MSc on Intelligent Critical Infrastructure Systems

Machine Learning

Lecture 13: Monitoring and Control (Part I)

Marios Polycarpou

Director, KIOS Research and Innovation Center of Excellence

Professor, Electrical and Computer Engineering

University of Cyprus

funded by:

Course outline

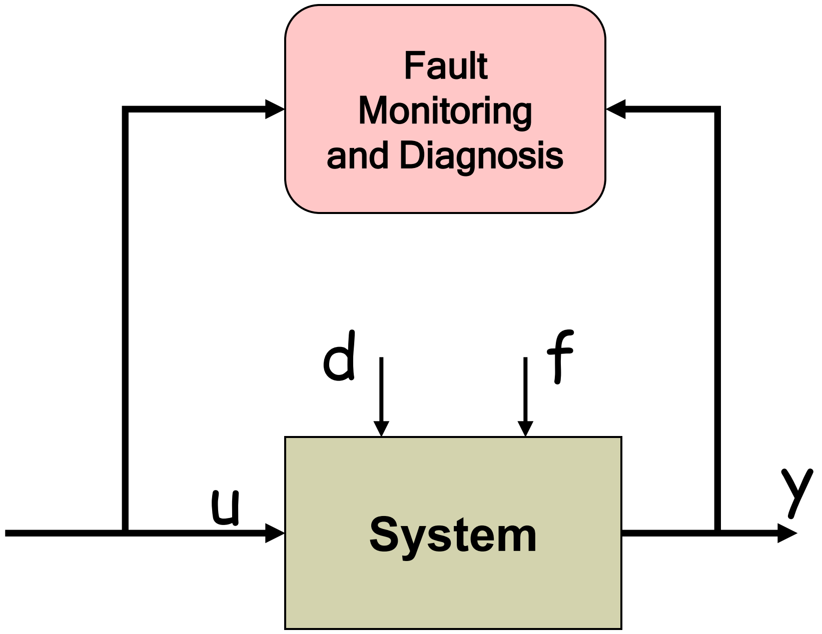
- **Week 1**
  - Introduction and Preliminaries
- **Week 2**
  - Linear Regression
  - Regularisation, Logistic Regression, SVMs
- **Week 3**
  - Neural Networks and Deep Learning
- **Week 4**
  - Feature Engineering and Evaluation
  - Online Learning
- **Week 5**
  - Unsupervised Learning
- **Week 6**
  - Reinforcement Learning
- **Week 7**
  - **Monitoring and Control**

Application to Monitoring and Control – Outline

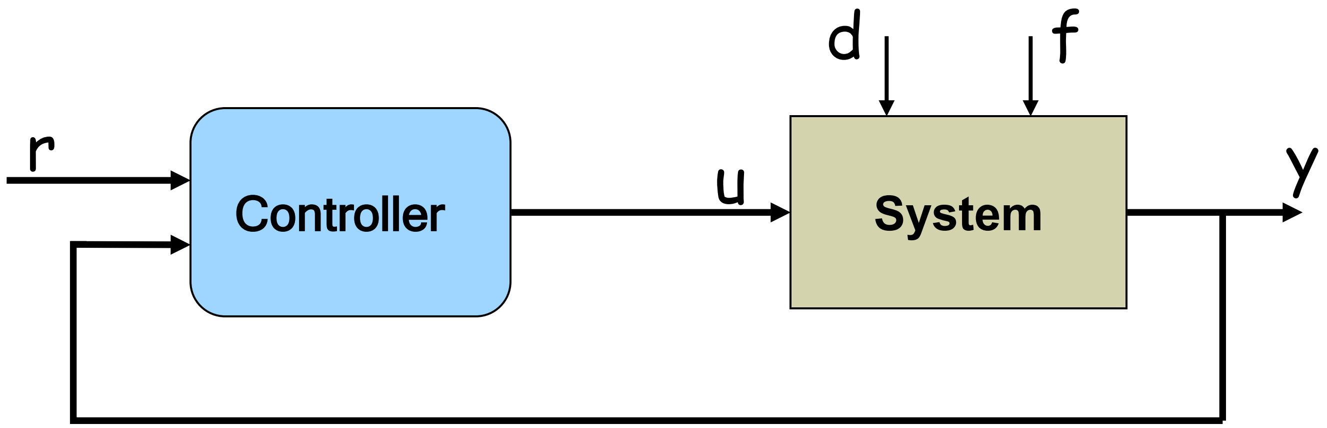
- Introduction to Monitoring and Control
- Mathematical Modelling of Dynamical Systems
- Adaptation and Learning in Control Systems
- System Identification (Continuous-time & Discrete-time)
- Learning Control using Neural Networks

