Model Based Design: Continuous Dynamics

01266212
CYBER PHYSICAL SYSTEM DESIGN

SEMESTER 1-2021

Objective

- To be familiar with modeling techniques
- To study about continuous dynamics system and modelling

Topics

- Modeling and CPS Systems
- Modeling of Continuous Dynamic System
- Mathematical Model
- Actor Model

General Concepts about Systems

What is a system?

- Space shuttle, tank, power system, robot, computer, Air-condition, etc.
- A system can contain sub-systems that are themselves systems. (e.g. Power sys. Contains generators, generators are subsystems with different controls, protections, etc.)
- **A system is <u>an object or collection of objects</u> whose properties are of interest.**

We want to study selected properties of these objects.

Why study a system?

- Understand it in order to build it: engineer's point of view
- Satisfy human curiosity (understand more): research's point of view

Example:

Boeing's Model-Based Systems Engineering (MBSE) approach to build a ***Complex Cyber-Physical "System-of-Systems"***

Simulating Success

How do modeling and simulation activities, capabilities benefit Boeing? Let us count the ways—9 of them

By Debby Arkell

Hands-on experience often can be the best way to tackle complex problems or master challenging skills. But when it comes to navigating intricate, variable-laden scenarios, or combat situations involving complex military maneuvers using expensive equipment, "on-the-job training" often is not a prudent approach.

That's why Boeing Integrated Defense Systems, Commercial Airplanes and Phantom Works engage in a wide variety of modeling and simulation activities, designed to provide ever more realistic simulations to internal customers across the enterprise—and to external customers as well.

"There is a tremendous amount of diversity in modeling and simulation being worked on at Boeing, encompassing very complex issues within a very broad spectrum," said Ren Fuchs, director of Modeling and Simulation for IDS. "Right now there are more than

U.S. Navy legacy platform and deliver a comprehensive and credible concept of operations.

Boeing analysts have a variety of tools available—or under development—that can demonstrate concepts and provide significant cost savings by exploring ideas, developing systems, testing and manufacturing within a virtual environment before committing to specific approaches.

4 Product or system testing

Models and simulations are used to test

M&S used to test prototypes in variety of environments.

networked computer systems in the FCS system of systems will be tested in a large-scale distributed simulation facility called the FCS System of Systems Integration Lab. The SoSIL provides a

6 Training systems and maintenance

M&S are used to *train users in the operational environment* – enhancing learning.

Simulation *costs 1/10* of running actual scenarios.

Network communications

Tactical military communications networks—such as Joint

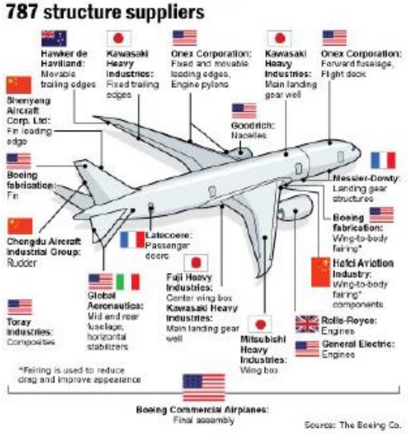
Scale of networks: cost-prohibitive or technically impossible for field tests.

M&S used to test and validate networking protocols in laboratory - environment acting as a test bed.

8

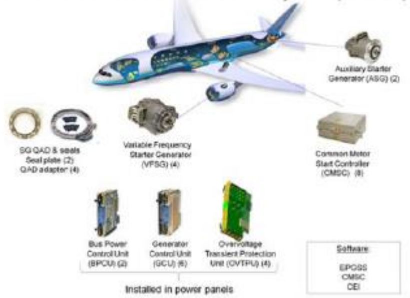
that acts as a distributed virtual test bed, and the Boeing Transformational Communications Laboratory in El Segundo, Calif., performs a similar function for satellite communications.

Large Number of Vendors for the Final System



A Flying Micro-Grid!

Electric Power Generation & Start System (EPGSS)



Cyber-Physical Systems - CPS

CPS: Engineered systems **integrating** cyber and physical components which are modeled as "hybrid" systems:

Cyber components: "follow the rules of algorithms" , examples

- Computation (e.g. computers, IT systems), communication protocols

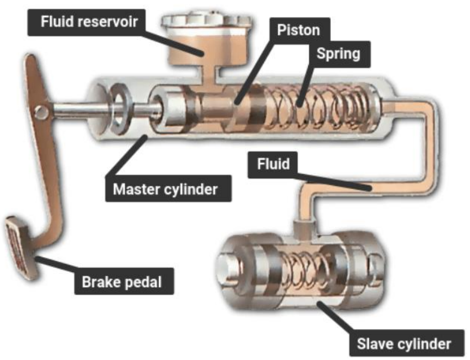
Physical components / systems: "follow the laws of nature" , examples

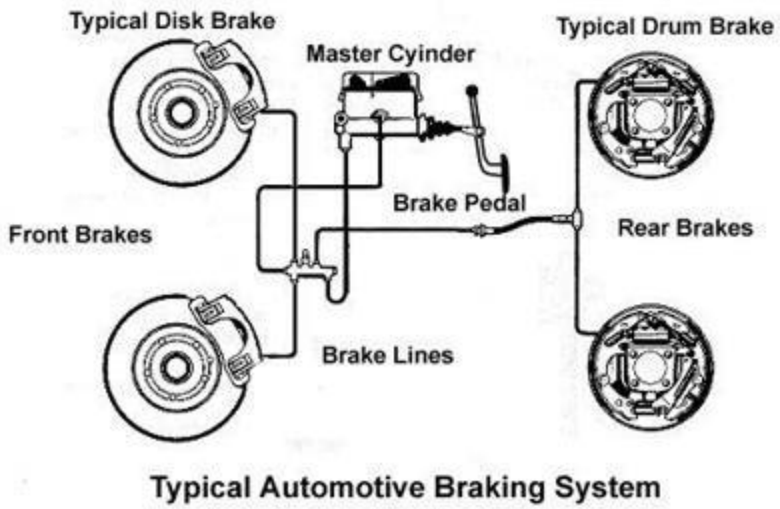
- Thermal Domain:
  - Component: a water heater
  - System: a building's Air Condition (HVAC) system
- IT:
  - Component: Touch Screen
  - System: Mobile Phone

Cyber-physical multi-domain:

- Cyber components: digital control system, digital monitoring sensors, IT…
- Cyber-physical System: power plant as above, but with digital monitoring and control!

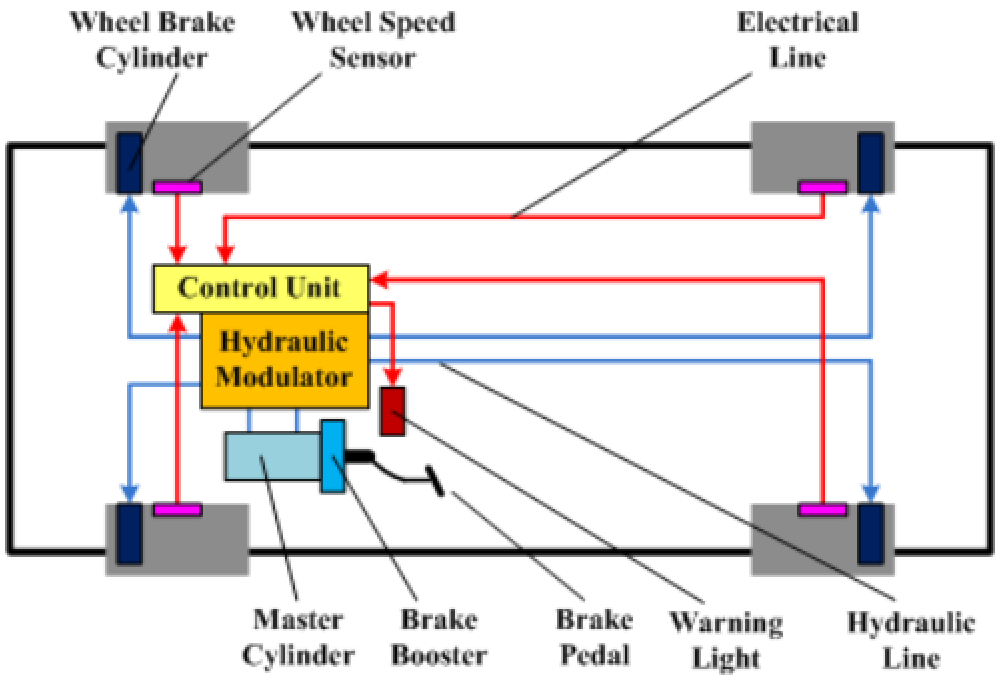
Automotive Braking System





https://www.cartrade.com/blog/2011/auto-guides/brake-systems-in-cars-17.html

Antilock Braking Systems (ABS) Braking System



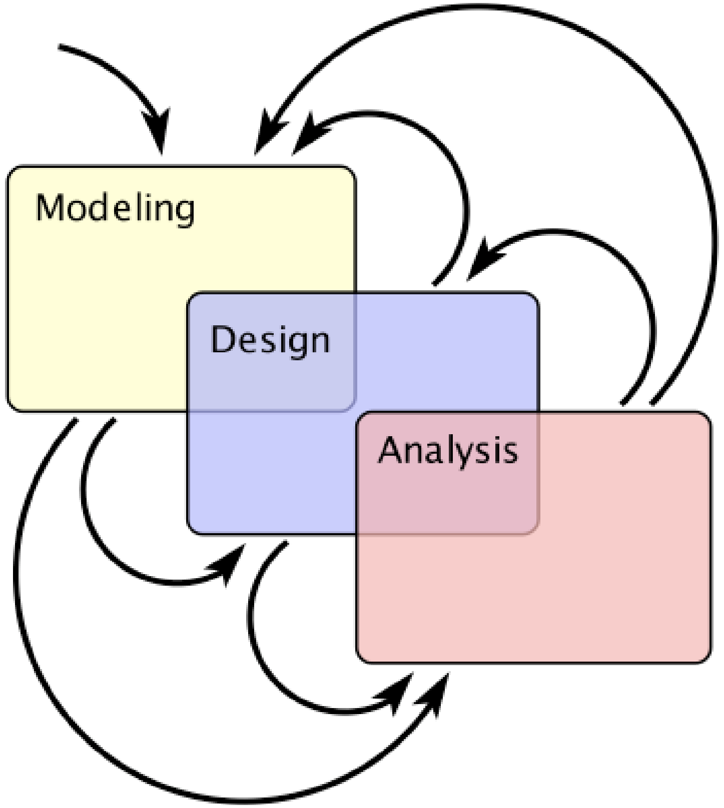
https://cecas.clemson.edu/cvel/auto/systems/braking.html

