COMPUTER ENGINEERING

UIT
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

EMBEDDED SYSTEM DESIGN

Model Based Design

Doan Duy, Ph. D.

Email: duyd@uit.edu.vn

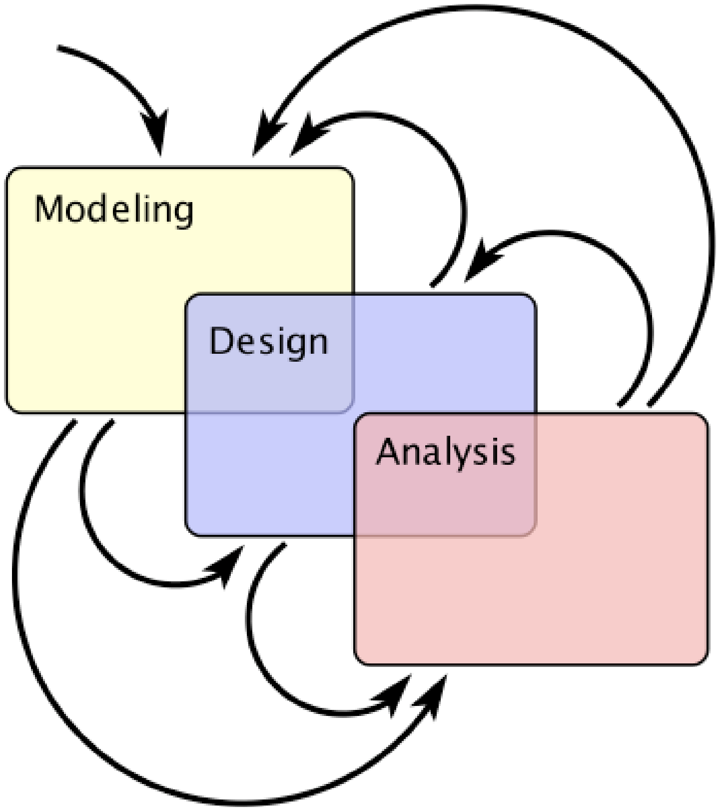
Objectives

- Understanding the concept of model
- Understanding problems of applying model to CPS
- Brief of Model of Continuous Dynamics

Contents

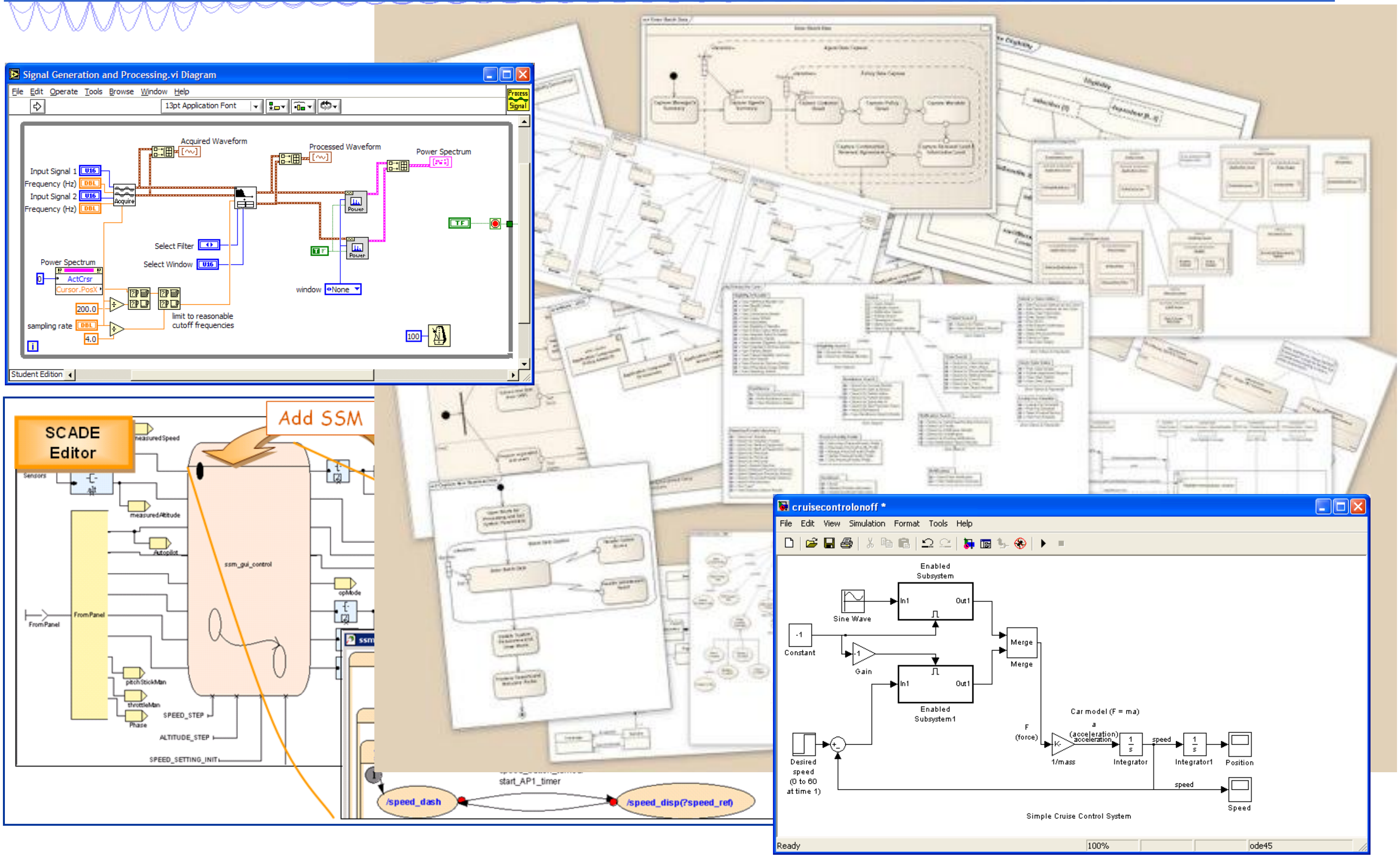
1. Introduction to Model

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).