→ CONCLUDING REMARKS FOR THE CLASS

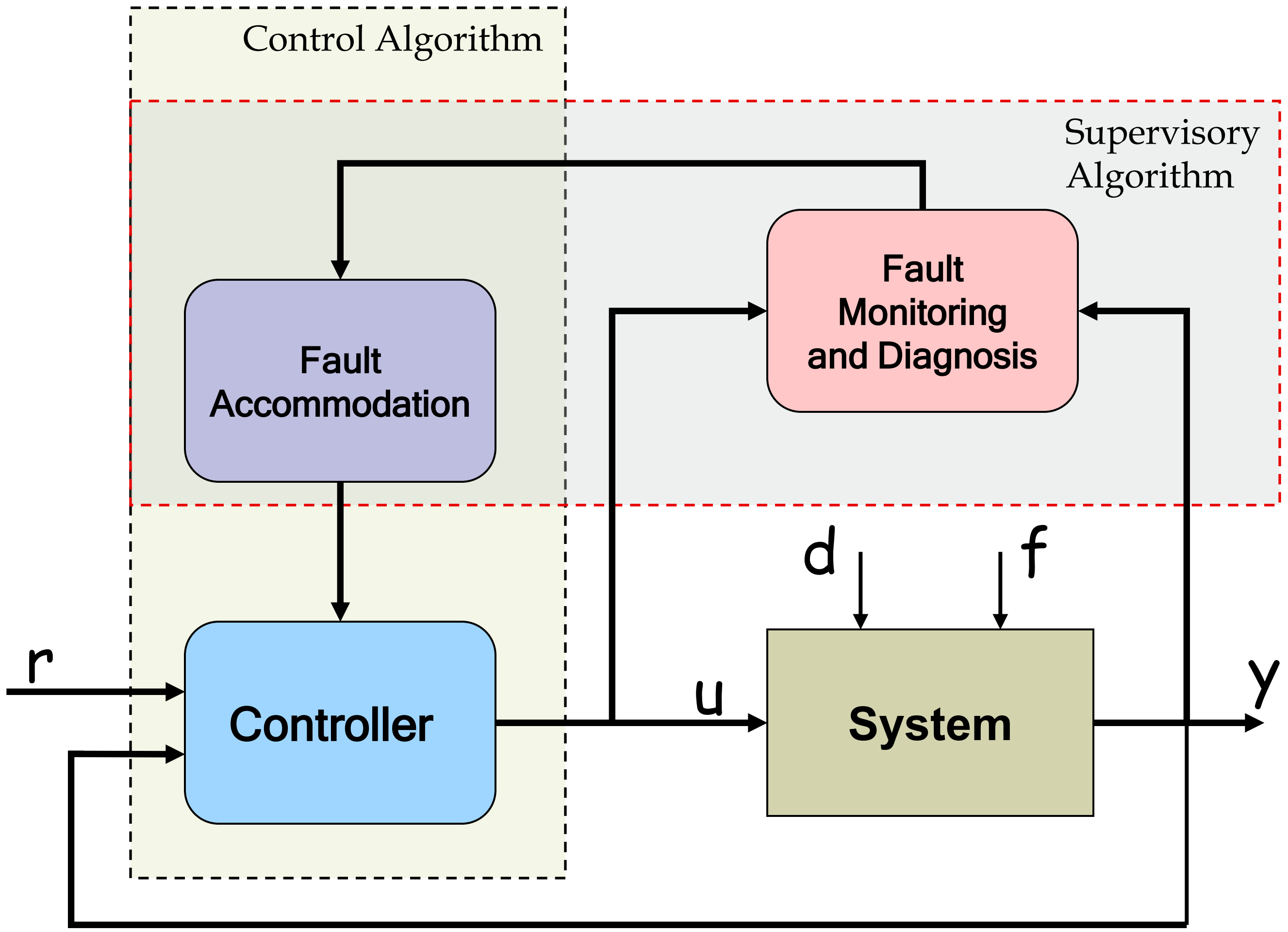
Monitoring and Control



Monitoring and Control



Monitoring and Control

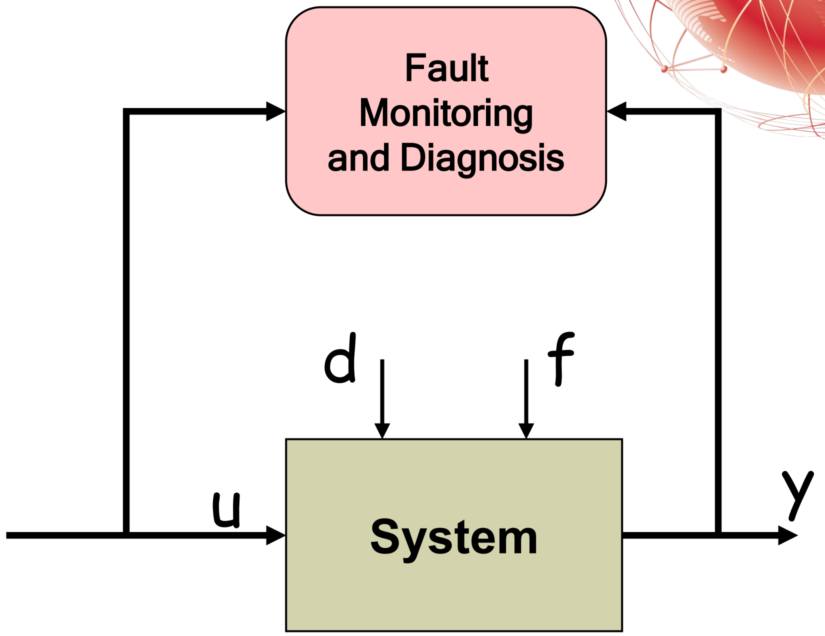


Monitoring and Control Applications

- Power and Energy Systems
- Transportation Systems
- Water Distributions Networks
- Distributed Autonomous Vehicles
- Chemical and Petrochemical Engineering Processes
- Smart Buildings
- Manufacturing Systems
- Biological and Biomedical Engineering Applications
- Environmental Monitoring and Control Applications
- Military and Security Applications