Modeling, Design, Analysis: An Iterative Process

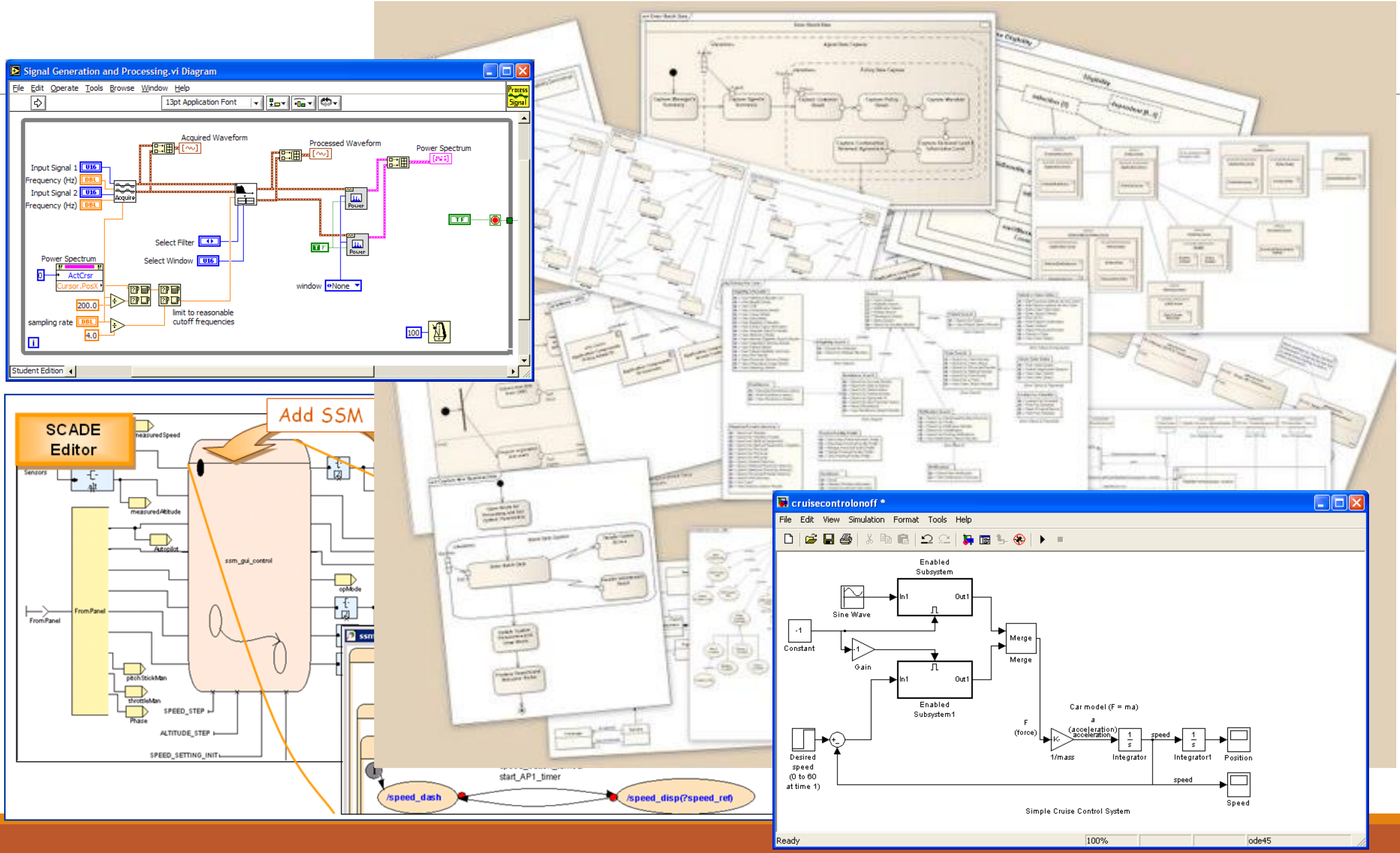
Modeling is the process of gaining a deeper understanding of a system through imitation.
Models specify **what** a system does.

Design is the structured creation of artifacts.
It specifies **how** a system does what it does.
This includes optimization.

Analysis is the process of gaining a deeper understanding of a system through dissection.
It specifies **why** a system does what it does
(or fails to do what a model says it should do).



Focus on Models



Model

Artifact that imitates the system

Mathematical models are Central to the Design Process

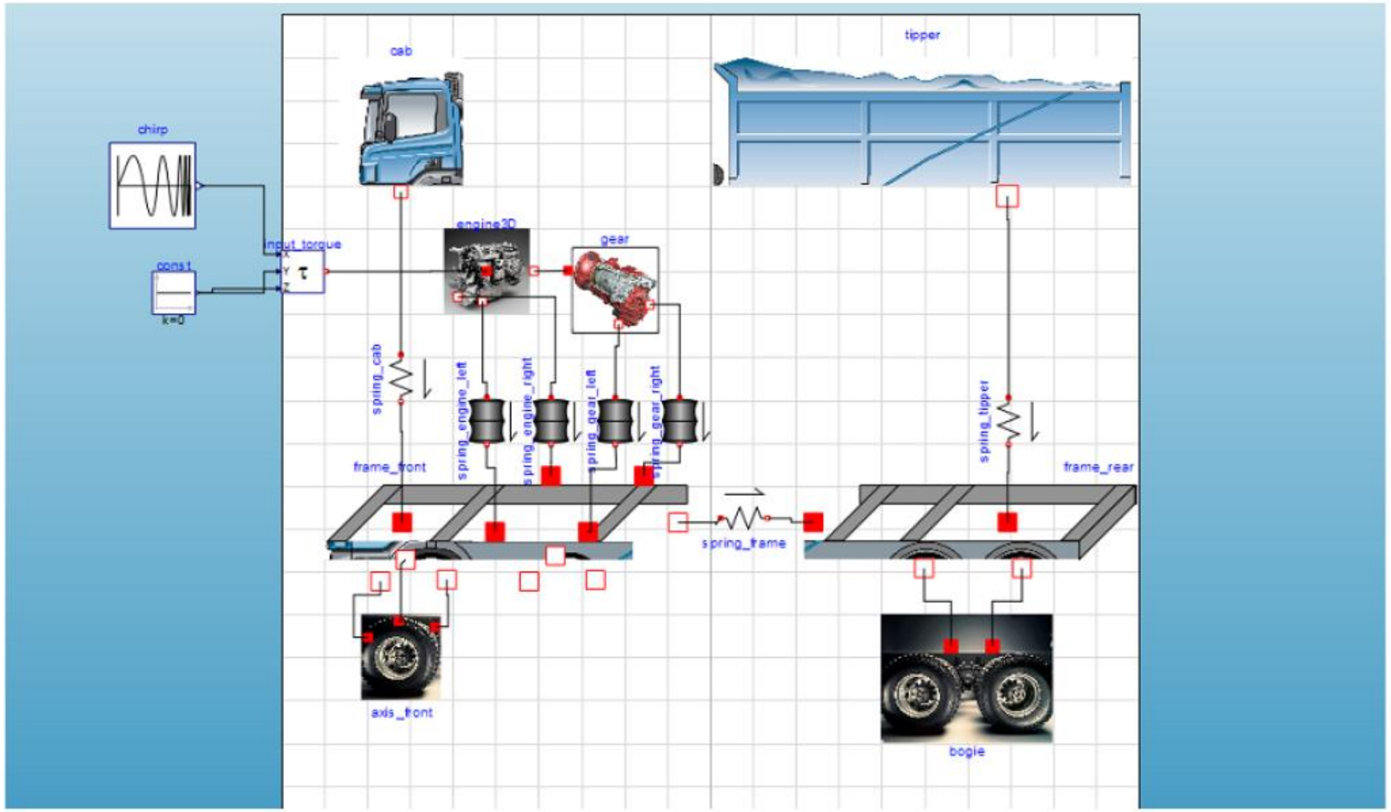
- *Definitions in terms formulas*
- *Mathematical correctness statement*
- *Formal, automatic correctness proof*

But there is no unique approach to Model-Based Design

(at least today).

Model of Track Car System

Modelling a mechanical system on a cyber system.

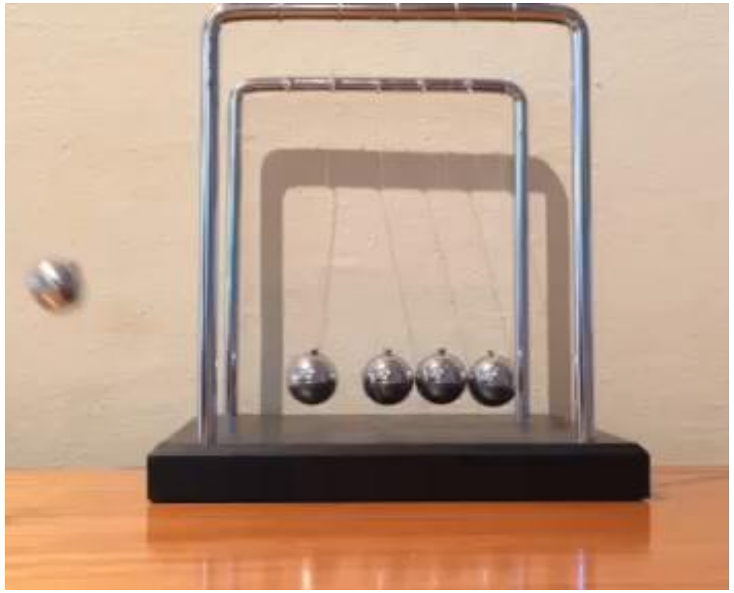


Models vs. Reality

$$x(t) = x(0) + \int_0^t v(\tau)d\tau$$

$$v(t) = v(0) + \frac{1}{m}\int_0^t F(\tau)d\tau,$$

The model



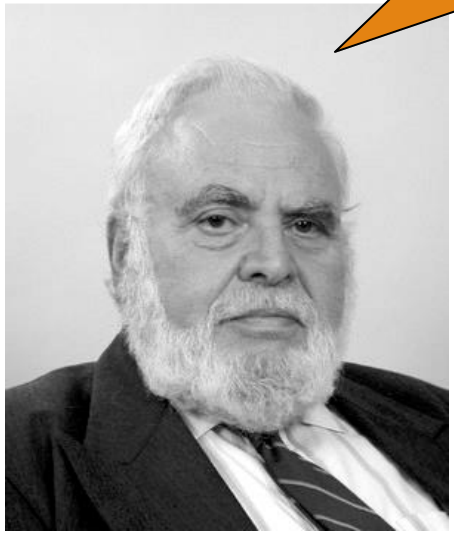
The target (the thing being modeled).

In this example, the *modeling framework* is calculus and Newton's laws.