Models



Signal Generation and Processing.vi Diagram
File Edit Operate Tools Browse Window Help
13pt Application Font
Process Signal
Acquired Waveform
Processed Waveform
Power Spectrum
Input Signal 1
Frequency (Hz)
Input Signal 2
Frequency (Hz)
Acquire
Power
Select Filter
Select Window
Power Spectrum
ActCrsr
Cursor.PosX
200.0
sampling rate
4.0
limit to reasonable cutoff frequencies
window
None
100
Student Edition
SCADE Editor
Add SSM
measuredSpeed
Sensors
measuredAltitude
Autopilot
ssm_gui_control
opMode
FromPanel
pitchStickMan
throttleMan
Phase
SPEED_STEP
ALTITUDE_STEP
SPEED_SETTING_INIT
start_AP1_timer
/speed_dash
/speed_disp(?speed_ref)
cruisecontrolonoff *
File Edit View Simulation Format Tools Help
Enabled Subsystem
Sine Wave
In1
Out1
Constant
Gain
Merge
Enabled Subsystem1
Car model (F = ma)
F (force)
a (acceleration)
1/mass
speed
Integrator
Integrator1
Position
Desired speed (0 to 60 at time 1)
Speed
Simple Cruise Control System
Ready
100%
ode45

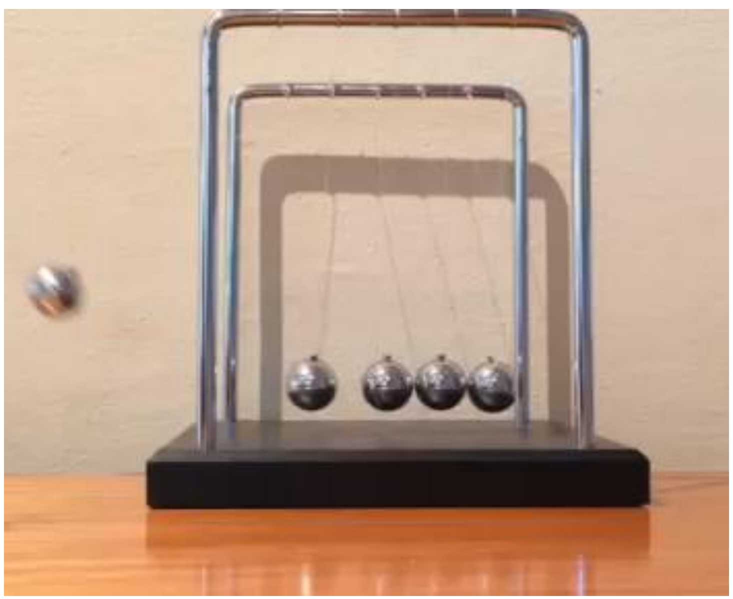
Models vs. Reality

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

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

The model: equation, function code, schematic

Fidelity is how well the model and its target match

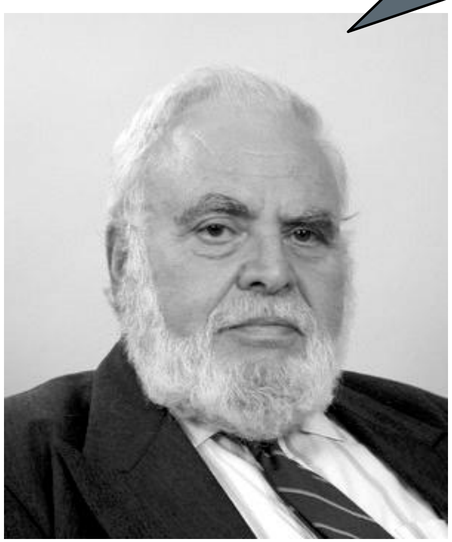


The target (the thing being modeled).

Why is model…

Engineers often confuse the model with its target

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!

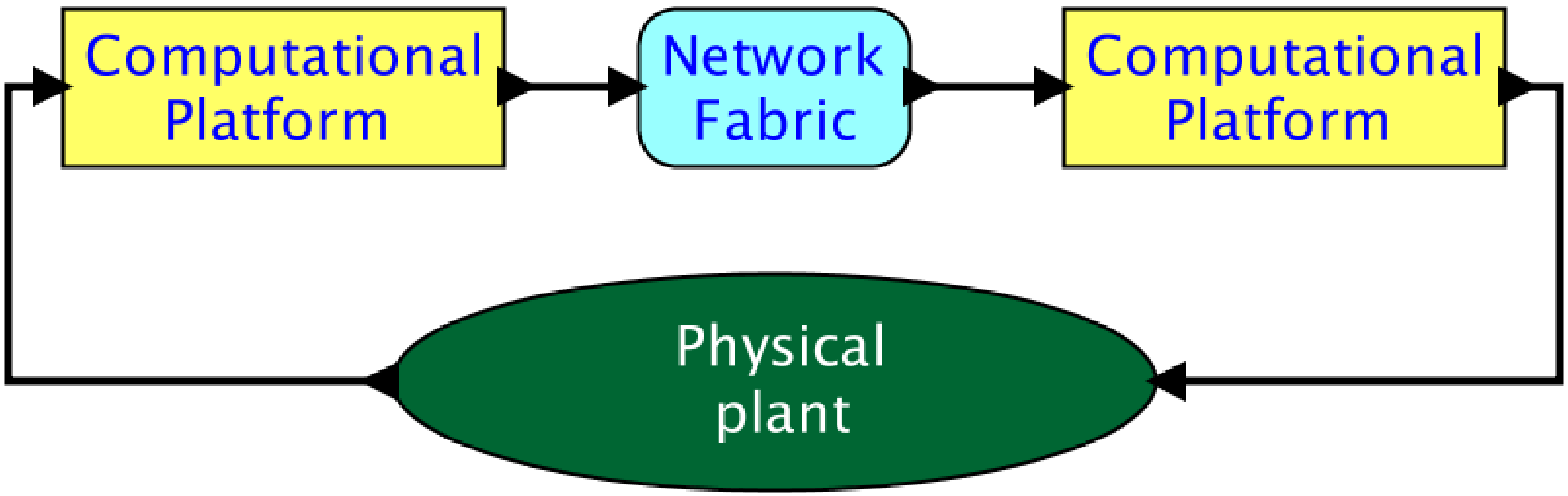
Determinacy

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.

Deterministic models have proven extremely valuable in the past.

