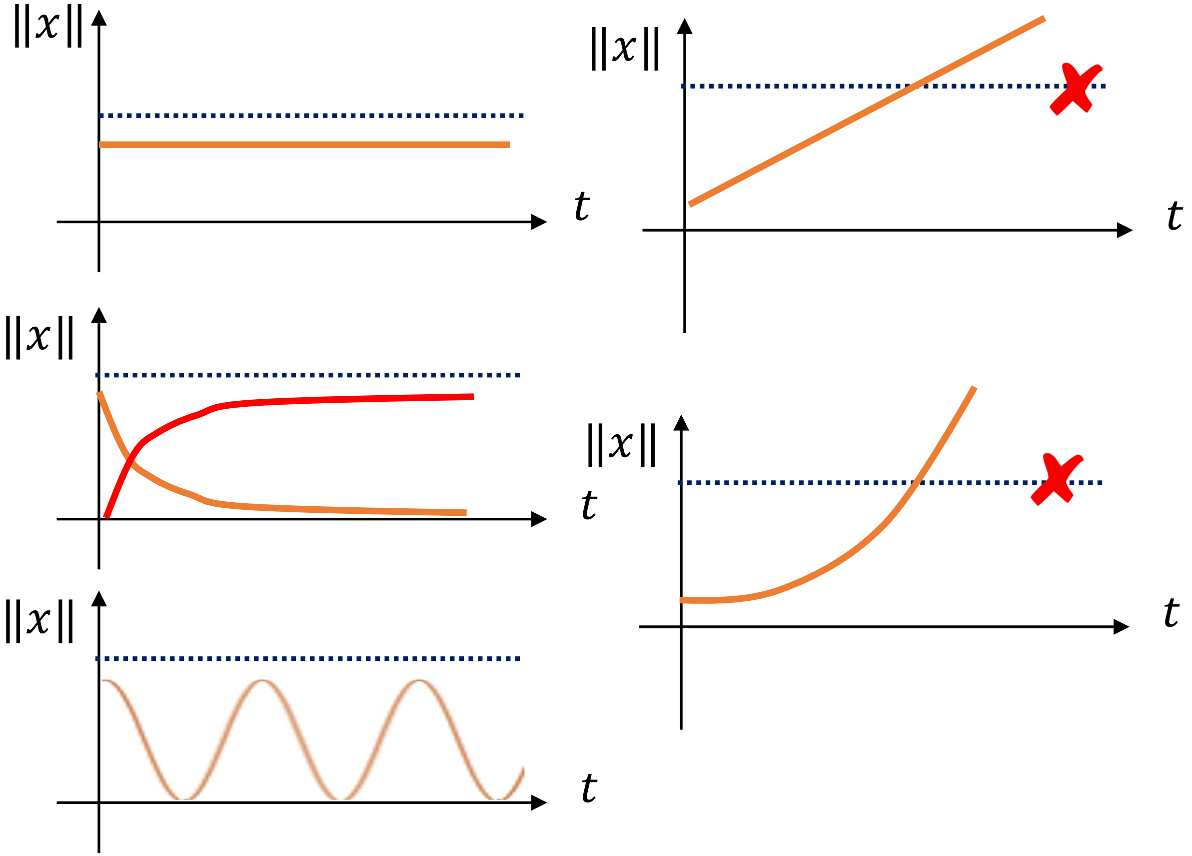
Stability analysis example for linear systems

- $A = \begin{bmatrix} 1 & -1 \\ 3 & -2 \end{bmatrix}$

- Characteristic equation: $|A \ - \lambda I| = 0$, i.e.
- $\begin{vmatrix} 1-\lambda & -1 \\ 3 & -2-\lambda \end{vmatrix} = 0$, or $(1-\lambda)(-2-\lambda) + 3 = 0$
- $(\lambda^2 + \lambda - 2 + 3) = 0$
- i.e., $\lambda = \frac{(-1 \pm \sqrt{-3})}{2} = -0.5 \pm i\sqrt{3}$
- Real part is negative $\Rightarrow$ A represents a stable linear system