Fidelity is how well the model and its target match

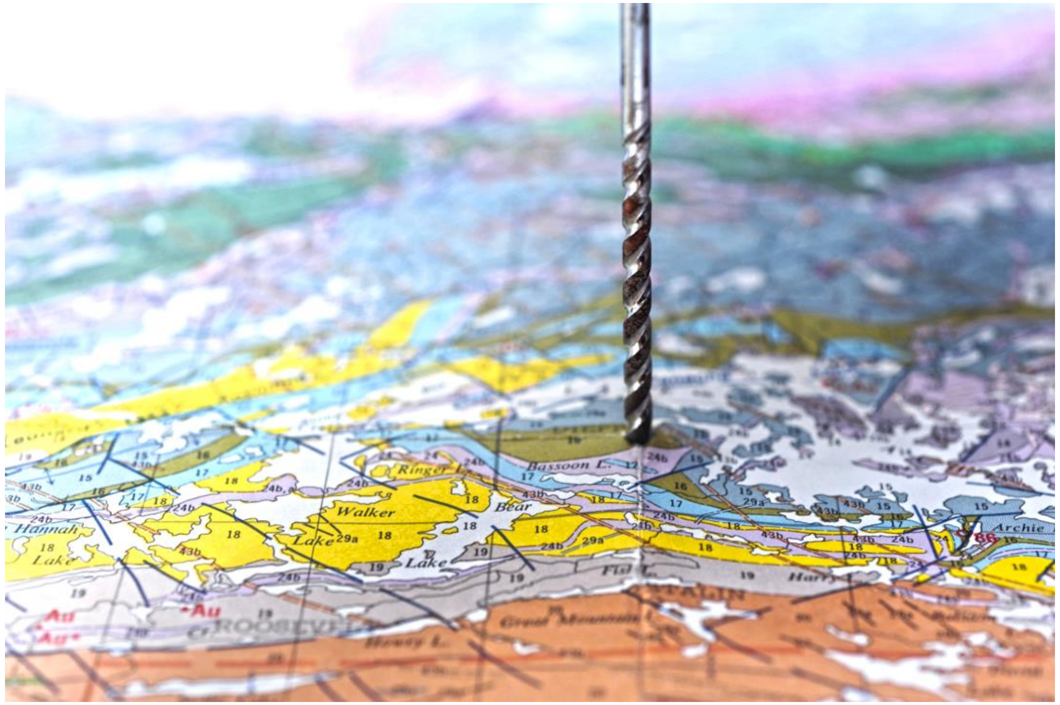
Models vs. Reality

You will never strike oil by drilling through the map!



[Solomon Wolf Golomb]

But this does not in any way diminish the value of a map!



Determinacy

Some of the most valuable models are *deterministic*.

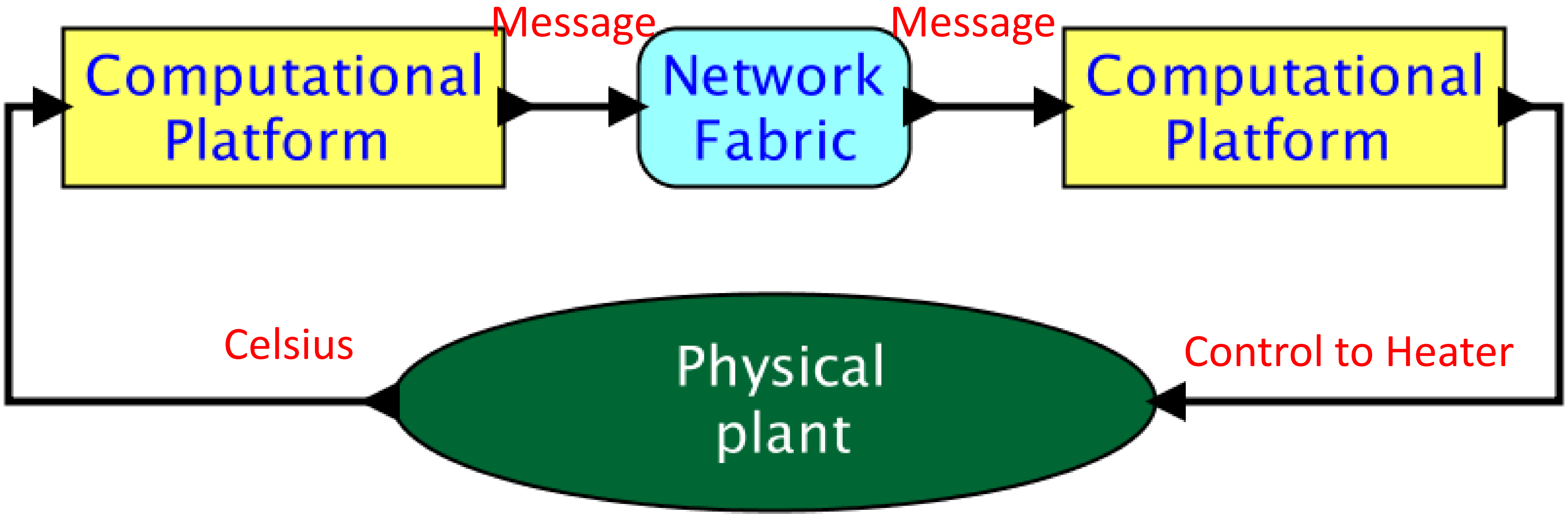
A model is *deterministic* if, given the initial state and the inputs, the model defines exactly one behavior.

Example: Given the voltage and the resistance, the value of current is always the same.

Deterministic models have proven extremely valuable in the past.

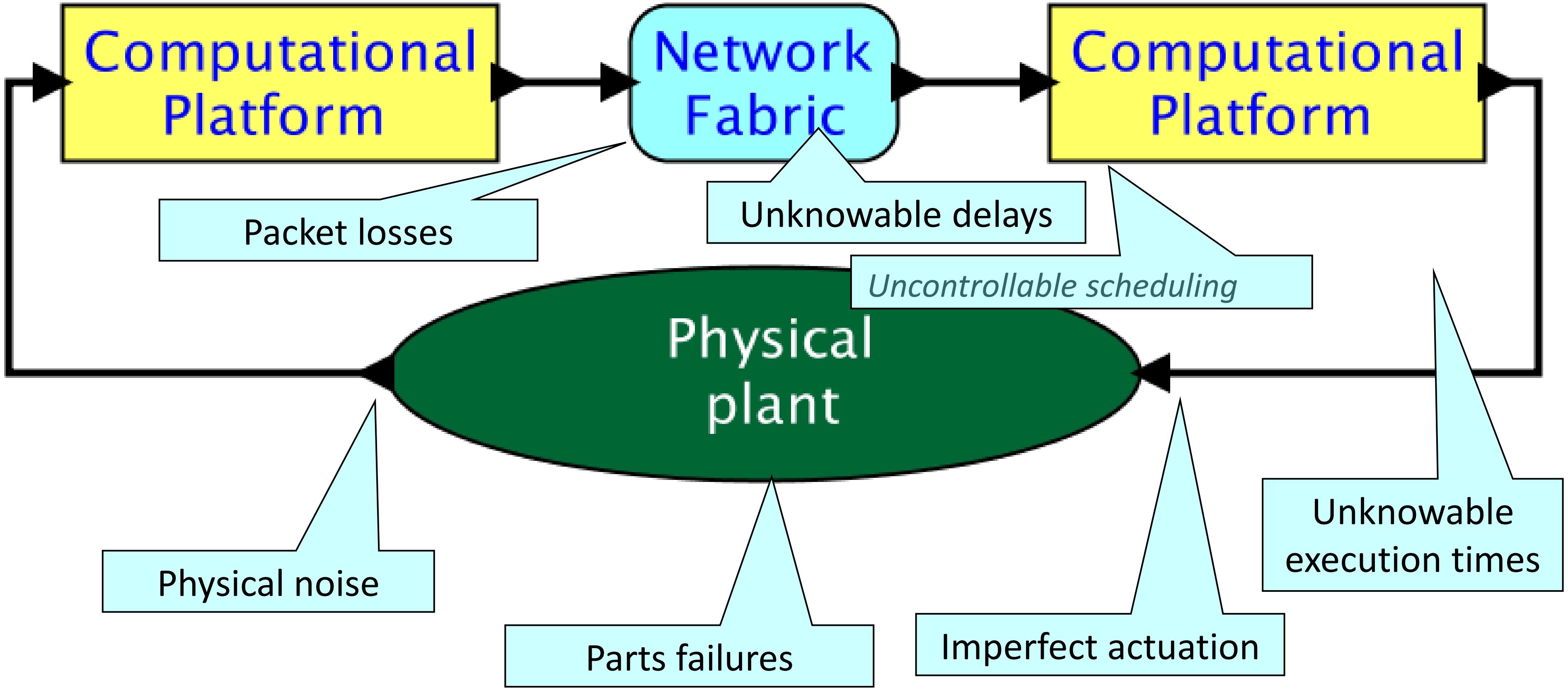
In a nondeterministic framework, the model specifies a family of behaviors.

Schematic of a simple CPS



$Q = m*c*\Delta T$

Do deterministic models make sense for Cyber-physical systems?



A Model Need not be *True* to be *Useful*

“Essentially, all models are wrong,
but some are useful.”

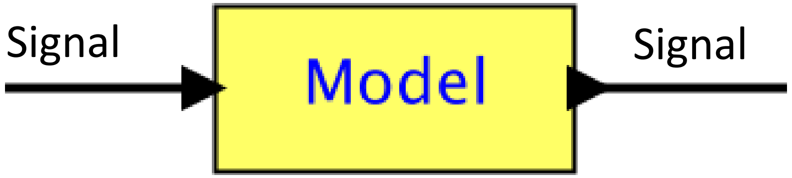
Box, G. E. P. and N. R. Draper, 1987: *Empirical Model-Building and Response Surfaces*. Wiley Series in Probability and Statistics, Wiley.

Deterministic Models for the Physical Side of CPS

Physical System

Image: Wikimedia Commons

Model

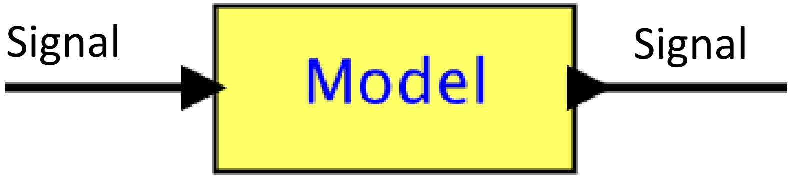


$$\dot{\mathbf{x}}(t) = \dot{\mathbf{x}}(0) + \frac{1}{M}\int_0^t \mathbf{F}(\tau)d\tau$$

Differential Equations are deterministic models

A Major Problem for CPS: Combinations of Deterministic Models are Nondeterministic