Monitoring Formulation

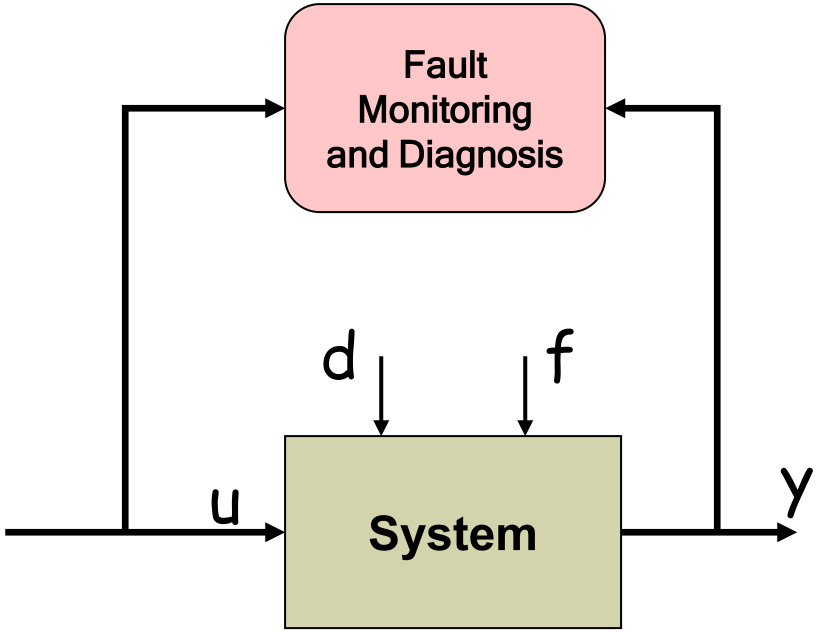
- Utilize input and output data from the system to determine whether it is behaving as it should
- Operates in real-time during operation of the system
- Acts passively; does not affect the behavior of the system
- If it is detected that something is not working well, then an action may be triggered to affect the behavior of the system
- Typically, a monitoring agent consists of:
  - Mathematical model of the system that is being monitored
  - Filtering algorithms
  - Decision logic algorithms



Monitoring Formulation

Why do we need to monitor the System? What can go wrong?

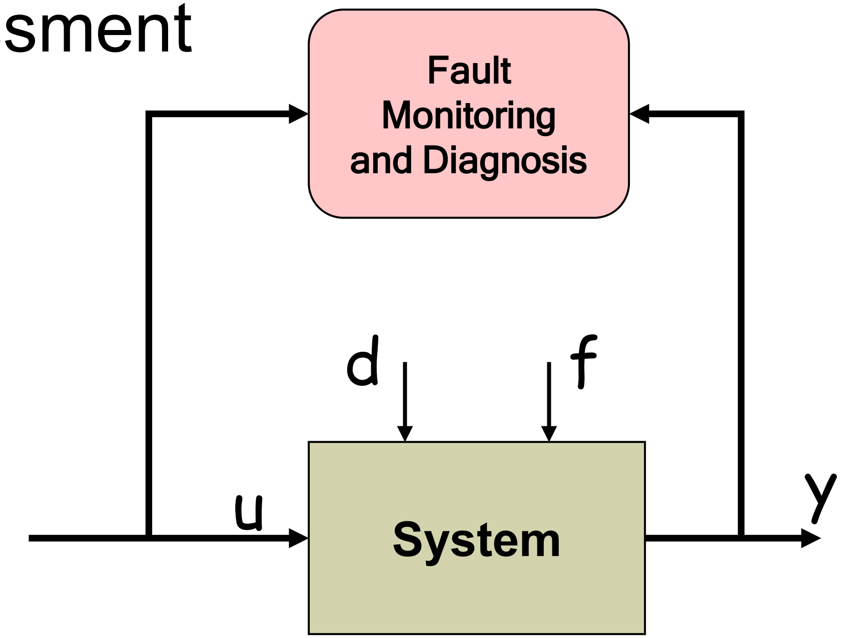
- System/Process Faults
- Actuator Faults
- Sensor Faults
- Communication Faults
- Controller Faults
- Environment Faults
- Malicious Attacks (cyber-security)



Monitoring Formulation

Fault/Event Diagnostic Steps:

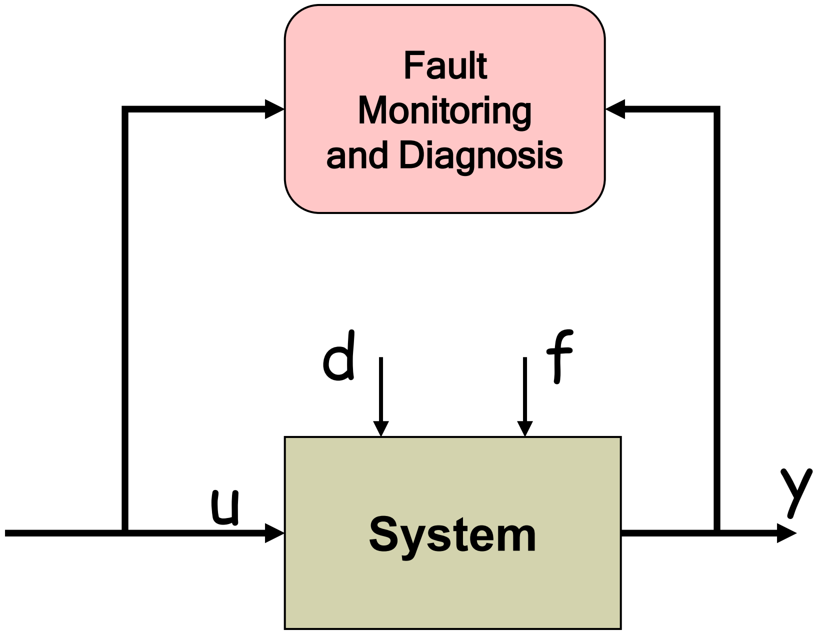
- Fault/event detection
- Fault/event isolation
- Fault/event identification and risk assessment
- Fault/event accommodation
  - Active fault accommodation
  - Passive fault accommodation



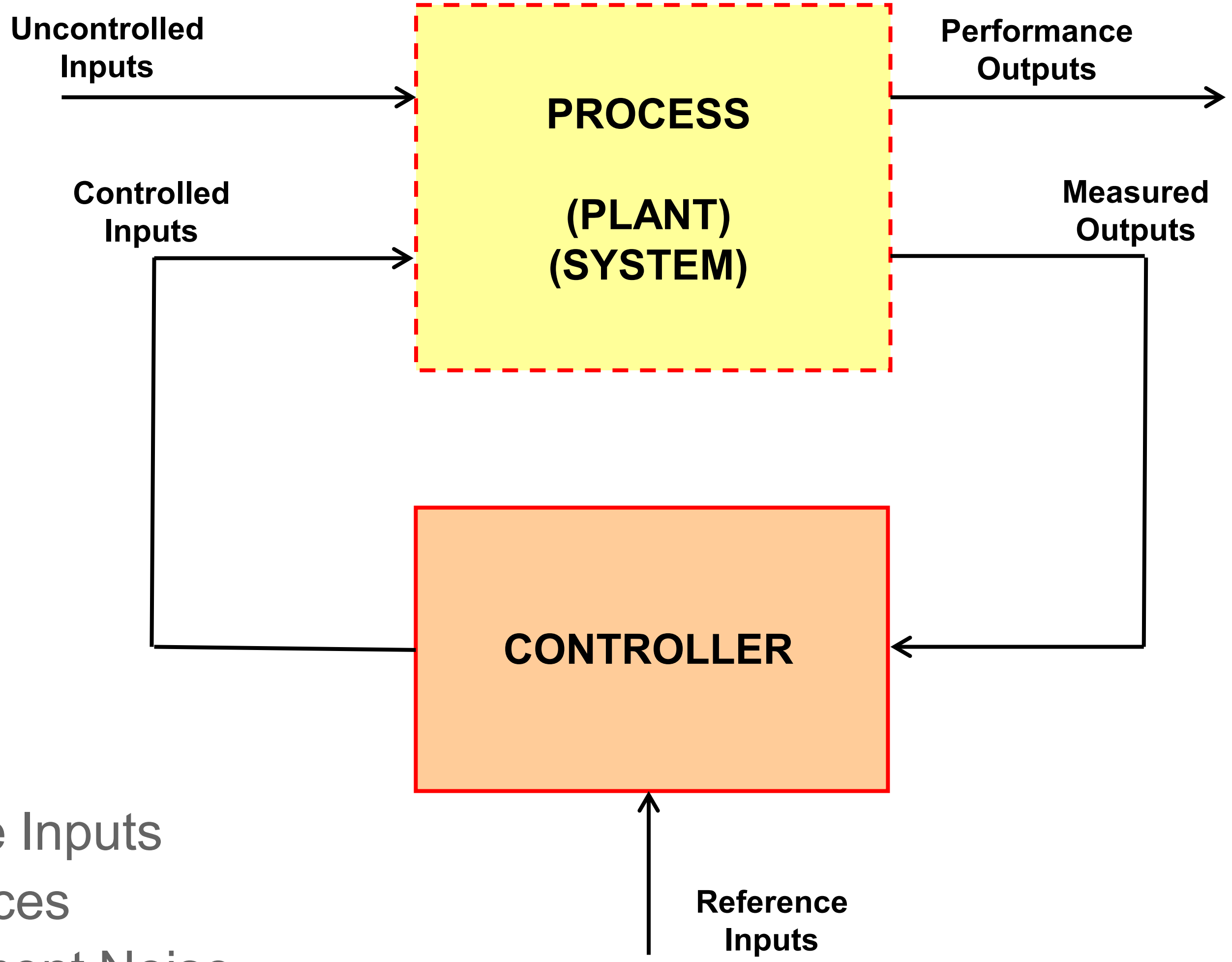
Monitoring Formulation

Key Challenges

- distinguish between faults and modeling uncertainty or measurement noise
- avoid false positives and false negatives
- handle multiple faults
- isolate faults in a large-scale system (needle in a haystack)
- prevent "small" faults from escalating into a major failure
- reliable fault accommodation
- cyber-physical security