Bounded signals

- A signal $\mathbf{x}$ is bounded if there is a constant c, s.t. $\forall t$: $\|\mathbf{x}(\mathrm{t})\| < \mathrm{c}$
- Bounded signals:
  - Constant signal : $x(t) = 1$
  - Exponential signal: $x(t) = ae^{bt}$, for $b \leq 0$
  - Sinusoidal signals: $x(t) = a\sin \omega t$
- Not bounded:
  - $x(t) = a + bt$ for any $b \neq 0$
  - Exponential signal: $x(t) = ae^{bt}$, for $b > 0$



Bounded-Input-Bounded-Output (BIBO) stability

The dynamical system is seen as a transformer, mapping input signals to output signals, and demands that a small change to the input signal should cause only a small change to the output signal.

- A system with Lipschitz-continuous dynamics is BIBO-stable if:
  - For every bounded input $\mathbf{u}(t)$, the output $\mathbf{y}(t)$ from initial state $\mathbf{x}(0) = 0$ is bounded

Helicopter Model continued

- Simple helicopter model:
  - Two rotors: Main rotor gives lift, tail rotor prevents helicopter from spinning
  - Torque produced by tail rotor must perfectly counterbalance friction with main rotor, or the helicopter spins

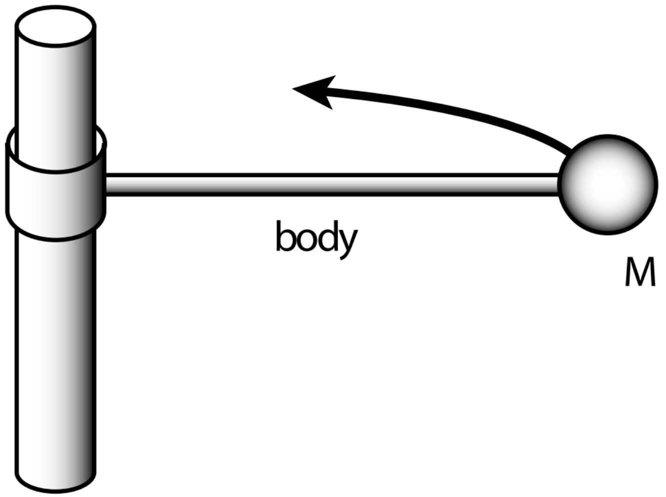
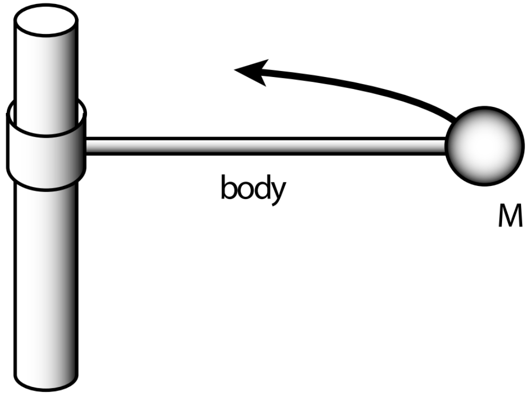


Image credit: From Lee & Seshia: Introduction to Embedded Systems - A Cyber-Physical Systems Approach, http://leeseshia.org/

Helicopter Model continued

- u: net torque on tail of the helicopter – difference between frictional torque exerted by main rotor shaft and counteracting torque by the tail rotor
- y: rotational velocity of the body
- Torque = Moment of inertia × Rotational acceleration
- $\dot{y}(t) = \frac{u(t)}{I}$



$$y(t) = \frac{1}{I}\int_0^t u(\tau)d\tau$$

- What happens when $u(t)$ is a constant input?
- $y(t)$ is not bounded ⇒ helicopter model is not BIBO-stable!