A simple CPS



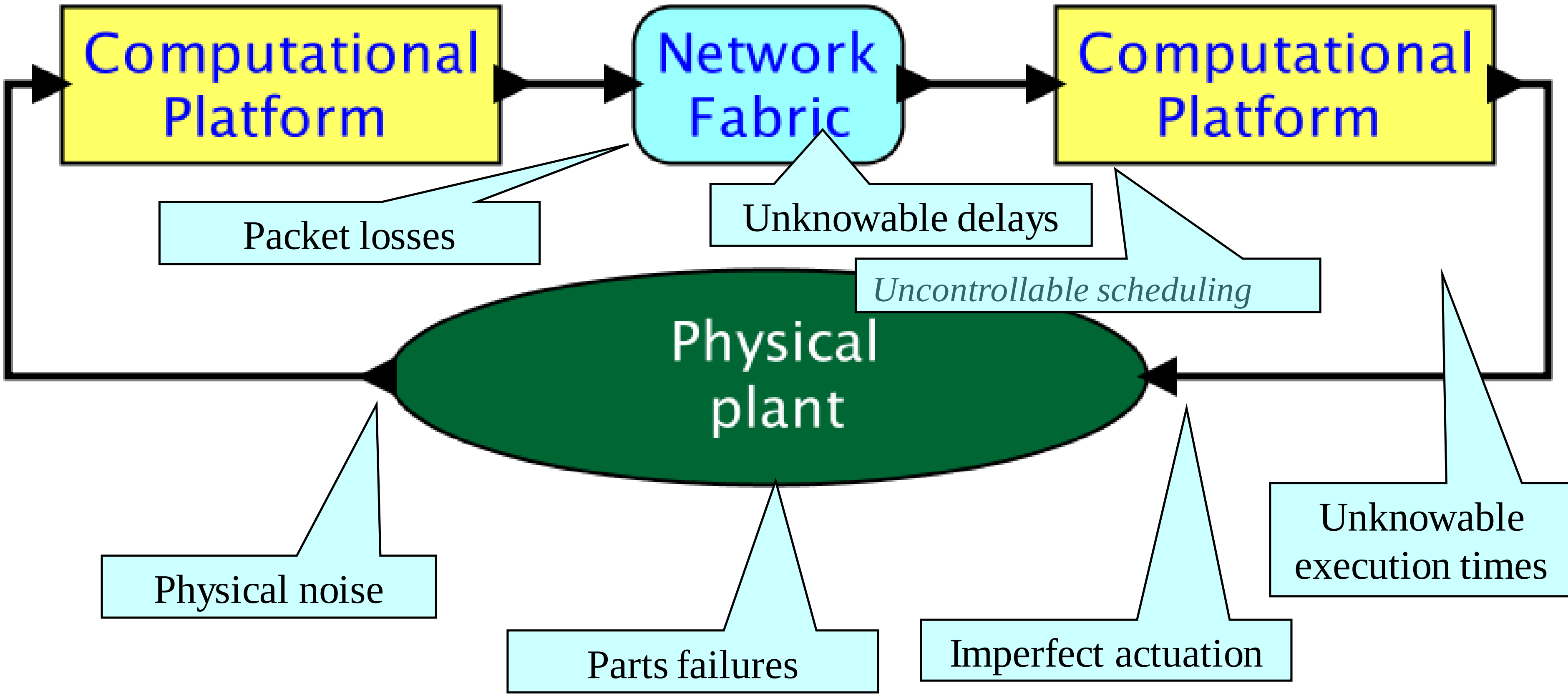
Computational Platform
Network Fabric
Computational Platform
Physical plant

COMPUTER ENGINEERING

2. Model Design for CPS

Model vs. CPS

Do deterministic models make sense for Cyber-physical systems?



Model need

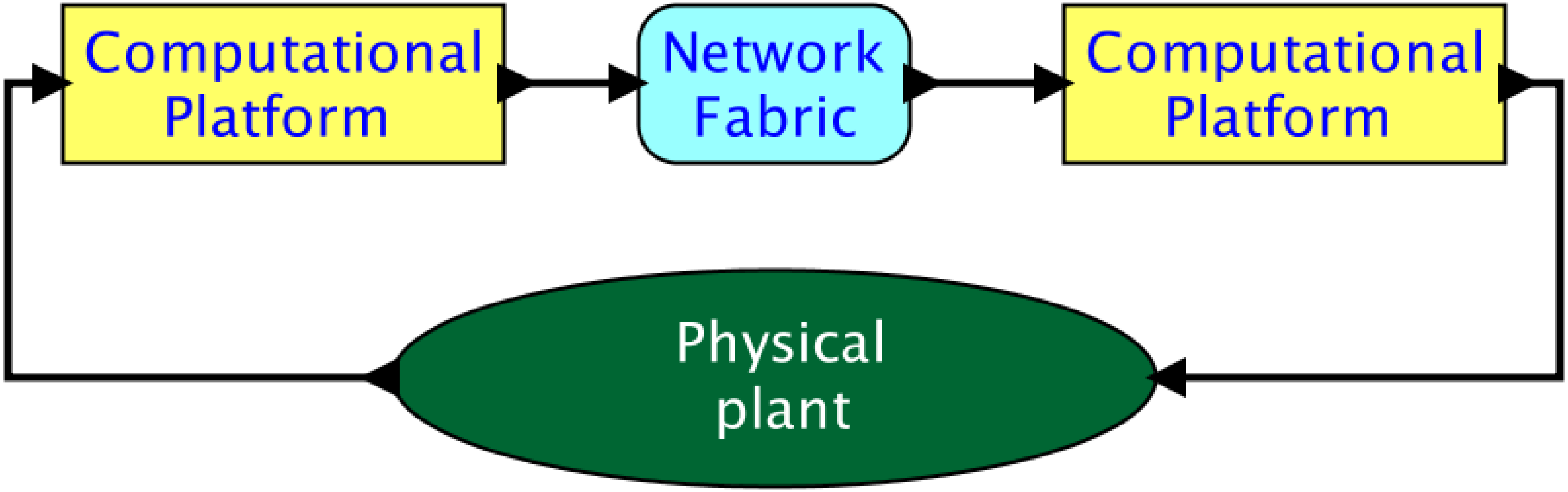
A Model Need not be *True* to be *Useful*

"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.

Kinds of models

What kinds of models should we use?



