Dynamics and Control

ΕΠΛ 428: IOT PROGRAMMING

Dr. Panayiotis Kolios
Assistant Professor, Dept. Computer Science,
KIOS CoE for Intelligent Systems and Networks
Office: FST 01, 116
Telephone: +357 22893450 / 22892695
Web: https://www.kios.ucy.ac.cy/pkolios/

Πανεπιστήμιο Κύπρου

- Dynamical systems provide a mathematical framework to:
    - describe and understand the world around us
    - modeling the rich interactions between quantities that co-evolve in time
    - make predictions and take actions/ control
- Use differential equations or iterative mappings (data-driven)
- Data is abundant, while physical laws or governing equations remain elusive

- Dynamical System

$$\frac{\partial}{\partial t}\boldsymbol{x}(t) = \boldsymbol{f}(\boldsymbol{x}(t), t; \boldsymbol{\beta})$$

where x is the state of the system and f is a function that models the system over time t with parameters β

- Discrete-Time Dynamical Systems

$$\boldsymbol{x_{k+1}} = \boldsymbol{F}(\boldsymbol{x_k})$$

where $\mathrm{x_k}$ can be obtained by sampling the signal in time, i.e. $x_k = x(k\Delta t)$

- Desirable to work with linear dynamics of the form

$$\frac{\partial}{\partial t}\boldsymbol{x} = \boldsymbol{A}\boldsymbol{x}$$

- Linear dynamical systems admit closed-form solutions

$$\boldsymbol{x}(t_0 + t) = e^{\boldsymbol{A}t}\boldsymbol{x}(t_0)$$

- Real-world systems are generally **nonlinear** and exhibit multi-scale behavior in both space and time
- Also there is **uncertainty** in the equations of motion, in the specification of parameters, and in the measurements of the system
- Identifying unknown dynamics from data and learning intrinsic coordinates can enable linearization of nonlinear and uncertain dynamical systems