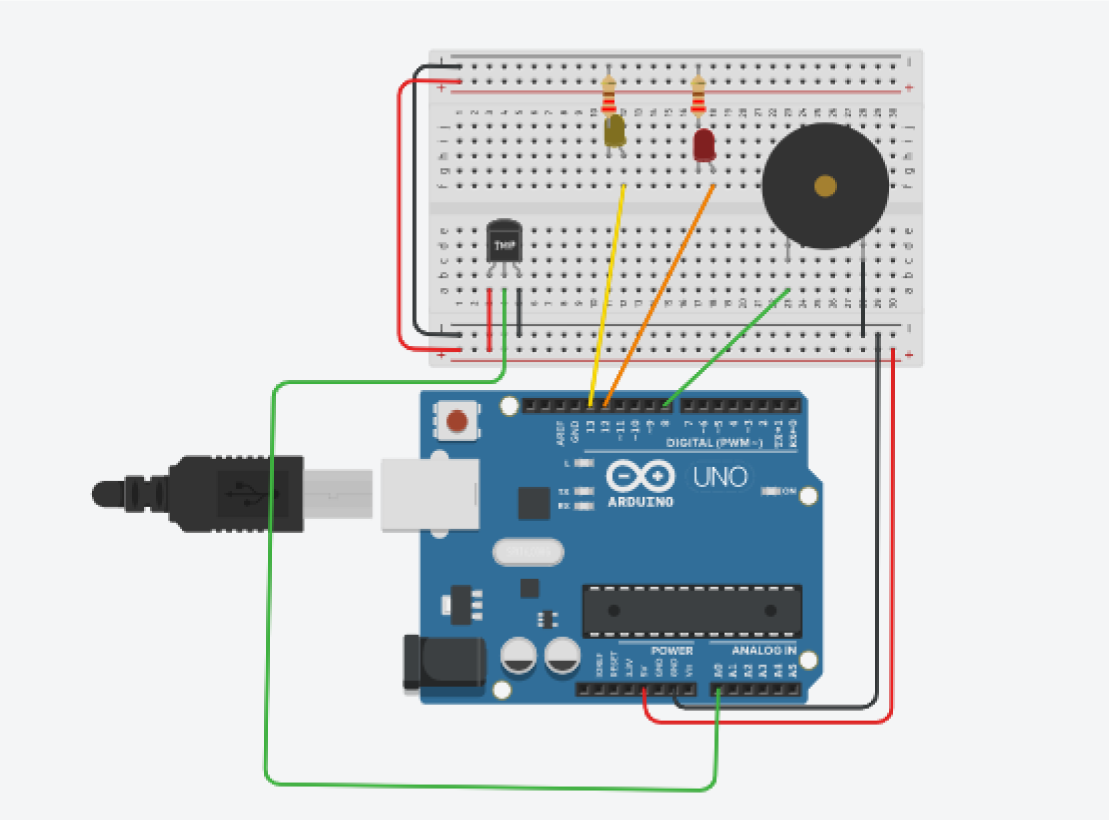
```
void initTimer(void) {
    SysTickPeriodSet(SysCtlClockGet() / 1000);
    SysTickEnable();
    SysTickIntEnable();
}
volatile uint timer_count = 0;
void ISR(void) {
    if(timer_count != 0) {
        timer_count--;
    }
}
int main(void) {
    SysTickIntRegister(&ISR);
    .. // other init
    timer_count = 2000;
    initTimer();
    while(timer_count != 0) {
        ... code to run for 2 seconds
    }
    ... // other code
}
```



$$\dot{\mathbf{x}}(t) = \dot{\mathbf{x}}(0) + \frac{1}{M}\int_0^t \mathbf{F}(\tau)d\tau$$

Image: Wikimedia Commons

Model Build on Logic



Determinism? Really?

CPS applications operate in an intrinsically nondeterministic world.

Does it really make sense to insist on deterministic models?

The Value of Models

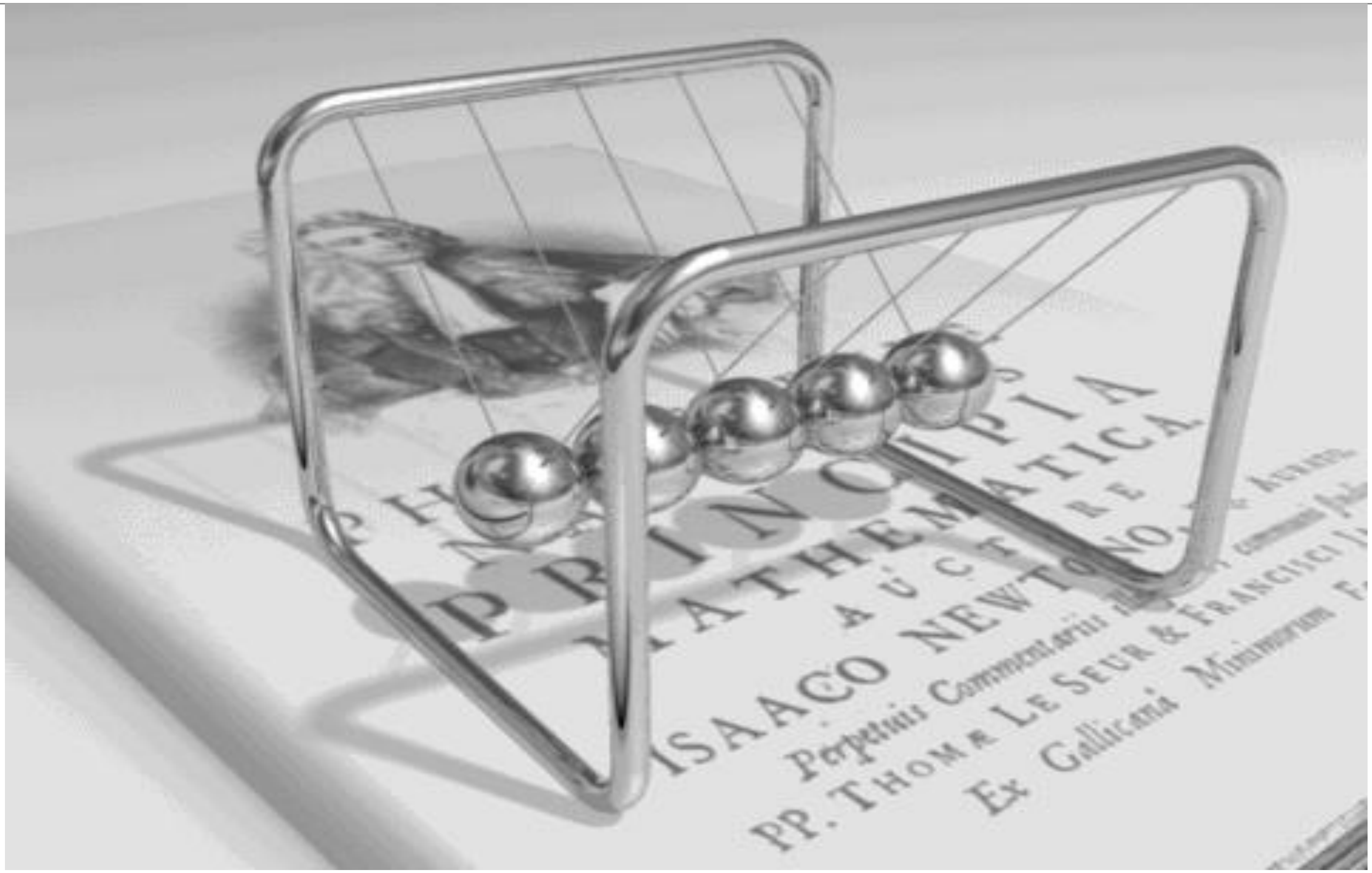
In *science*, the value of a *model* lies in how well its behavior matches that of the physical system.

In *engineering*, the value of the *physical system* lies in how well its behavior matches that of the model.

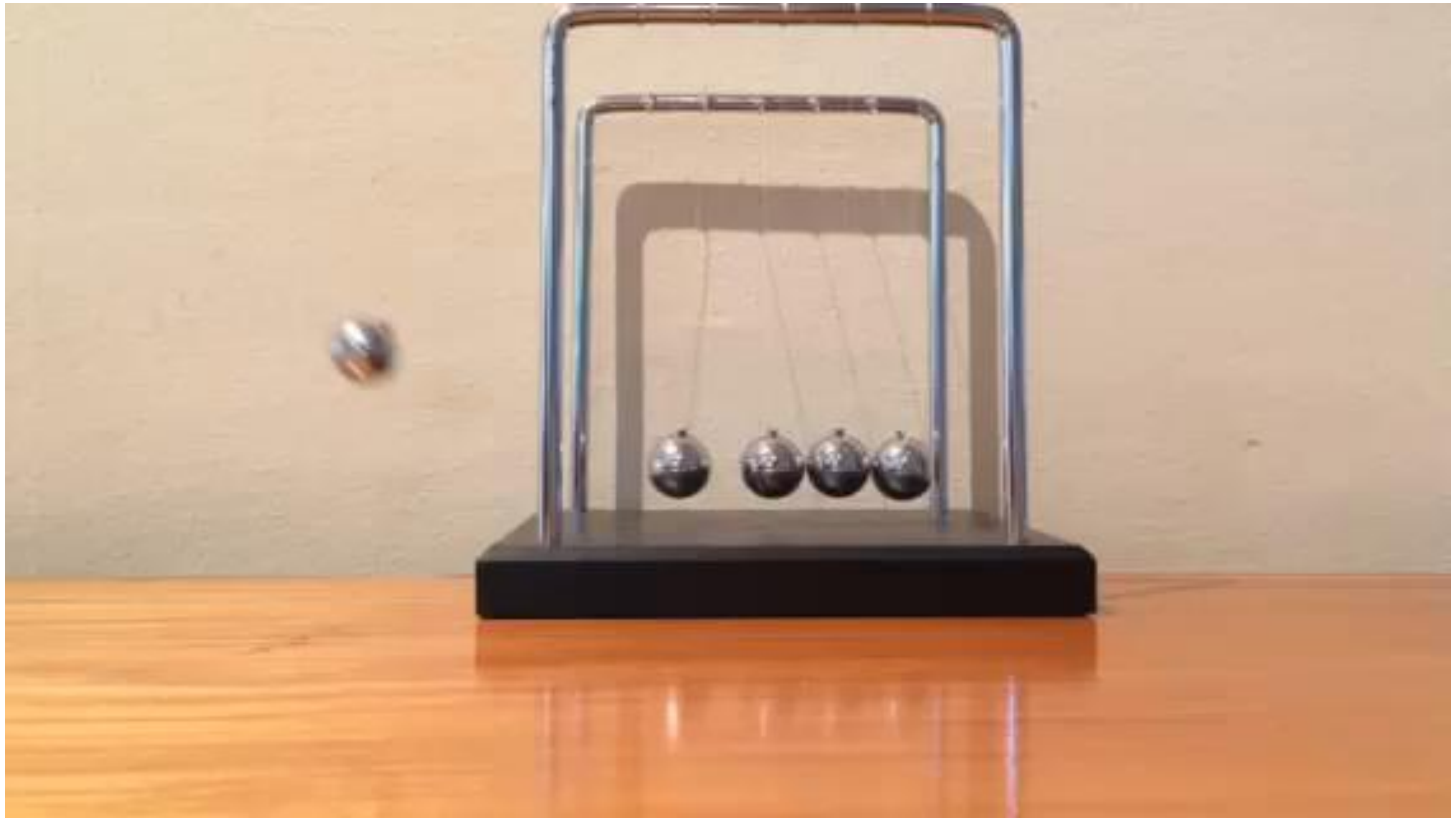
In engineering, model fidelity is a two-way street!

For a model to be useful, it is necessary (but not sufficient) to be able to be able to construct a faithful physical realization.

A Model


Perpetuis Commentariis
PP. THOMÆ LE SEUR & FRANCISCI
Ex Gallicana Minimorum

A Physical Realization



Model Fidelity

To a *scientist*, the model is flawed.

To an *engineer*, the realization is flawed.

I'm an engineer…

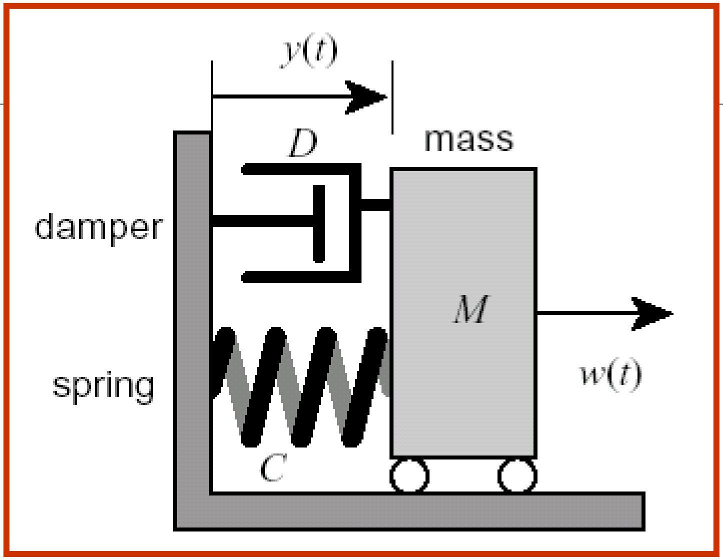
Determinism?
What about Resilience? Adaptability?