Software Models

Physical System

Model

```
void foo(int32_t x) {
    if (x > 1000) {
        x = 1000;
    }
    if (x > 0) {
        x = x + 1000;
        if (x < 0) {
            panic();
        }
    }
}
```

Single-threaded imperative programs are deterministic models

Hardware Abstracts

Instruction Set Architectures (ISAs) are deterministic models

Physical System

Model

Image: Wikimedia Commons

Integer Register-Register Operations

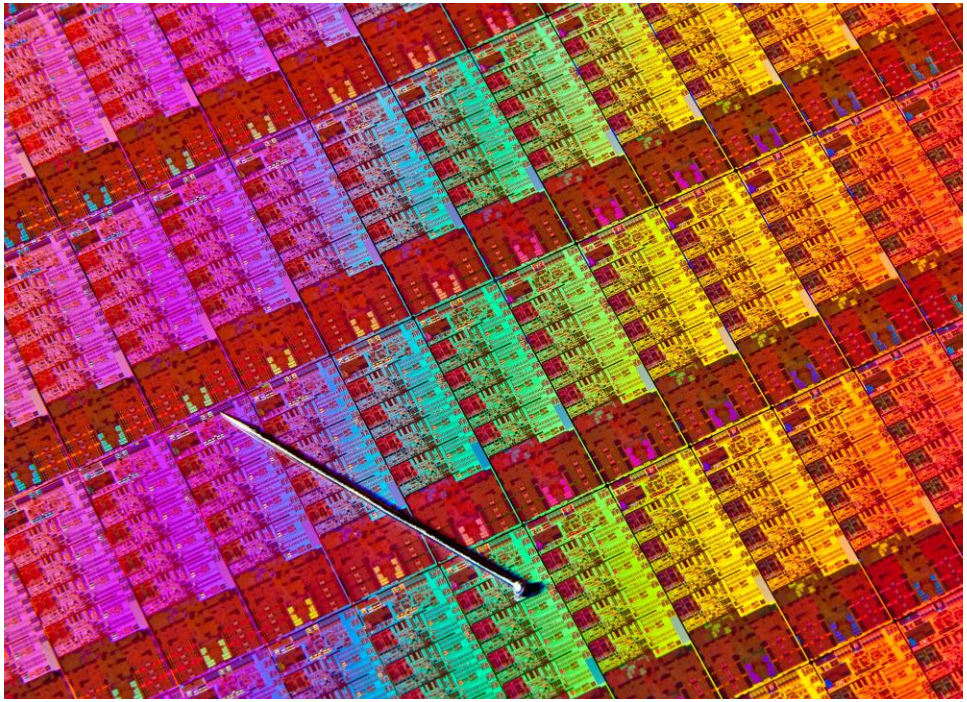
RISC-V defines several arithmetic R-type operations. All operations read the *rs1* and *rs2* registers as source operands and write the result into register *rd*. The *funct* field selects the type of operation.

31 27	26 22	21 17	16 7	6 0
rd	rs1	rs2	funct10	opcode
5	5	5	10	7
dest	src1	src2	ADD/SUB/SLT/SLTU	OP
dest	src1	src2	AND/OR/XOR	OP
dest	src1	src2	SLL/SRL/SRA	OP
dest	src1	src2	ADDW/SUBW	OP-32
dest	src1	src2	SLLW/SRLW/SRAW	OP-32

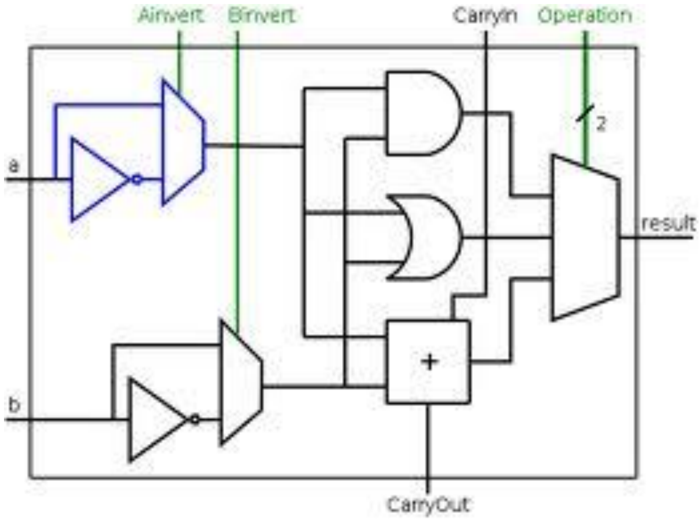
Waterman, et al., The RISC-V Instruction Set Manual, UCB/EECS-2011-62, 2011

Schematic models

Physical System



Model



Synchronous digital logic
is a deterministic model

Equation

Physical System

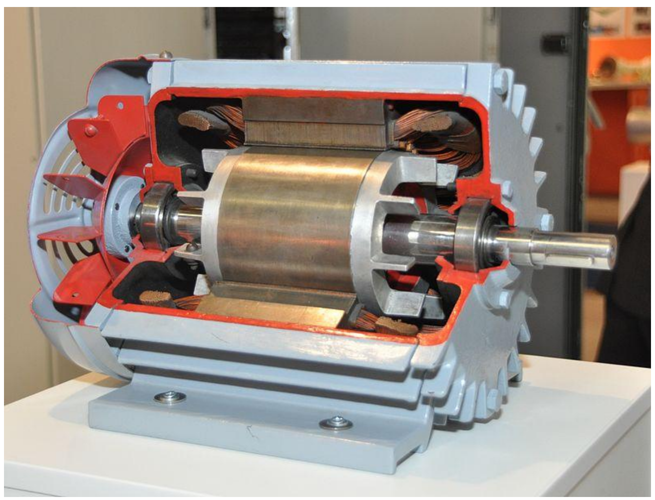
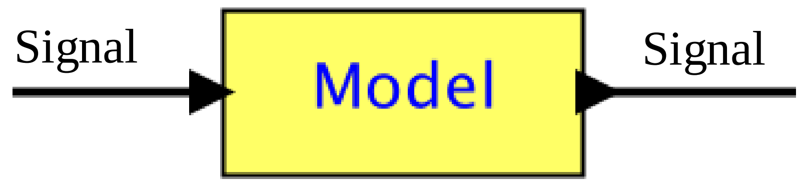