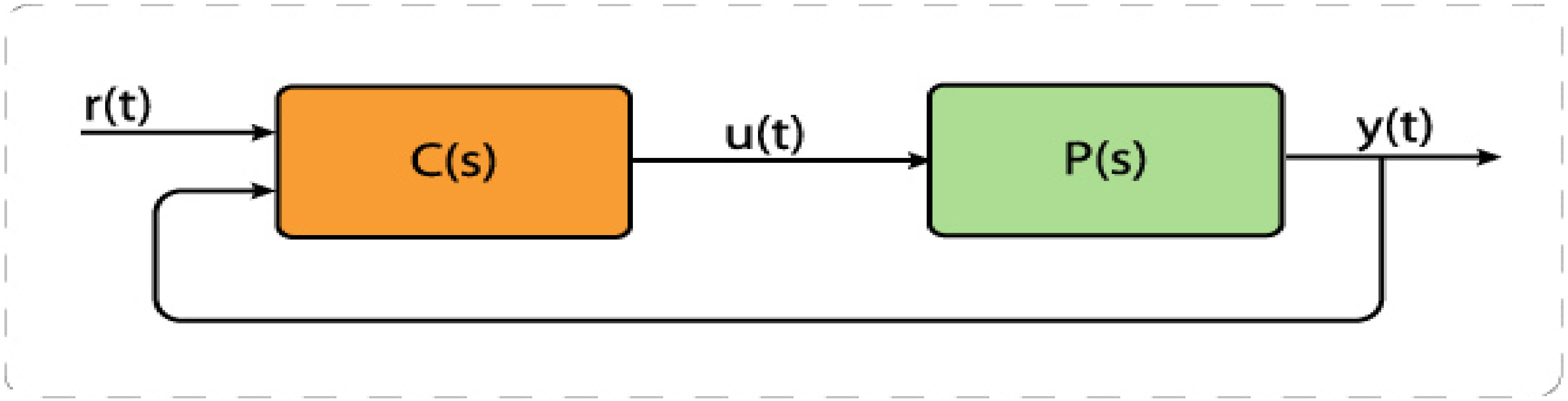
General Control Formulation



Key Issues:

- Sensors
- Actuators
- Reference Inputs
- Disturbances
- Measurement Noise
- ***Feedback***