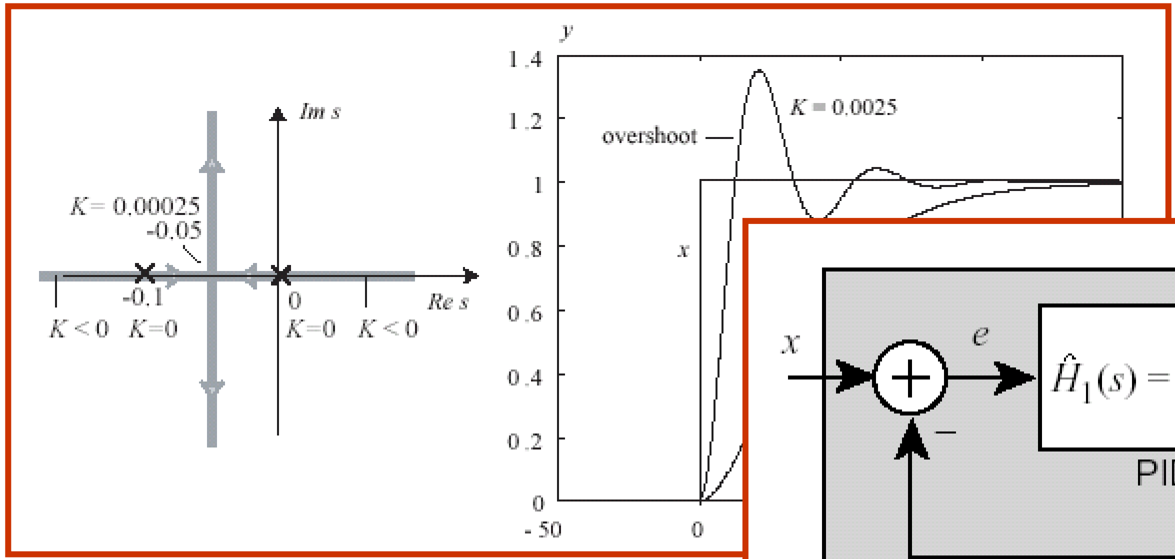
Deterministic models do not eliminate the need for robust, fault-tolerant designs.

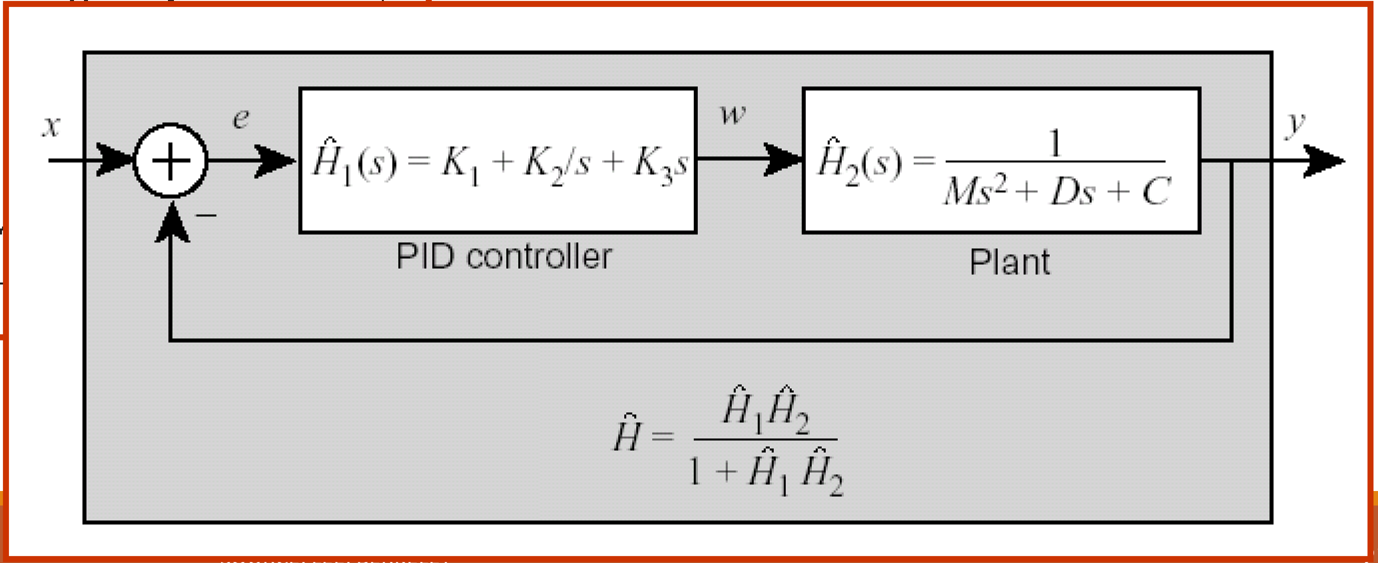
In fact, they *enable* such designs, because they make it much clearer what it means to have a fault!

Today's Lecture: Modeling of Continuous Dynamics

Ordinary differential equations, Laplace transforms, feedback control models, …







Mathematical Model for Physical System

Systems: Physical, Chemical, Biological, Economic or Social $\rightarrow$ Mathematical Formulas (Laws, Equations, etc.)

In CPS design, mechanics, electromagnetics, optics, and thermodynamics are important.

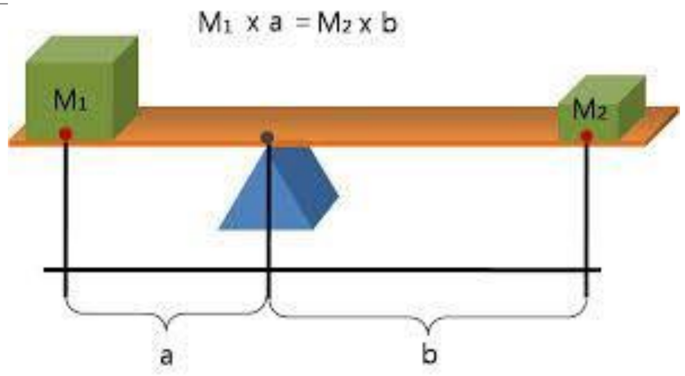
Conservation Laws for physical modeling.

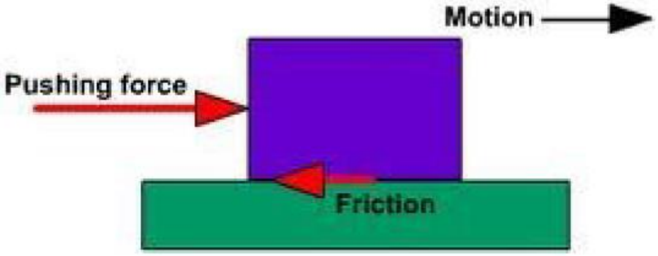
- Conservation of energy
- Conservation of momentum (translational and rotational)
- Conservation of mass
- Conservation of current
- Etc.

Khan Academy has a great collection on [basic physics](https://www.khanacademy.org/science/physics)
https://www.khanacademy.org/science/physics

Elements in Mechanical Systems

Lever





Mass: The basic rule for a mass is $F = m * a$, where F is force, m is mass, and a is acceleration.

Force (including gravity) : an element that can represent the basic mechanical interaction between two objects.

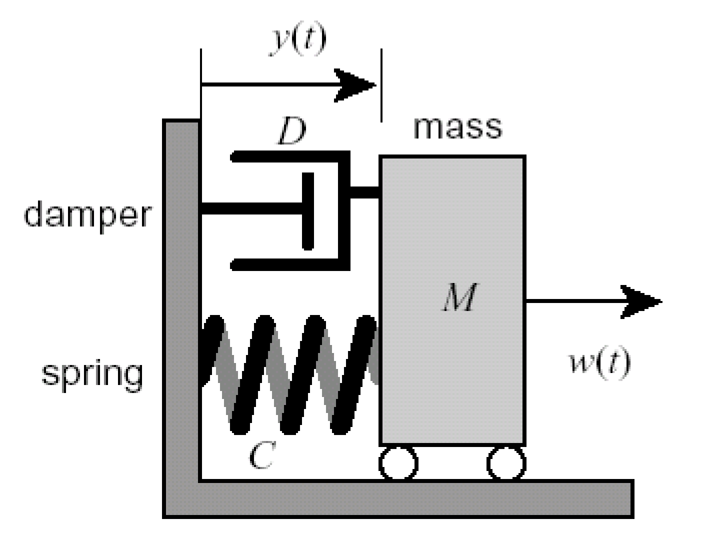
Friction: a phenomena that generates a force that opposes the direction of the movement, and is usually proportional to a formal force that is normal to the direction of the movement.

Momentum: the quantity of motion that object has (mass in motion). $L = m * v$.

Elements in Mechanical Systems

Spring : The spring is a component that is modeled by Hooke's law, which is usually represented by the equation , where is a force, is a displacement $F = -k * x$ F x relative to a neutral (natural position) for the spring, and k is a (scalar) coefficient that relates these two values.

Damper : A damper is a device that creates a force against the direction of movement, and in proportion to the speed of the movement (compare this to friction mentioned above). The rule for a damper therefore has the form $F = -k * v$, where F is a force, k is a constant, and v is a velocity.



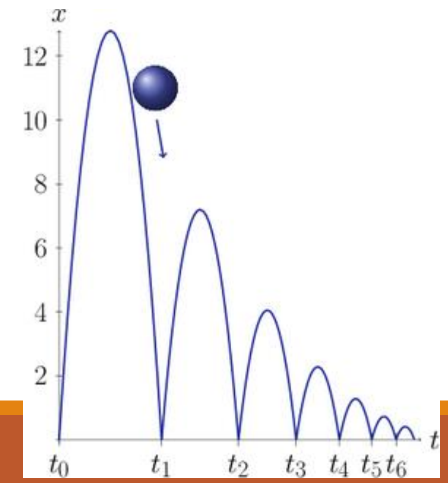
A **statics** problem is one where the system components are not moving, and nothing internal to the system can cause them to move.