Image: Wikimedia Commons

Model



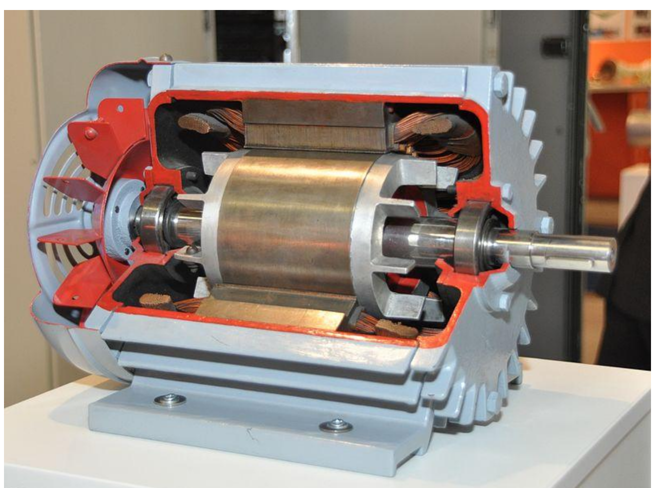
$$\dot{\mathbf{x}}(t) = \dot{\mathbf{x}}(0) + \frac{1}{M}\int_0^t \mathbf{F}(\tau)\,d\tau$$

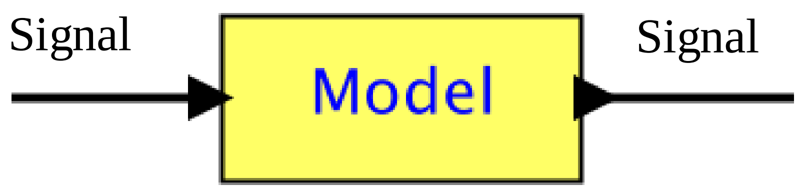
Differential Equations
are deterministic models