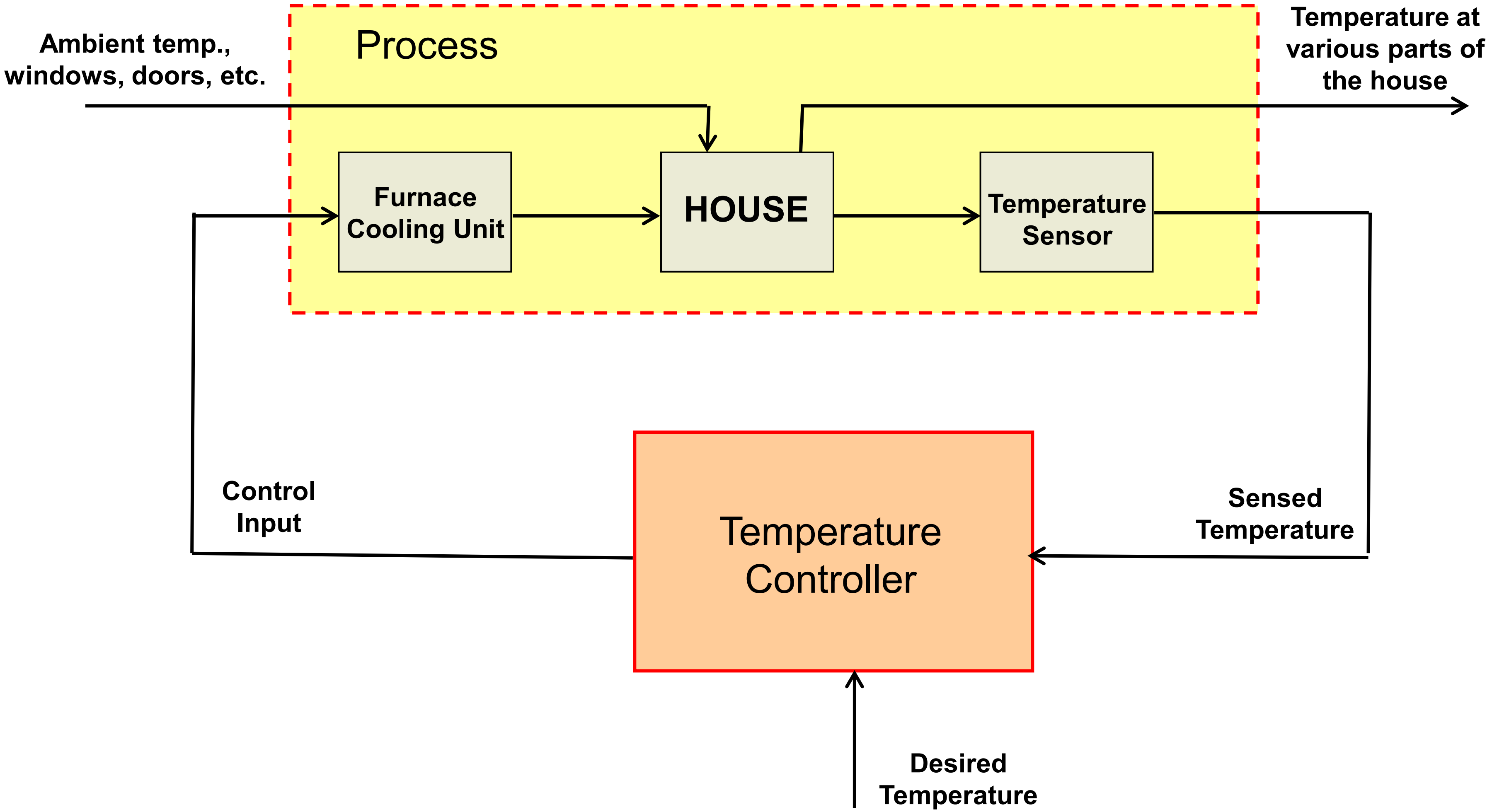
General Control Formulation



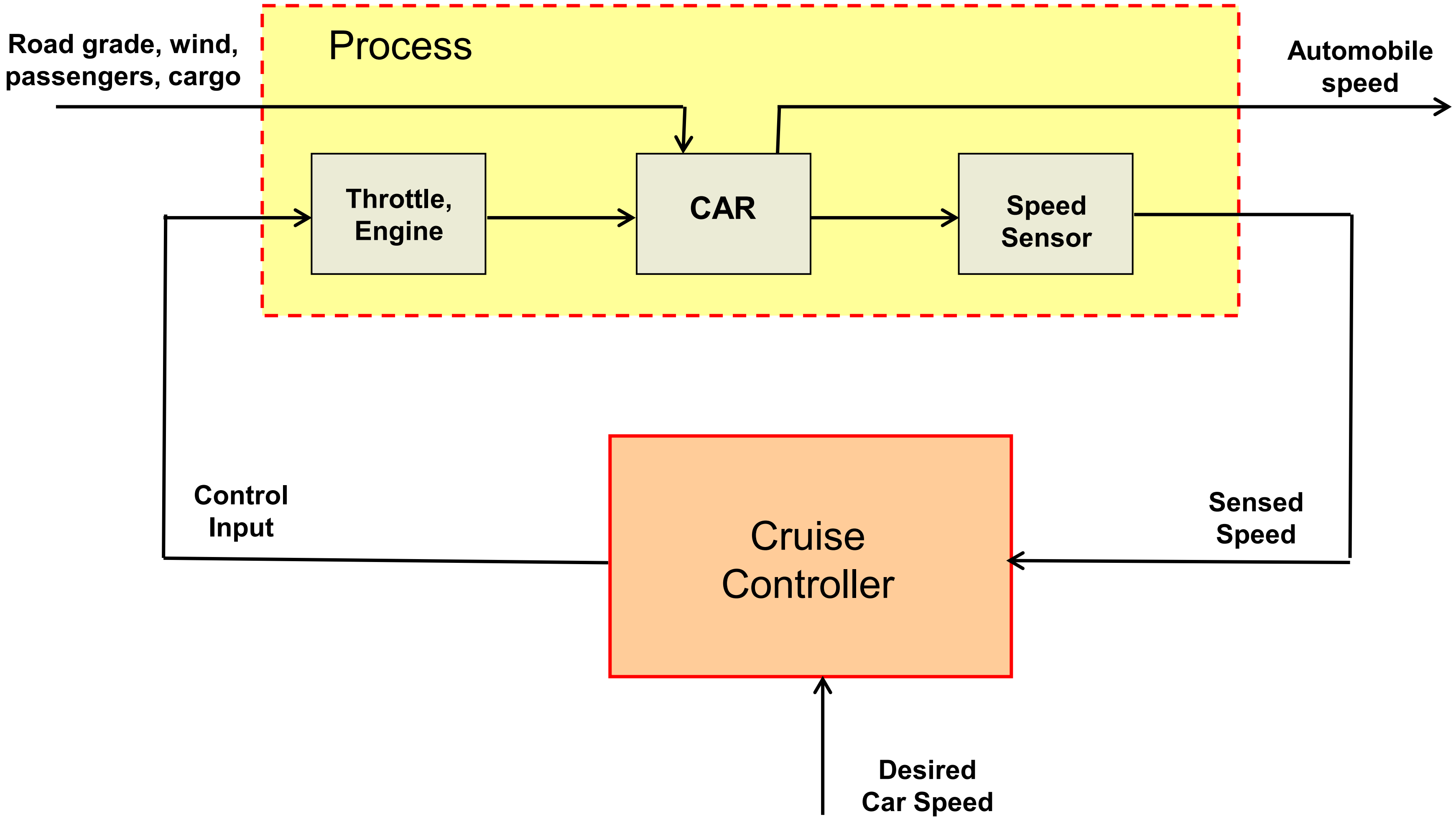
<u>Key Issues:</u>

- Sensors
- Actuators
- Reference Inputs
- Disturbances
- Measurement Noise
- ***Feedback***

Simple Example: Temperature Control



Simple Example: Automobile Cruise Control



Mathematical Modelling

a) Based on first principles of physics (chemistry, biology, economics, etc.)

b) System identification using real data

c) Combination of first principles and system identification

Why do we Need a Mathematical Model?