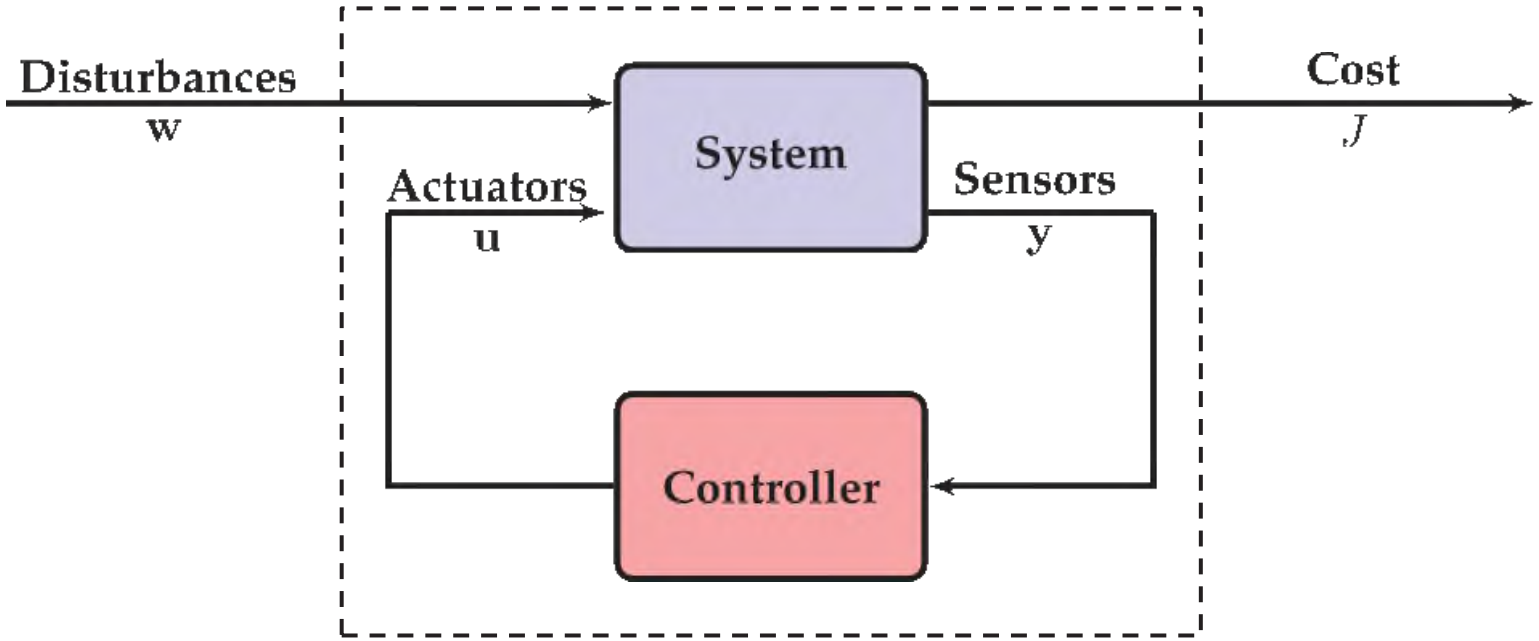
CONTROL SYSTEMS

Πανεπιστήμιο Κύπρου

- An overarching goal for many systems is the ability to actively manipulate their behavior for a given engineering objective
- Control theory -> manipulating dynamical systems
    - Relies on sensor measurements (data) obtained from a system to achieve a given objective
- Helped shape modern technology
    - cruise control in automobiles
    - Position control in drones
    - Packet routing in the Internet
    - Heating/cooling ventilation
    - Temperature and pressure control in modern espresso machines

- **Passive control**
  - does not require input
  - Desirable due to simplicity, reliability, and low cost
  - e.g., stop signs at a traffic intersection regulate the flow of traffic
- **Active control**: **Open-loop** or **closed loop** depending if sensors measurements are used to inform control
- Open-loop relies on pre-programmed control sequences
  - e.g. traffic signal periods set at different times of day
- Closed-loop control uses sensors to measure the system directly and then shapes the control to achieve the goal
  - e.g., smart traffic lights with a control logic informed by inductive sensors in the roadbed that measure traffic density

- Used for systems with uncertainty, instability, and/or external disturbances



- Sensor measurements, y, are fed back into a controller, which then decides on an actuation signal, u, to manipulate the system despite model uncertainty and exogenous disturbances

- Exogenous disturbances w may be decomposed as

$$w = \left[w_d^T w_n^T w_r^T\right]^T$$

where w_d are disturbances to the state of the system, w_n is measurement noise, and w_r is a reference trajectory that should be tracked by the closed-loop system

- System & measurement model are:

$$\frac{\partial}{\partial t}\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w}_d)$$

$$\boldsymbol{y} = \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w}_n)$$

- and the objective (control law) is:

$$\boldsymbol{u} = \boldsymbol{k}(\boldsymbol{y}, \boldsymbol{w}_r)$$

- that minimizes the cost function $J = J(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w}_r)$

- Modern control relies heavily on techniques from optimization
- Benefits of feed-back control
    - Possible to stabilize an unstable system;
    - Possible to compensate for external disturbances;
    - Possible to correct for unmodeled dynamics and model uncertainty

$$\frac{\partial}{\partial t}\boldsymbol{x} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} = (\mathbf{A} - \mathbf{B}\mathbf{K})\boldsymbol{x}$$

- Let u be the rate of gas fed into the engine, and let y be the car's speed
- Neglecting transients, a crude model is:

$$y = u$$

- If we double the gas, we double the automobile's speed
- Consider the closed-loop control law:

$$u = K(w_r - y)$$

- so that gas is increased when the measured velocity is too low, and decreased when is too high

- In practice, the model could be $y = 2u$, or external disturbances, such as rolling hills, affect speed by $y = u + \sin(t)$
- The performance of the closed-loop system is:

$$y = 2K(w_r - y)$$

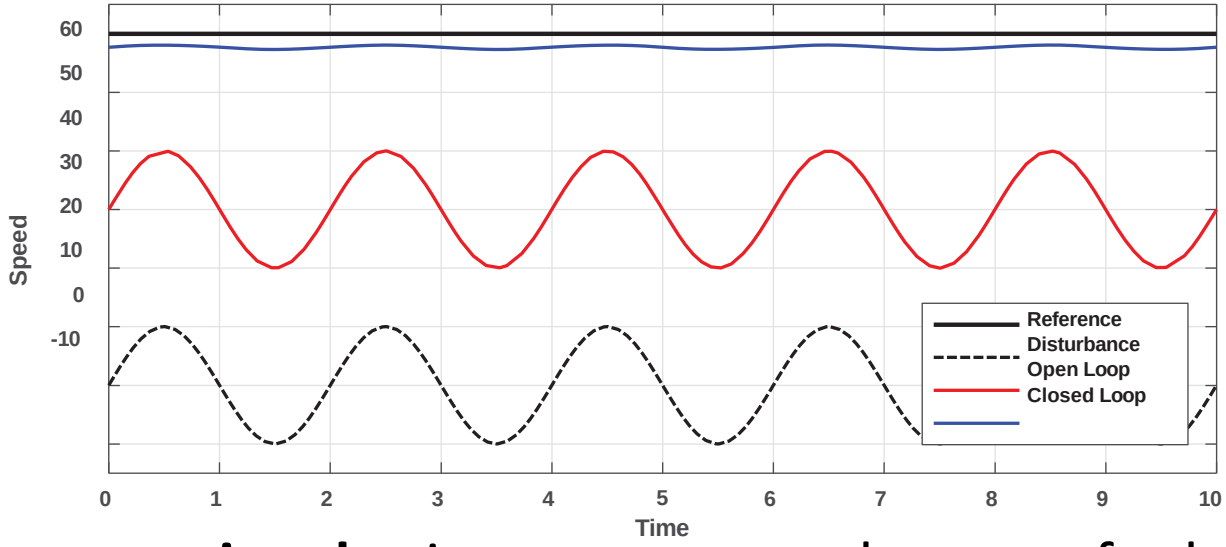
$$(1 + 2K)y = 2Kw_r$$

$$y = \frac{2K}{1 + 2K} w_r$$

- For K = 50, the closed-loop system only has 1% steady-state tracking error
- Similarly, an added disturbance w_d will be attenuated by a factor of 1/(2K + 1)

- Consider a reference tracking problem
  - desired reference speed of 60 mph
  - model is y = u (true system is y = 0.5u)
  - disturbance in the form of rolling hills that increase and decrease the speed by ±10 mph at a frequency of 0.5 Hz.
  - K = 50

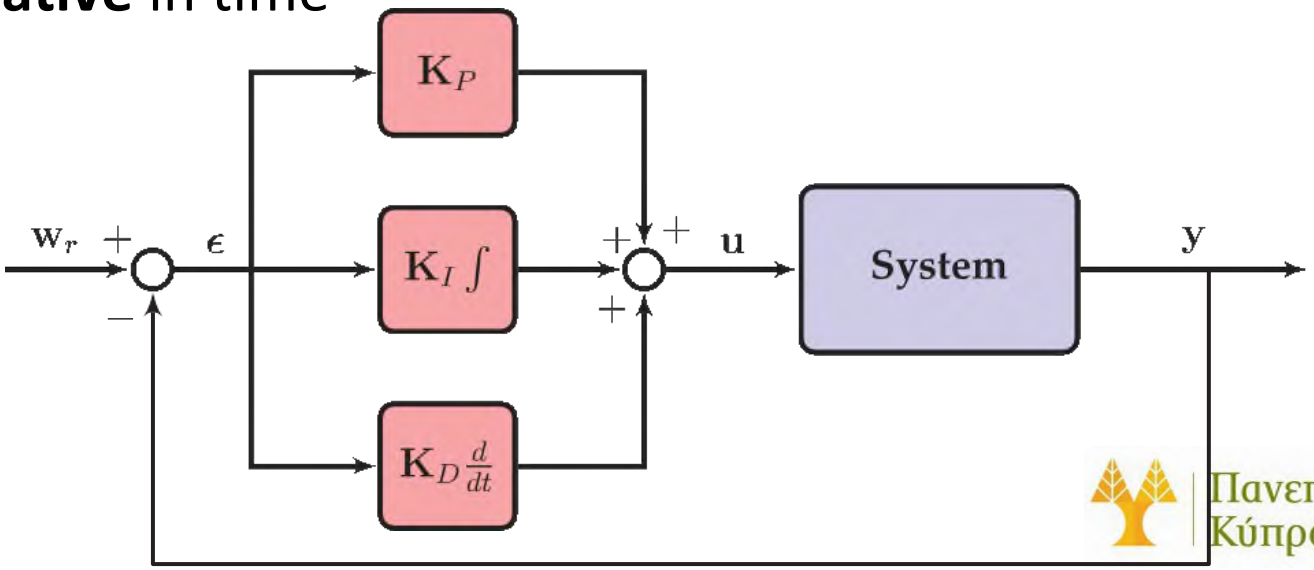
```
# Comparison of open- and close-loop control
t = 0:.01:10;                   % time

wr = 60*ones(size(t));          % reference speed
d = 10*sin(pi*t);               % disturbance

aModel = 1;                     % y = aModel*u
aTrue = .5;                     % y = aTrue*u

uOL = wr/aModel;                % Open-loop u based on model
yOL = aTrue*uOL + d;            % Open-loop response

K = 50;                         % control gain, u=K(wr-y);
yCL = aTrue*K/(1+aTrue*K)*wr + d/(1+aTrue*K);
```