Problem of modelling 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
}
```





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

Example: Problem with WCET

A war story:

Ferdinand et al. [2001] determine the WCET of astonishingly simple avionics code from Airbus running on a Motorola ColdFire 5307, a pipelined CPU with a unified code and data cache. Despite the software consisting of a fixed set of non-interacting tasks containing only simple control structures, their solution required detailed modeling of the seven-stage pipeline and its precise interaction with the cache, generating a large integer linear programming problem.

Fundamentally, the ISA of the processor has failed to provide an adequate abstraction. And the problem has gotten worse since 2001!

COMPUTER ENGINEERING

Timing is not Part of Software and Network Semantics

***Correct execution** of a program in all widely used programming languages, and **correct delivery** of a network message in all general-purpose networks has nothing to do with how long it takes to do anything.*

Programmers have to step outside the programming abstractions to specify timing behavior.

Embedded software designers have no map!

Determinism? Really?

CPS applications operate in an intrinsically nondeterministic world.

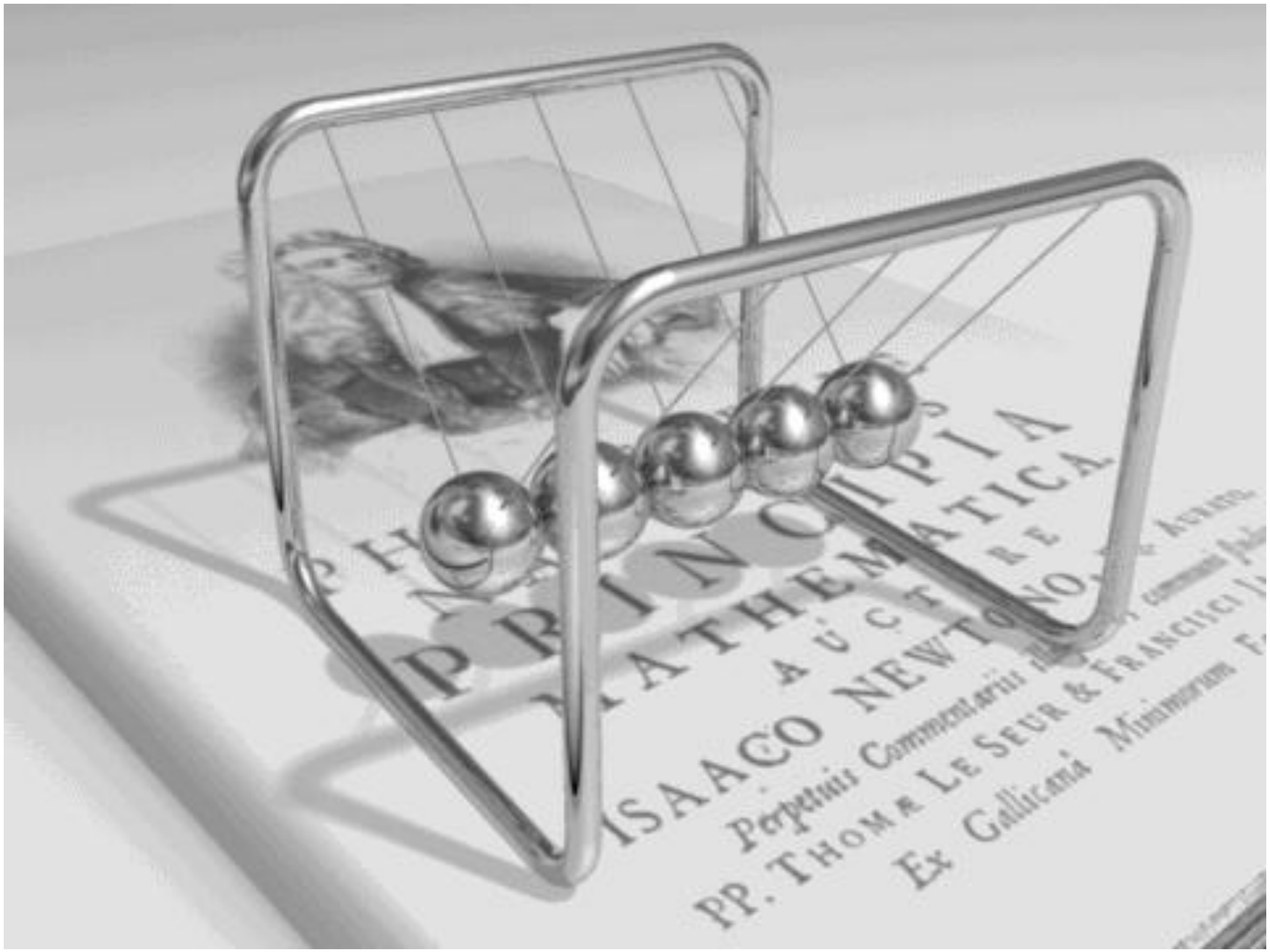
Does it really make sense to insist on deterministic models?

Value of Models

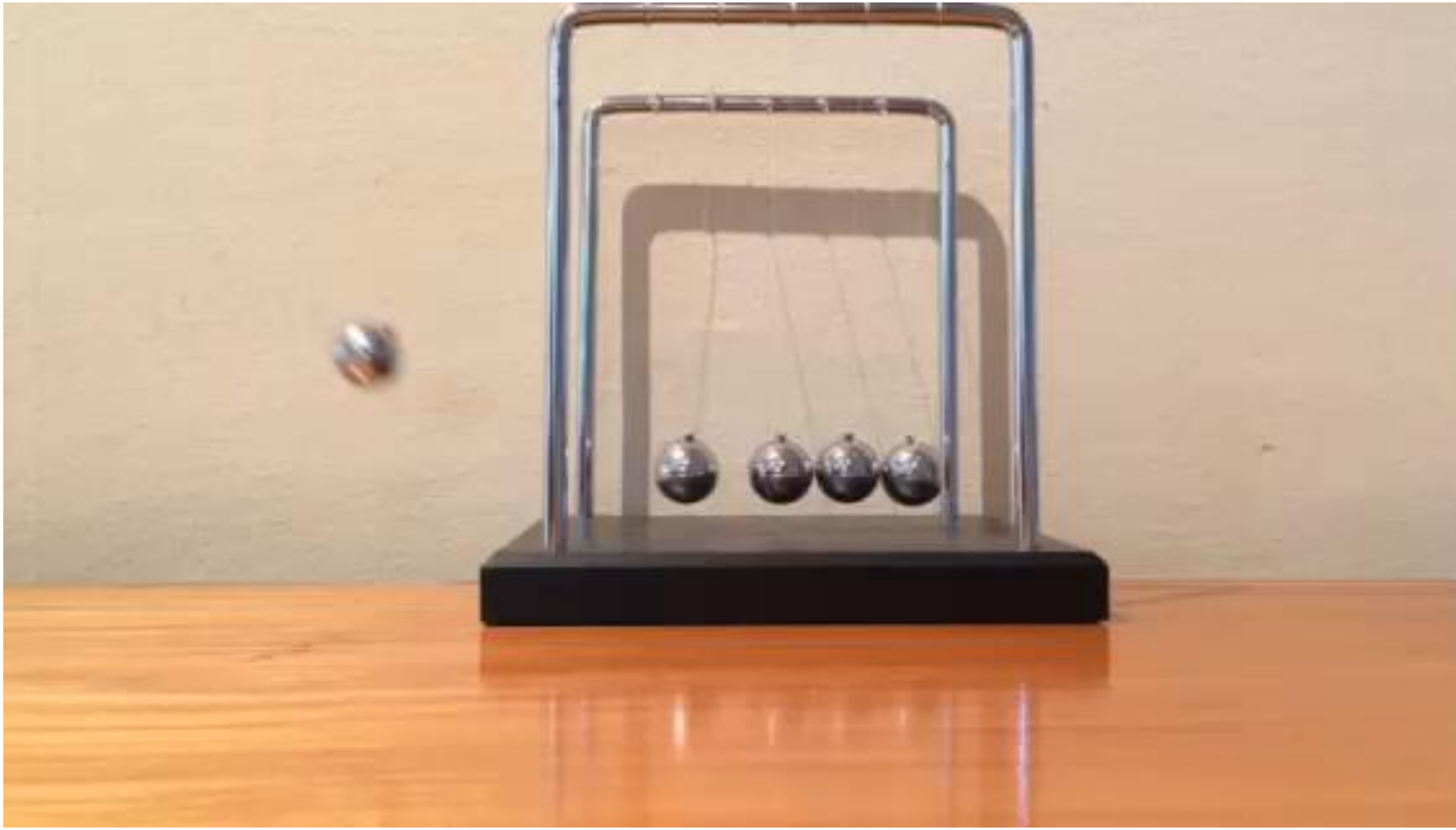
- In *science*, the value of a *model* lies in how well its behavior matches that of the physical system.
- In *engineering*, the value of the *physical system* lies in how well its behavior matches that of the model.

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

Model


PRINCIPIA
MATHEMATICA
AUCTORE
ISAACO NEWTONO
Perpetuis Commentariis
PP. THOMÆ LE SEUR & FRANCISCI
Ex Gallicanâ Minimorum

Physical Realization



Model Fidelity

- To a *scientist*, the model is flawed.
- To an *engineer*, the realization is flawed.

I'm an engineer…

Changing question…

The question is *not* whether deterministic models can describe the behavior of cyber-physical systems (with high fidelity).

The question is whether we can build cyber-physical systems whose behavior matches that of a deterministic model (with high probability).

COMPUTER ENGINEERING

Determinism vs. Resilience and 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!

We have to fix the models!

But how?

COMPUTER ENGINEERING