- Prediction
- Control
- Monitoring
- Design

Mathematical Modelling

How Accurate does a Mathematical Model Need to Be?

- Prediction Model vs Control Design Model
- Limitations of mathematical modelling

→ *Everything Should Be Made as Simple as Possible, But Not Simpler – A. Einstein*

→ *Remember that all models are wrong; the practical question is how wrong do they have to be to not be useful. – George E. Box*

Mathematical Modelling

- Differential/dynamic systems vs algebraic systems
    - Systems with memory – the outputs depend not only on the inputs but also on the initial conditions
- Linear vs nonlinear models
- Continuous-time vs discrete-time models
- Time invariant (stationary) vs time-varying systems (non-stationary)

Mathematical Modelling: the 3 domains

$u(t) \rightarrow \boxed{H(s)} \rightarrow y(t)$

s-plane (Laplace transform)
Transfer Functions

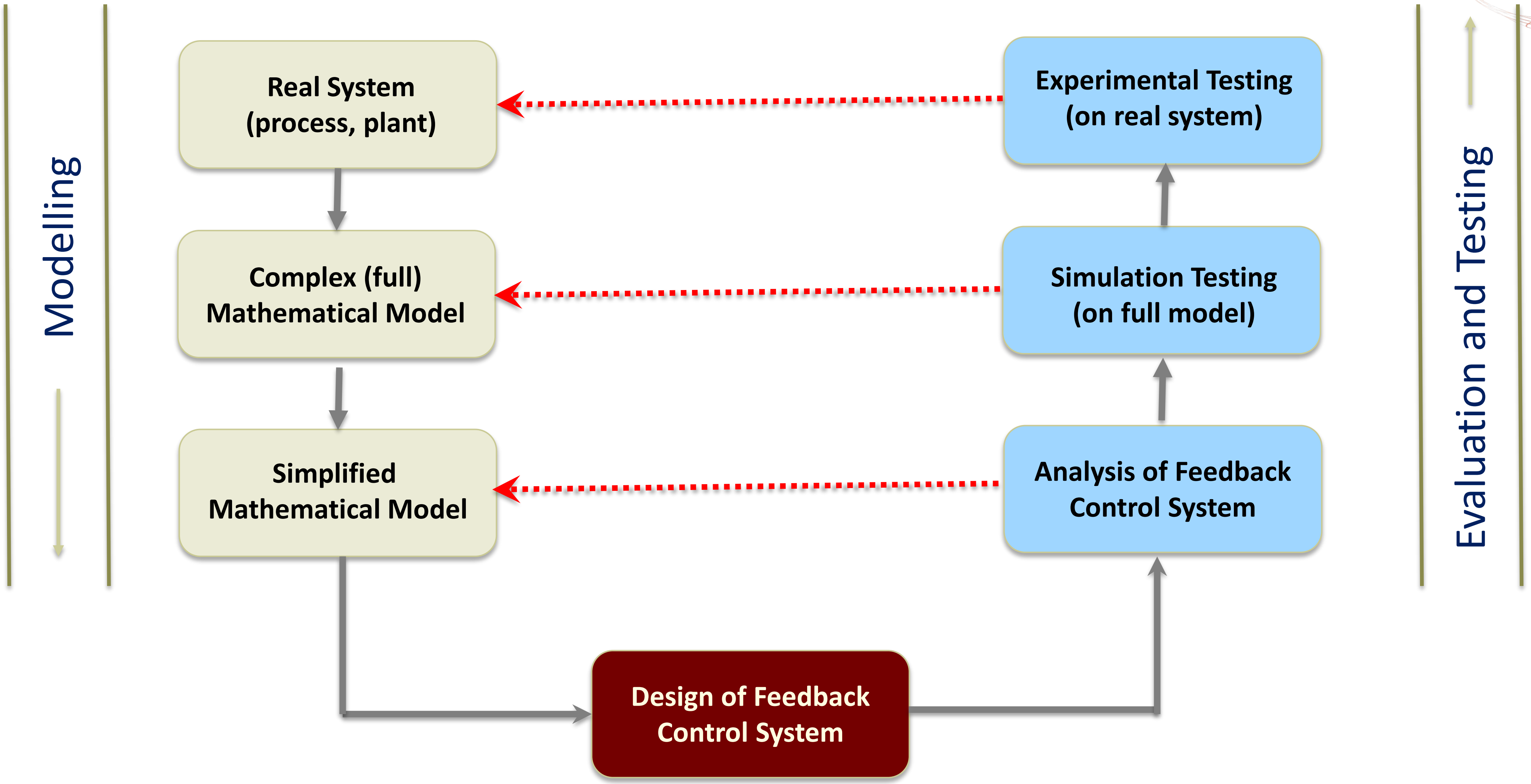
$u(t) \rightarrow \boxed{\begin{array}{c} |H(j\omega)| \\ \angle H(j\omega) \end{array}} \rightarrow y(t)$

Frequency Response
Bode diagrams

$u(t) \rightarrow \boxed{\begin{array}{l} \dot{x} = Ax + Bu \\ y = Cx + Du \end{array}} \rightarrow y(t)$