- A large **proportional** gain may come at the cost of robustness

Πανεπιστήμιο Κύπρου

- PID control is among the simplest and most widely used control architectures for:
  - motor position and velocity control
  - tuning of various sub-systems in an automobile
  - pressure and temperature controls in modern espresso machines
- PID control additively combines three terms to form the actuation signal, based on the **error signal**, its **integral** and **derivative** in time



- In the cruise control example
  - was possible to reduce reference tracking error by increasing the proportional control gain K_P in the control law $u = -K_P(w_r - y)$
  - However, increasing the gain may eventually cause instability in some systems, and it will not completely eliminate the steady-state tracking error
  - Addition of an integral control term $K_I \int_0^t (w_r - y)$ is useful to eliminate steady-state reference tracking error while alleviating the work required by the proportional term

- There are formal rules for how to choose the PID gains for various design specifications:
  - fast response
  - minimal overshoot
  - Ringing

- Most complete theory of control has been developed for linear systems
- Linear systems are generally obtained by linearizing a nonlinear system about a fixed point or a periodic orbit
- However, instability may quickly take a trajectory far away from the fixed point
- Fortunately, an effective stabilizing controller will keep the state of the system in a small neighborhood of the fixed point where the linear approximation is valid

- Also many systems of interest are exceedingly high dimensional, making them difficult to characterize/ model
- High dimensionality also limits controller robustness due to significant computational time delays
- Reduced-order models capture the most relevant dynamics for feedback control
- Related procedures include model reduction and system identification (when having access only to data)

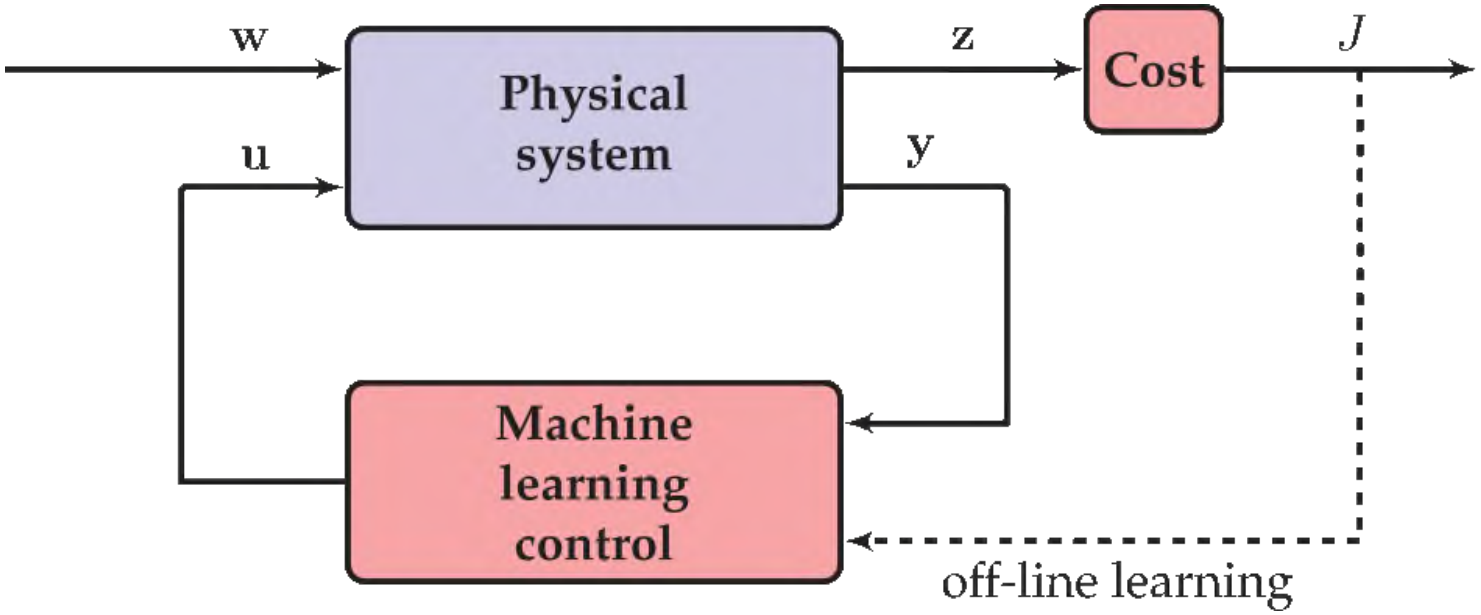
- The goal of **system identification** is to identify a low-order model of the input–output dynamics from actuation u to measurements y.
- If we are able to measure the full state x of the system, then this reduces to identifying the dynamics $\boldsymbol{F}$ that satisfy:

$$\boldsymbol{x_{k+1}} = \boldsymbol{F}(\boldsymbol{x_k}, \boldsymbol{u_k})$$

- When the dynamics are approximately linear:

$$\boldsymbol{x_{k+1}} = A\boldsymbol{x_k} + \boldsymbol{B u_k}$$

- Machine learning enables complex systems to be described by measurement data, rather than first-principles modeling



- Use ML to learn control laws (i.e., determine map from sensors to actuators)

- ML encompasses a broad range of high-dimensional, possibly nonlinear, optimization techniques, it is natural to apply machine learning to the control of complex, nonlinear systems
- ML methods for control include:
  - Adaptive neural networks
  - Genetic algorithms
  - Genetic programming
  - Reinforcement learning
- All of these are **model-free control methodologies**
- Model-free methods have some sort of macroscopic objective function, typically based on sensor measurements (past and present)
  - Objectives involve some minimization or maximization of a given quantity subject to some constraints
  - Constraints may be hard, or they may involve a complex multi-objective tradeoff

- Reinforcement learning (RL)
  - at the intersection of machine learning and control
  - used for generalized & generative AI, autonomous robots, and self-driving cars

- Markov decision process (RL framework), where the dynamics of the system and the control policy are described in a probabilistic setting, so that stochasticity is built into the state dynamics and the actuation strategy

- In this way, control policies are probabilistic, promoting a balance of **optimization** and **exploration**

- With RL may not always have immediate rewards
- Or RL agent may receive partial information about the effectiveness of their control strategy
    - For example, when learning to play a game like tic-tac-toe or chess, it is not clear if a specific intermediate move is responsible for winning or losing.
    - The player receives binary feedback at the end of the game as to whether or not they win or lose
- RL defines a value function, **Q**, for the quality of being in a particular state and taking a particular action
- Over time, the agent learns and refines **Q function**, improving their ability to make good decisions
    - In the example of chess, an expert player begins to have intuition for good strategy based on board position, which is a complex value function over an extremely high-dimensional state space (i.e., the space of all possible board configurations)
- Q-learning is a model-free reinforcement learning strategy, where the value function is learned from experience

- **Environment**: The world in which the agent operates
- **State**: A specific situation in the environment
- **Action**: A move made by the agent
- **Reward**: Feedback from the environment based on the action taken
- **Policy**: A strategy used by the agent to determine the next action based on the current state
- **Q-value**: A measure of the expected future rewards for an action taken in a specific state

- Q-learning is an off-policy RL algorithm that seeks to find the best action to take given the current state
- It learns a Q-function, which is the expected utility of taking a given action in a given state, and following the optimal policy thereafter
- The Q-value is updated using the formula:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[r + \gamma \max_{a'} Q(s',a') - Q(s,a) \right]$$

where:

- $Q(s,a)$ is the current Q-value
- α is the learning rate
- r is the reward
- γ is the discount factor
- s' is the next state
- $\max_{a'} Q(s',a')$ is the maximum Q-value for the next state