Dynamics of Physical Systems

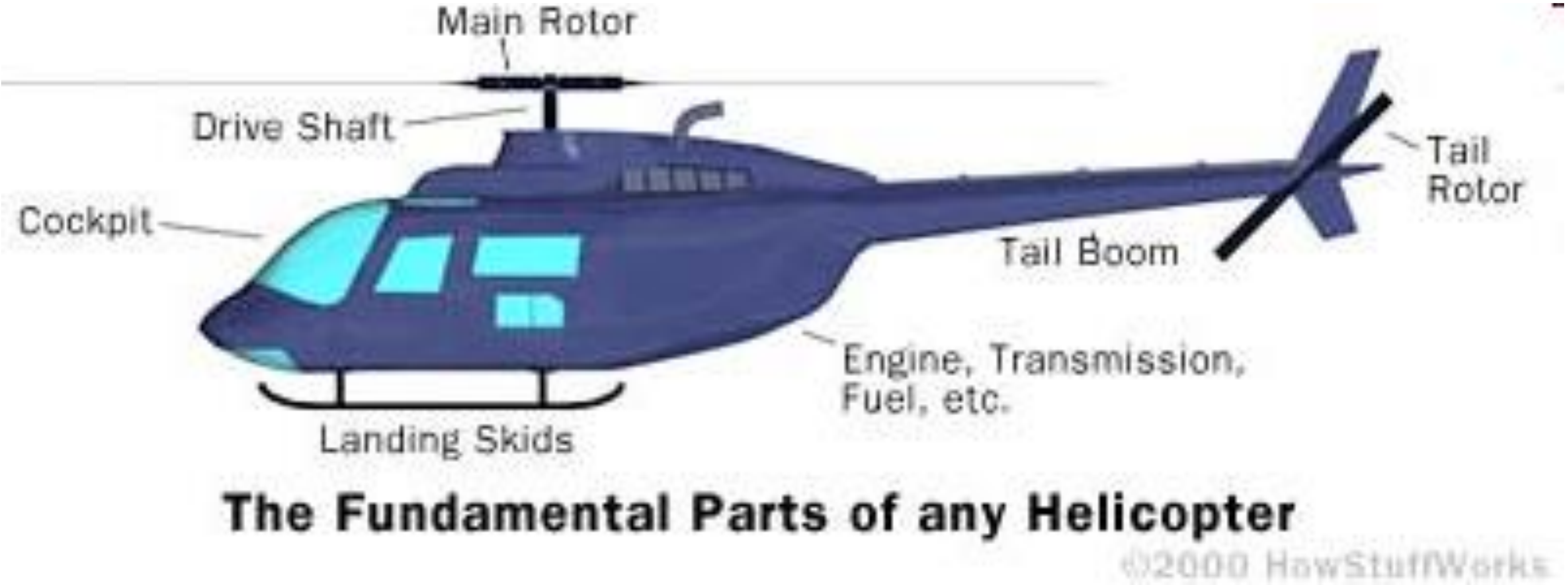
The study of mechanical systems where time is not involved is called "statics".

The same system can become a **dynamics** system if we are told that at least one part may be moving.

For example, the mass m (at position x) may be moving, in which case it would have speed x' and acceleration x'' .



An Example: Helicopter Dynamics



Degrees of Freedom

In physics, the **degrees of freedom** (**DOF**) of a mechanical system is the number of independent parameters that define its configuration or state.

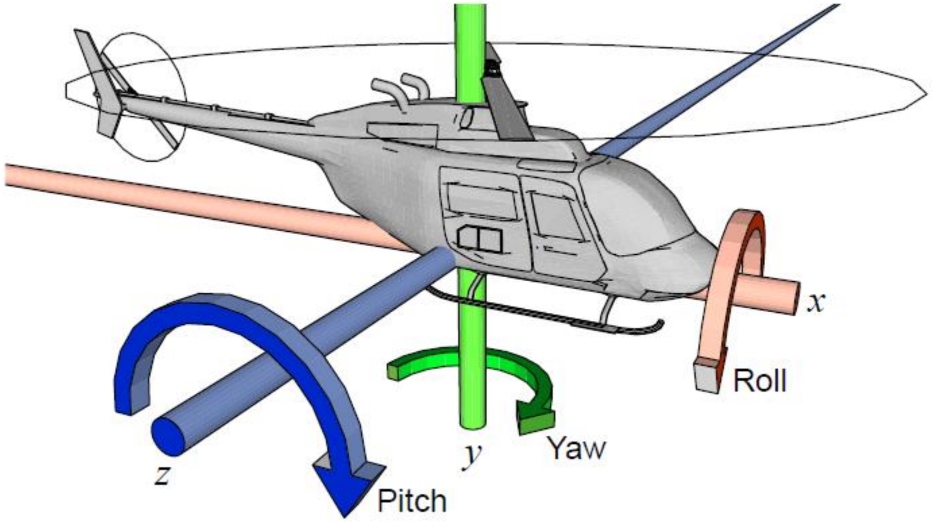
The position of a single railcar (engine) moving along a track has one degree of freedom.

The automobile with highly stiff suspension travelling on a plane has three degree of freedom (x, y and θ), but in reality six degree of freedom.

Modeling Physical Motion

Six degrees of freedom:

- Position: x, y, z
- Orientation: pitch, yaw, roll



- By convention x is drawn increasing to the right, y is drawn increasing upwards, and z is drawn increasing out of the page.
- Roll x is an angle of rotation around the x axis, where an angle of 0 radians represents horizontally flat along the z axis (i.e., the angle is given relative to the z axis).
- Yaw y is the rotation around the y axis, where 0 radians represents pointing directly to the right (i.e., the angle is given relative to the x axis).
- Pitch z is rotation around the z axis, where 0 radians represents pointing horizontally (i.e., the angle is given relative to the x axis).

Notation

Position is given by three functions:

$$x(t) \Rightarrow x : \mathbb{R} \to \mathbb{R}$$

$$y(t) \Rightarrow y : \mathbb{R} \to \mathbb{R}$$

$$z(t) \Rightarrow z : \mathbb{R} \to \mathbb{R}$$

Functions of this form are known as continuous-time signals.

where the domain $\mathbb{R}$ represents time and the co-domain (range) $\mathbb{R}$ represents position along the axis. Collecting into a vector:

$$\mathbf{x} : \mathbb{R} \to \mathbb{R}^3$$

Position at time $t \in \mathbb{R}$ is $\mathbf{x}(t) \in \mathbb{R}^3$.

Velocity and Acceleration

Velocity

$$\dot{\mathbf{x}} \colon \mathbb{R} \to \mathbb{R}^3$$

is the derivative, $\forall\, t \in \mathbb{R}$,

$$\dot{\mathbf{x}}(t) = \frac{d}{dt}\mathbf{x}(t)$$

Acceleration $\ddot{\mathbf{x}} \colon \mathbb{R} \to \mathbb{R}^3$ is the second derivative,

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

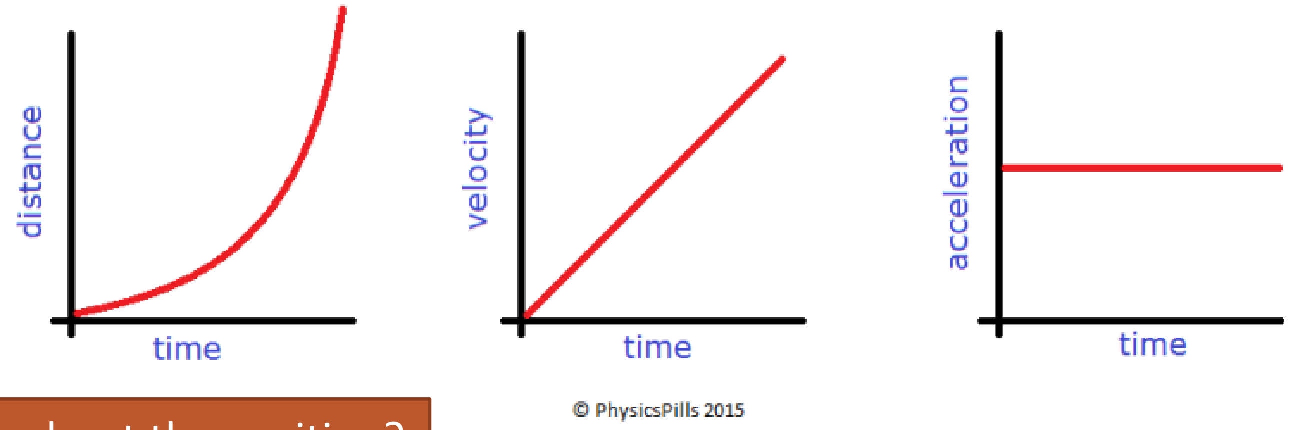
Force on an object is $\mathbf{F} \colon \mathbb{R} \to \mathbb{R}^3$.

Velocity and Acceleration

Velocity is the integral of acceleration, given by

$$\forall\, t > 0, \quad \dot{\mathbf{x}}(t) \;=\; \dot{\mathbf{x}}(0) + \int_0^t \ddot{\mathbf{x}}(\tau)\, d\tau$$

where $\dot{\mathbf{x}}(0)$ is the initial velocity in three directions.



How about the position?

Examples

- An object moving with a constant velocity $\dot{x} = 1$

- An object moving with a constant cancelation $\ddot{x} =$ -9.8

- An object moving with a velocity which increases linearly with its distance. $\dot{x} = x$

- $\ddot{x} = -x$

Newton's Second Law

Newton's second law states $\forall\, t \in \mathbb{R}$,

$$\mathbf{F}(t) = M\ddot{\mathbf{x}}(t)$$

Force vs Acceleration

where M is the mass. To account for initial position and velocity, convert this to an integral equation

$$\begin{aligned}\mathbf{x}(t) &= \mathbf{x}(0) + \int_0^t \dot{\mathbf{x}}(\tau)d\tau \\ &= \mathbf{x}(0) + t\dot{\mathbf{x}}(0) + \frac{1}{M}\int_0^t \int_0^\tau \mathbf{F}(\alpha)d\alpha d\tau,\end{aligned}$$

If you know the initial position and initial velocity of an object and the forces on the object in all three directions as a function of time, you can determine the acceleration, velocity, and position of the object at any time.

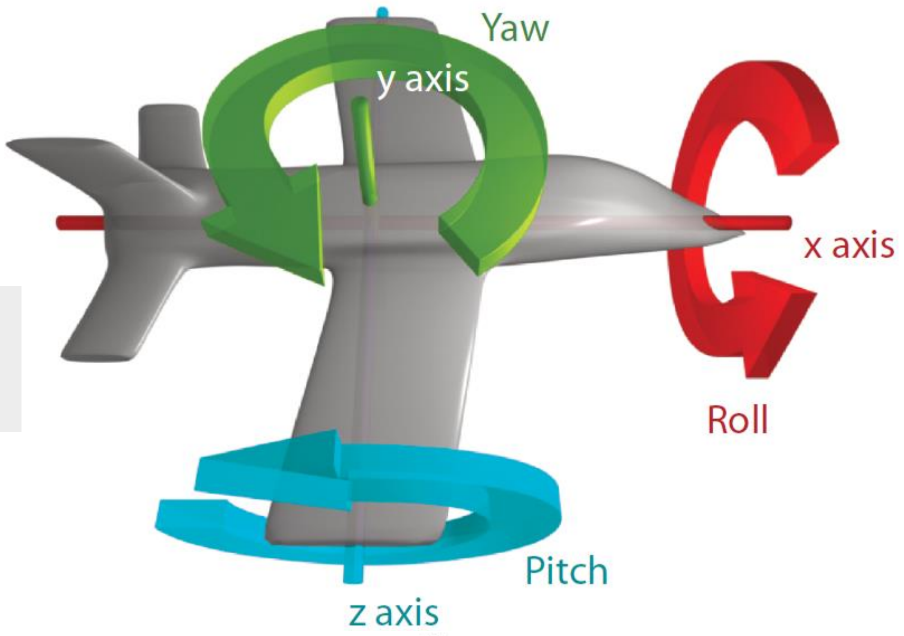
Orientation

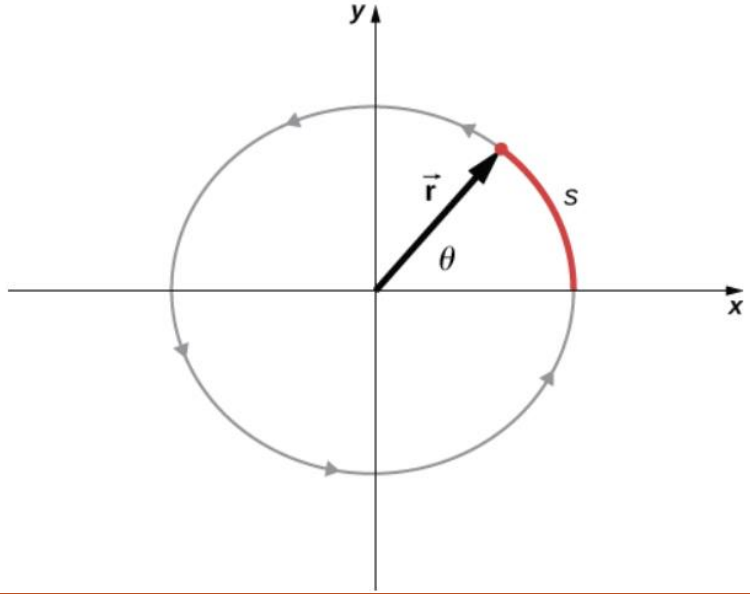
- Orientation: $\theta : \mathbb{R} \rightarrow \mathbb{R}^3$