State-Space (time domain)
Differential Equations

Modelling, Control Design and Evaluation



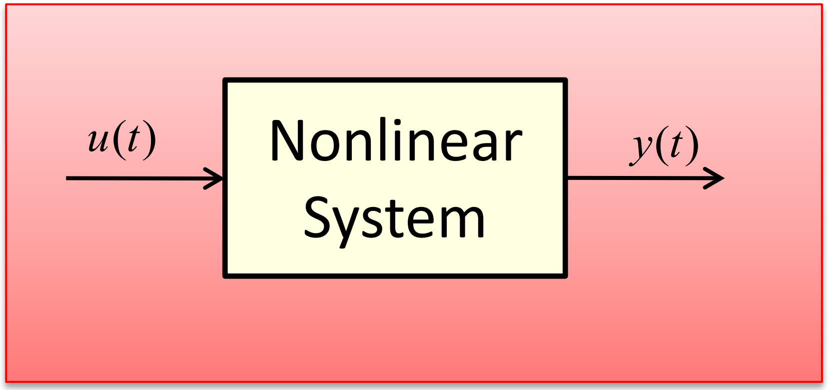
Model-based design vs Data-driven design

- Model-based design methods for monitoring and control are based on developing a model of the system, while data-driven (model-free) methods are based on experimental data
- Some key trade-offs:
  - Model-based design methods facilitates the use of powerful design tools and analysis methods
  - Errors can be detected ar various steps of the procedure
  - Developing a model may be time consuming and expensive
  - Inaccurate models may lead to bad performance

Learning Control: Motivation

$$\dot{x}(t) = f_0(x(t)) + \boxed{f(x(t))} + u(t)$$
$$y(t) = x(t)$$

unknown



Feedback control:

$$u(t) = -f_0(x(t)) + \dot{x}_d(t) - K[x(t) - x_d(t)]$$

Closed-Loop Dynamics:

$$\dot{e}(t) = -Ke(t) + f(x(t))$$
$$e(t) = x(t) - x_d(t)$$

May lead to instability

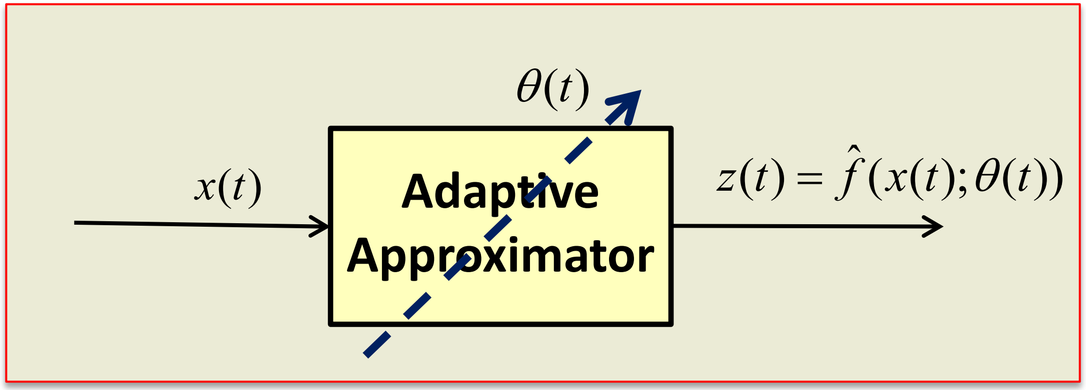
Learning Control: Motivation

Unknown $f(x(t))$ → $\hat{f}(x(t);\theta^*)$ → $\hat{f}(x(t);\theta(t))$

$\hat{f}$: Adaptive approximation structure

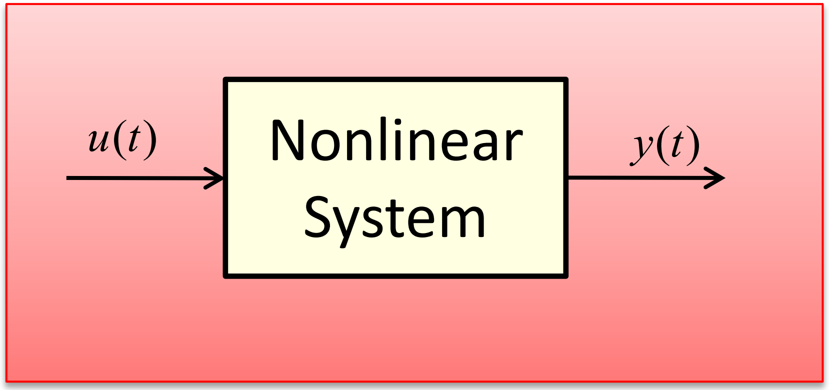
$\theta(t)$: Parameters, weights;
obtained using adaptive methods

LEARNING THE UNCERTAINTY DURING OPERATION OF THE SYSTEM



Learning Control: Motivation

$$\dot{x}(t) = f_0(x(t)) + \underbrace{\boxed{f(x(t))}}_{\text{unknown}} + u(t)$$
$$y(t) = x(t)$$



Feedback control:

$$u(t) = -f_0(x(t)) + \dot{x}_d(t) - K[x(t) - x_d(t)] - \hat{f}(x(t);\theta(t))$$

Closed-Loop Dynamics:

$$\dot{e}(t) = -Ke(t) + \boxed{f(x(t)) - \hat{f}(x(t);\theta(t))}$$
$$e(t) = x(t) - x_d(t)$$

Improves performance and reduces the possibility of instability