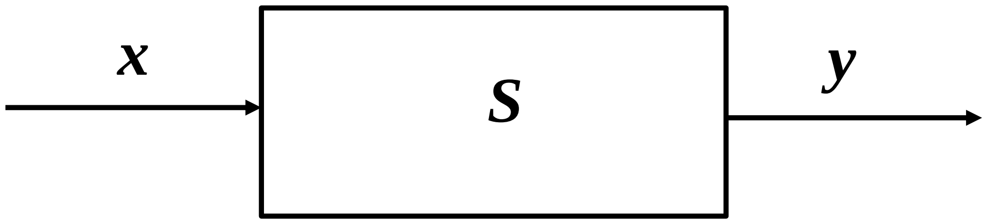
3. Models of Continuous Dynamic

Modeling Techniques

- Models that are abstractions of **system dynamics**
- (how system behavior changes over time)

- Modeling physical phenomena – differential equations
- Feedback control systems – time-domain modeling
- Modeling modal behavior – FSMs, hybrid automata, …
- Modeling sensors and actuators –calibration, noise, …
- Hardware and software – concurrency, timing, power, …
- Networks – latencies, error rates, packet losses, …

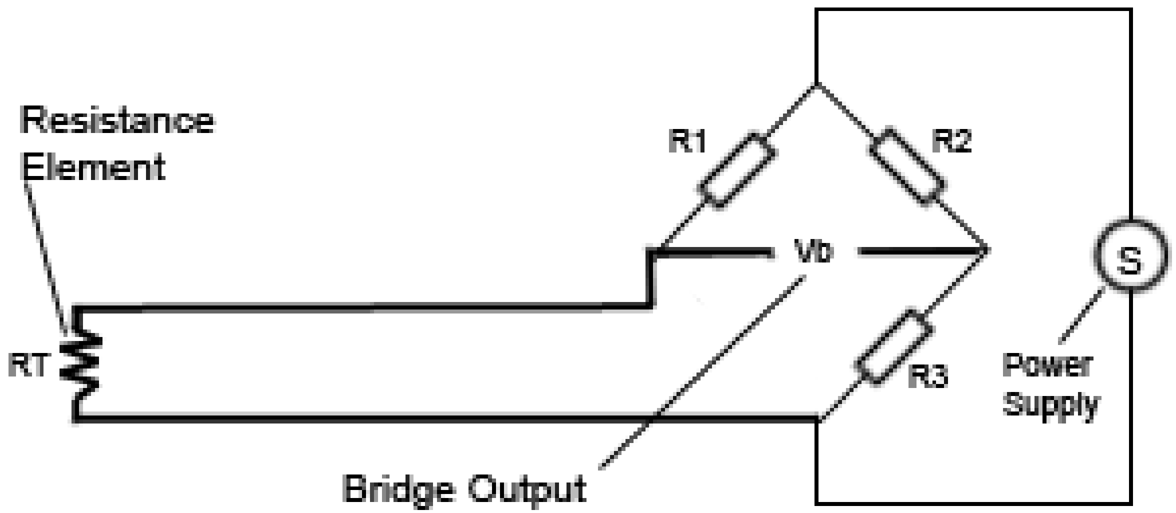
Actor Models with Single Input

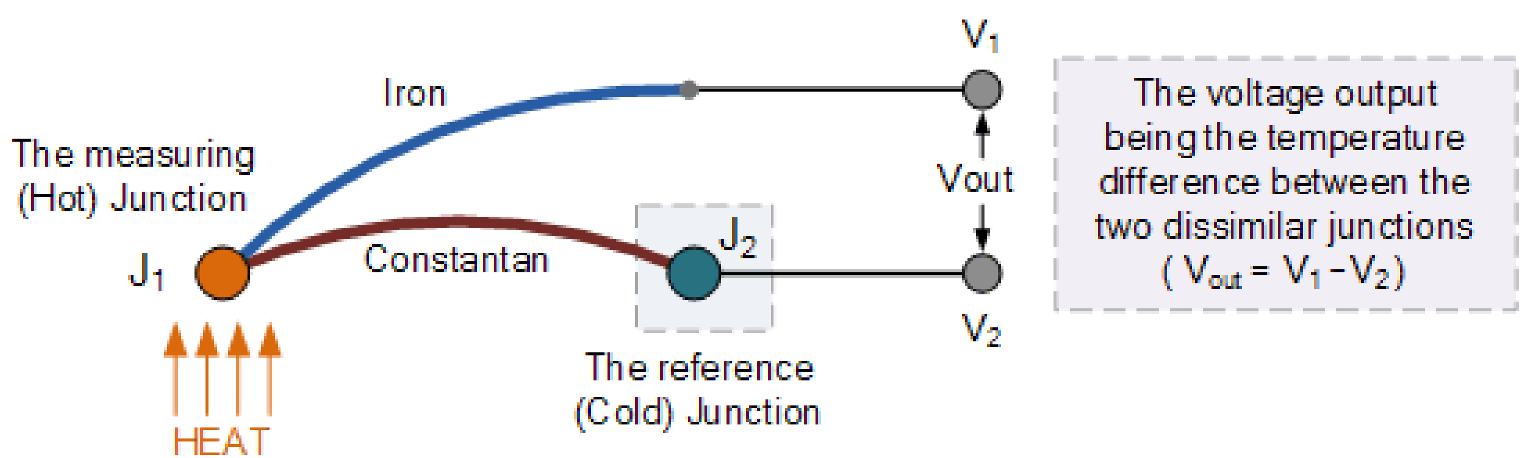


$$x: R \rightarrow R$$
$$y: R \rightarrow R$$

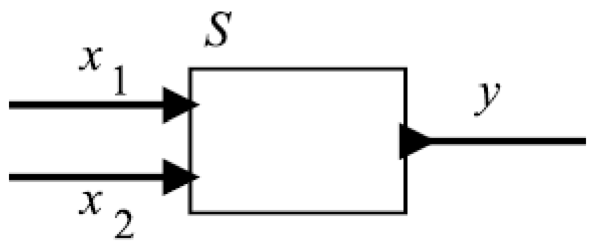
$$S: (R \rightarrow R) \rightarrow (R \rightarrow R)$$

Thermal Sensor

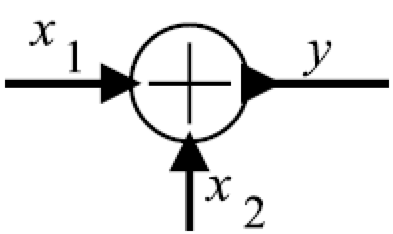




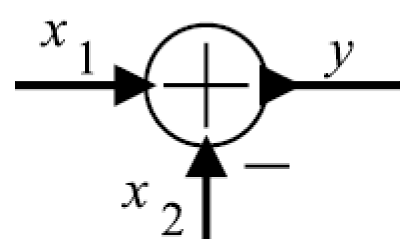
Actor Models with Multiple Inputs



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

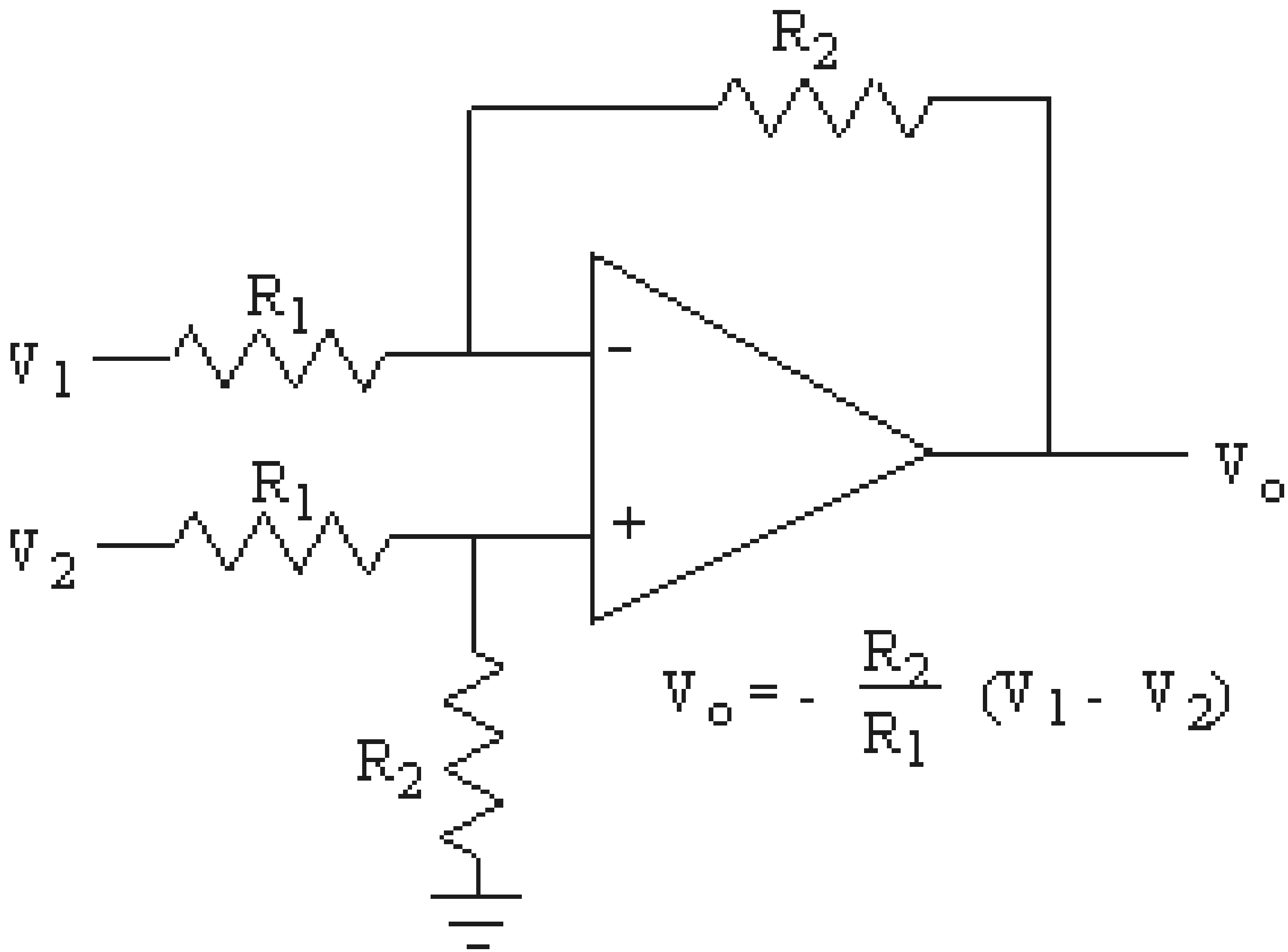


$$\forall\, t \in \mathbb{R}, \quad y(t) = x_1(t) + x_2(t)$$



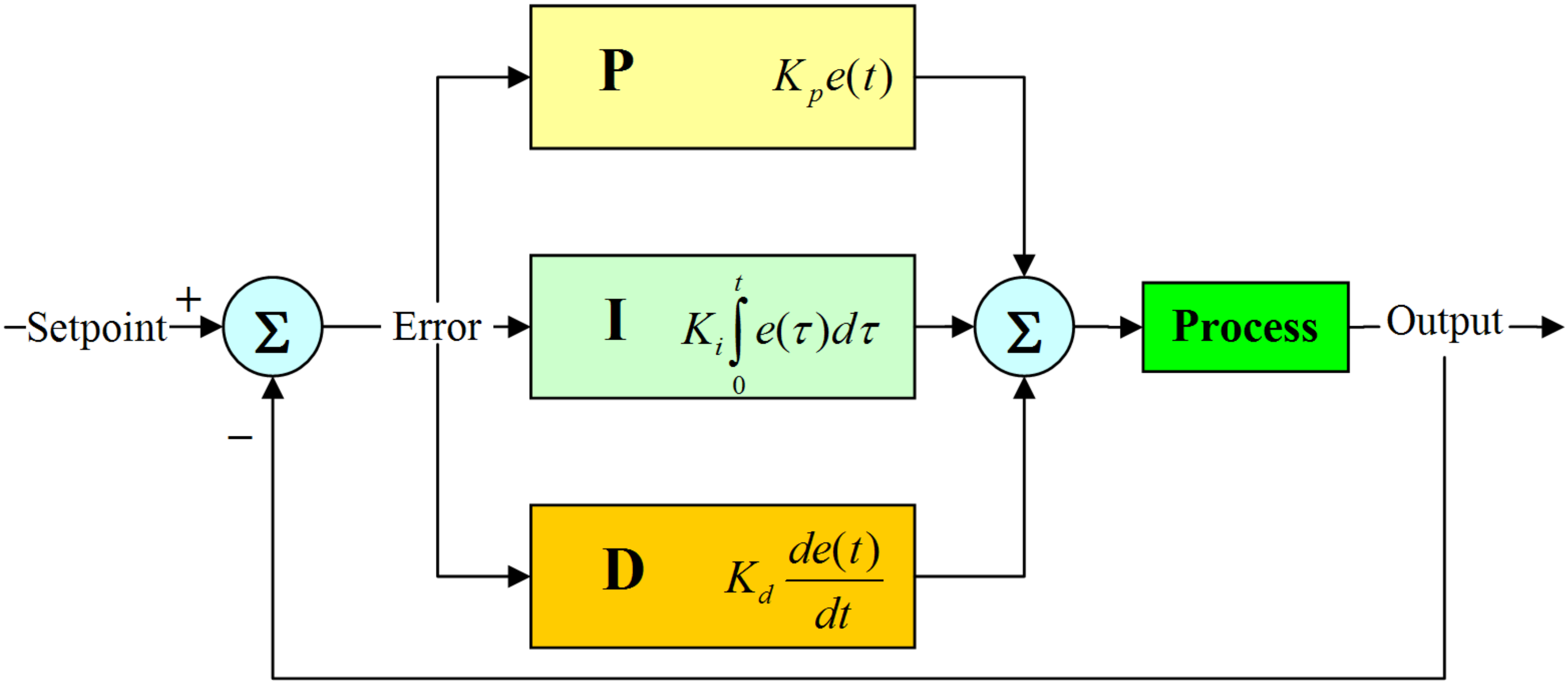
$$(S(x_1, x_2))(t) = y(t) = x_1(t) - x_2(t)$$

Opamp



$$V_o = -\frac{R_2}{R_1}(V_1 - V_2)$$

Model of PID Control



Q&A