Angular position

- Angular velocity: $\dot{\theta} : \mathbb{R} \rightarrow \mathbb{R}^3$ $\omega = \lim_{\Delta t \to 0} \frac{\Delta \theta}{\Delta t} = \frac{d\theta}{dt},$
- Angular acceleration: $\ddot{\theta} : \mathbb{R} \rightarrow \mathbb{R}^3$ $\frac{d\omega}{dt} = \frac{d^2\theta}{dt^2},$
- Torque: $\mathbf{T} : \mathbb{R} \rightarrow \mathbb{R}^3$

$$\theta(t) = \begin{bmatrix} \dot{\theta}_x(t) \\ \dot{\theta}_y(t) \\ \dot{\theta}_z(t) \end{bmatrix} = \begin{bmatrix} \text{roll} \\ \text{yaw} \\ \text{pitch} \end{bmatrix}$$

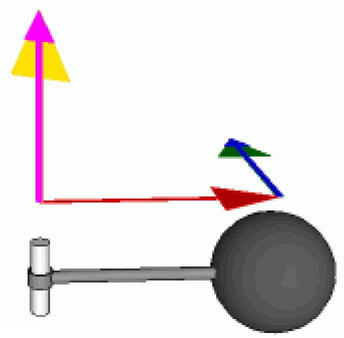




The versions of these equations of motion that affect orientation use **torque**, the rotational version of force.

Angular version of force is torque.
For a point mass rotating around a fixed axis:

- radius of the arm: $r \in \mathbb{R}$
- force orthogonal to arm: $f \in \mathbb{R}$
- mass of the object: $m \in \mathbb{R}$



$$T_y(t) = r f(t)$$

angular momentum, momentum

Just as force is a push or a pull, a torque is a twist.

Units: newton-meters/radian, Joules/radian

Note that radians are meters/meter (2π meters of circumference per 1 meter of radius), so as units, are optional.

Rotational Version of Newton's Second Law

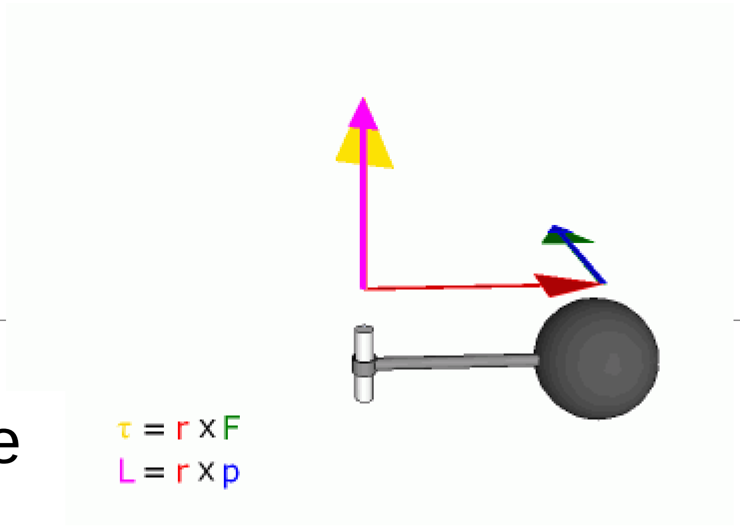
$$\mathbf{T}(t) = \frac{d}{dt}\left(I(t)\dot{\theta}(t)\right),$$

where $I(t)$ is a 3×3 matrix called the moment of inertia tensor.

$$\begin{bmatrix} T_x(t) \\ T_y(t) \\ T_z(t) \end{bmatrix} = \frac{d}{dt}\left(\begin{bmatrix} I_{xx}(t) & I_{xy}(t) & I_{xz}(t) \\ I_{yx}(t) & I_{yy}(t) & I_{yz}(t) \\ I_{zx}(t) & I_{zy}(t) & I_{zz}(t) \end{bmatrix}\begin{bmatrix} \dot{\theta}_x(t) \\ \dot{\theta}_y(t) \\ \dot{\theta}_z(t) \end{bmatrix}\right)$$

Here, for example, $T_y(t)$ is the net torque around the y axis (which would cause changes in yaw), $I_{yx}(t)$ is the inertia that determines how acceleration around the x axis is related to torque around the y axis.

Feedback Control Problem



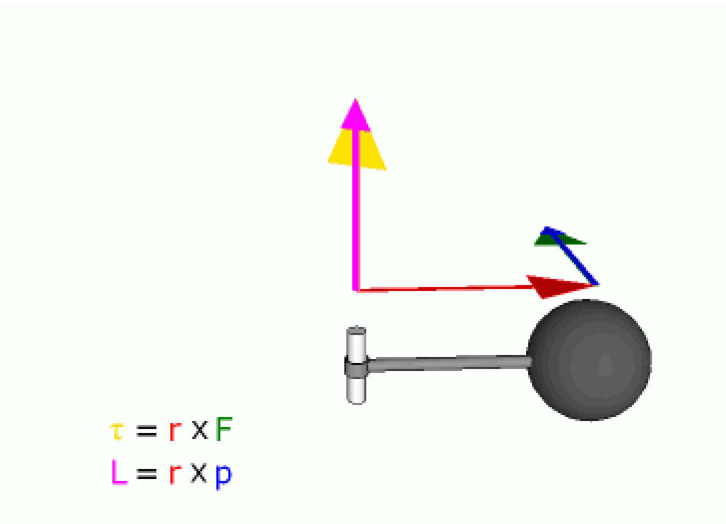
A helicopter without a tail rotor, like the one below, will spin uncontrollably due to the torque induced by friction in the rotor shaft.

Control system problem:
Apply torque using the tail rotor to counterbalance the torque of the top rotor.

Simplified Model

We assume that the helicopter position is fixed at the origin, so there is no need to consider equations describing position and the helicopter remains vertical, so pitch and roll are fixed at zero.

With these assumptions, the moment of inertia reduces to a scalar that represents a torque that resists changes in yaw.



Yaw dynamics:

$$T_y(t) = I_{yy}\ddot{\theta}_y(t)$$

The sum of the torque caused by the main rotor and that caused by the tail rotor.

To account for initial angular velocity, write as

$$\dot{\theta}_y(t) = \dot{\theta}_y(0) + \frac{1}{I_{yy}}\int_0^t T_y(\tau)d\tau.$$

Simplified Model

$$\begin{aligned}\theta(t) &= \theta(0) + \int_0^t \dot{\theta}(\tau)d\tau \\ &= \theta(0) + t\dot{\theta}(0) + \frac{1}{I}\int_0^t \int_0^\tau \mathbf{T}(\alpha)d\alpha d\tau\end{aligned}$$

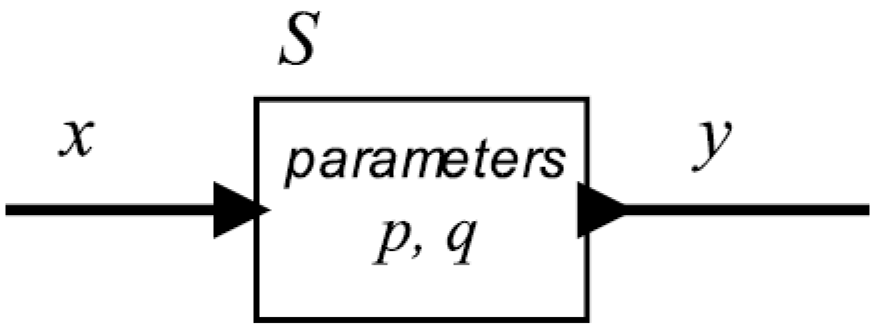
If you know the initial orientation and initial rotational velocity of an object and the torques on the object in all three axes as a function of time, you can determine the rotational acceleration, velocity, and orientation of the object at any time.

Actor Model of Systems

- A continuous-time system (one that operates on continuous-time signals) may be modeled by a box with an input port and an output port (Actor model).
- Actor represents a unit of computation
- Actors can
  - ✓ Create more actors
  - ✓ Send messages to other actors
  - ✓ Designate what to do with the next message
- Multiple actors may execute at the same time.

Actor Model of Systems

A *system* is a function that accepts an input *signal* and yields an output signal.



$$x : \mathbb{R} \to \mathbb{R}, \quad y : \mathbb{R} \to \mathbb{R}$$

The domain and range of the system function are sets of signals, which themselves are functions.

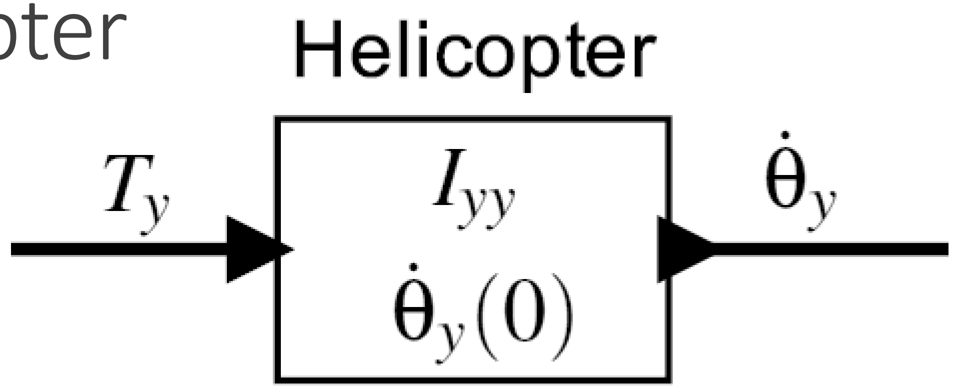
$$S : X \to Y$$

$$X = Y = (\mathbb{R} \to \mathbb{R})$$

Parameters may affect the definition of the function *S*.

Actor Model of the Helicopter

Input is the net torque of the tail rotor and the top rotor. Output is the angular velocity around the *y* axis.

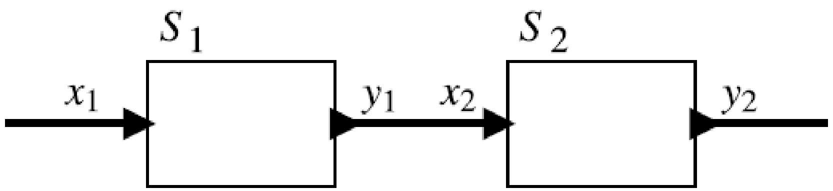


Parameters of the model are shown in the box. The input and output relation is given by the equation to the right.

$$\dot{\theta}_y(t) = \dot{\theta}_y(0) + \frac{1}{I_{yy}} \int_0^t T_y(\tau) d\tau$$

Composition of Actor Models

Actor models are composable.

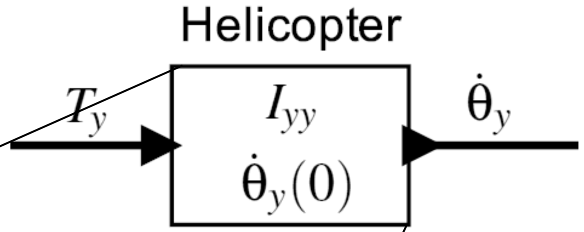


cascade composition

the "wire" between the output of S_1 and the input of S_2 means precisely that $y_1 = x_2$, or more pedantically

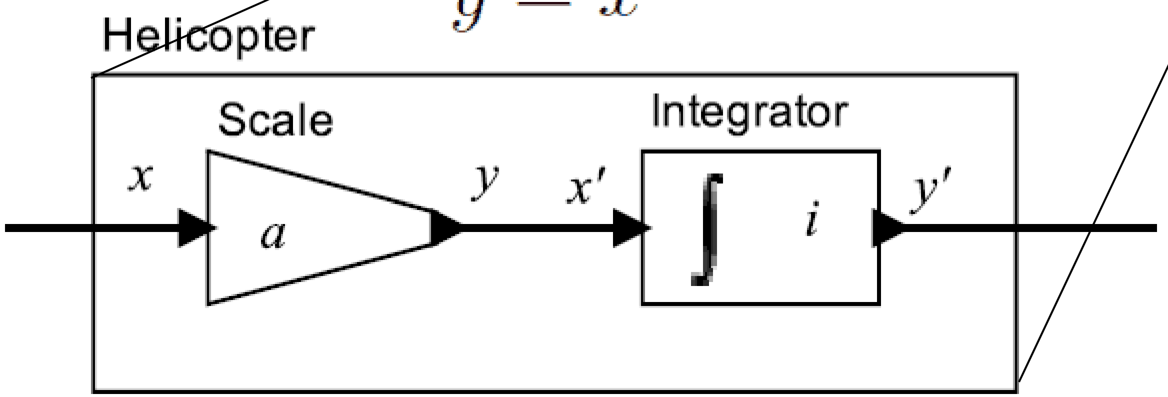
$$\forall\, t \in \mathbb{R}, \quad y_1(t) = x_2(t).$$

Composition of Actor Models



$$y = x'$$

$$x = T_y$$



$$y' = \dot{\theta}_y$$

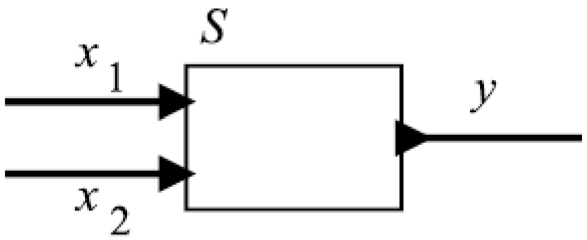
$$\forall\, t \in \mathbb{R}, \quad y(t) = ax(t) \qquad y'(t) = i + \int_0^t x'(\tau)\,d\tau$$

$$y = ax$$

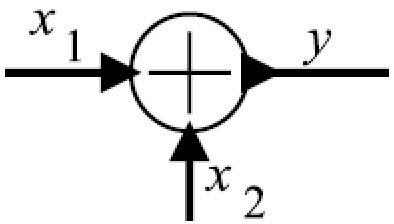
$$a = 1/I_{yy}$$

$$i = \dot{\theta}_y(0)$$

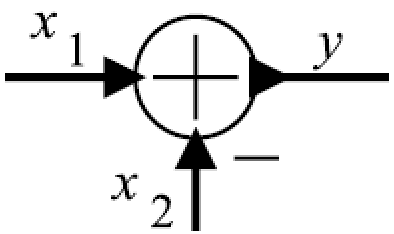
Actor Models with Multiple Inputs



$$S\colon (\mathbb{R} \to \mathbb{R})^2 \to (\mathbb{R} \to \mathbb{R})$$

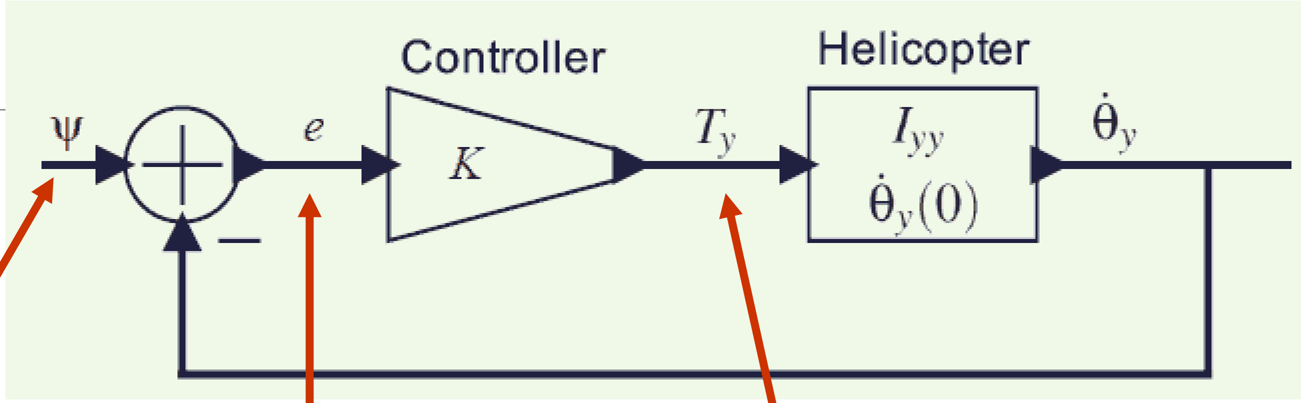


$$\forall\, t \in \mathbb{R}, \quad y(t) = x_1(t) + x_2(t)$$



$$(S(x_1, x_2))(t) = y(t) = x_1(t) - x_2(t)$$

Proportional controller



desired angular velocity

error signal

net torque

$$e(t) = \psi(t) - \dot{\theta}_y(t) \qquad T_y(t) = Ke(t)$$

$$\begin{aligned}\dot{\theta}_y(t) &= \dot{\theta}_y(0) + \frac{1}{I_{yy}} \int_0^t T_y(\tau)d\tau \\ &= \dot{\theta}_y(0) + \frac{K}{I_{yy}} \int_0^t (\psi(\tau) - \dot{\theta}_y(\tau))d\tau\end{aligned}$$

Note that the angular velocity appears on both sides, so this equation is not trivial to solve.

Simulation Model

- Providing a well-motivated opportunity to be exposed to mathematical modeling
- Avoiding the need to have analytical solutions, which usually exist only for a smaller class of problems than what we can simulate
- Enabling many more virtual experiments to be run than would be possible with physical ones. Physical testing can be prohibitive for reasons including cost, safety, and controllability.
- Enabling easier measurement and evaluation than would be possible with physical experiments
- Providing an opportunity to learn an important skill in CPS design, namely, systematic experimentation
- Increasing the chances of producing a successful CPS design
- Producing many useful visualizations
- Facilitating the creation of animations and computer games

Simulation Model on Python

Login your KMTIL Gmail account

https://colab.research.google.com/

https://apmonitor.com/pdc/index.php/Main/SolveDifferentialEquations

Properties of Systems

- Causal Systems
- Memoryless Systems
- Linearity and Time Invariance
- Stability

Causal Systems

A system is causal if its output depends only on current and past inputs.

restriction in time $x|_{t \leq \tau}(t) = x(t)$

If x is an input to a system, then $x|_{t\leq\tau}$ is the "current and past inputs" at time τ.

$$x_1|_{t\leq\tau} = x_2|_{t\leq\tau} \Rightarrow S(x_1)|_{t\leq\tau} = S(x_2)|_{t\leq\tau}$$

the system is causal if for two possible inputs x_1 and x_2 that are identical up to (and including) time τ, the outputs are identical up to (and including) time τ.

Memoryless Systems

A system has memory if the output depends not only on the current inputs, but also on past inputs (or future inputs, if the system is not causal).

$$(S(x))(t) = f(x(t))$$

the output $(S(x))(t)$ at time t depends only on the input $x(t)$ at time t.

The Integrator considered above is not memoryless, but the adder is.

Linearity and Time Invariance

Systems that are linear and time invariant (LTI) have particularly nice mathematical properties.

Much of the theory of control systems depends on these properties.

A system $S : X \to Y$, where X and Y are sets of signals, is linear if it satisfies the **superposition** property:

$$\forall\, x_1, x_2 \in X \text{ and } \forall\, a, b \in \mathbb{R}, \quad S(ax_1 + bx_2) = aS(x_1) + bS(x_2).$$

A system $S : X \to Y$ is time invariant if

$$\forall\, x \in X \text{ and } \forall\, \tau \in \mathbb{R}, \quad S(D_\tau(x)) = D_\tau(S(x)). \qquad (D_\tau(x))(t) = x(t - \tau)$$

A **linear time-invariant system** (LTI) is a system that is both linear and time invariant.

Stability

A system is said to be bounded-input bounded-output stable (BIBO stable or just stable) if the output signal is bounded for all input signals that are bounded.

Recommended Reading

- Khan Academy has a great collection on basic physics
- To dig deeper, consult as needed these basic technical references on differentiation and integration, and on resistive circuit elements, analysis of resistive circuits, and RLC circuit analysis.
- Sections 6.1 and 6.2 of this draft of David Morin's book on Classical Mechanics.
- For fun, you may enjoy checking out this great illustration on the scale of the universe.
- Close et al.'s textbook on modeling and analysis of dynamic systems.
- The Ethicist: A Heating Problem (Short article)
- Article about 17 Equations that Changed the World
- An example of a model that won a Nobel Prize: the Hodgkin-Huxkley model.
- Check out Wolfram's online Alpha tool.
- Readers interested in more advanced methods for modeling mechanical systems (beyond the scope of this course) may wish to consult the Wikipedia article on the Euler-Lagrange equation.
- Much modeling and simulation aims at predicting the behavior of systems. NY Review has an interesting article on three books on prediction.

Reference

- Lee, Edward & Seshia, Sanjit. (2011). Introduction to Embedded Systems - A Cyber-Physical Systems Approach.
- Lecture Note Slides from EECS 149/249A: Introduction to Embedded Systems (UC Berkeley) by Prof. Prabal Dutta and Sanjit A. Seshia
- *Lecture Notes on Cyber-Physical Systems, Walid Taha, 2018* bit.ly/LNCPS-2018
- *https://apmonitor.com/pdc/index.php/Main/FirstOrderSystems*