Digital Control

CSE421

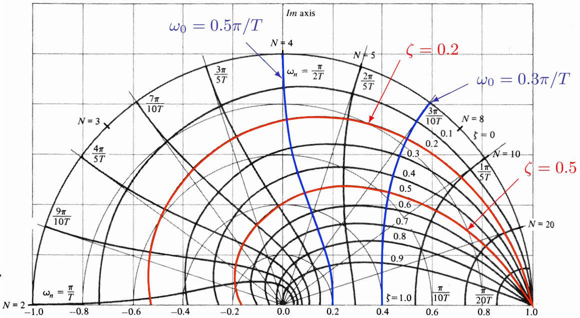
Assoc. Prof. Dr.Ing.

Mohammed Ahmed

mnahmed@eng.zu.edu.eg

goo.gl/GHZZio

Lecture 1: **Introduction**



The Lecturer

About Me

Mohammed Nour *Abdelgwad* **Ahmed**

- **Asst.Prof.Dr.Ing.** at Computer and Systems Engineering Dept., Faculty of Engineering, Zagazig University.
- **Researcher** at DFKI–Robotic Innovation Center, Bremen, Germany.
- Research Interests: Robotics, Control, Modelling and Simulation, and Mechatronics

- **web**: `https://mnourgwad.github.io`, `www.mnahmed.faculty.zu.edu.eg`
- **email**: `mnahmed@eng.zu.edu.eg`
- **Office**: Electric Eng. Building (27), 3rd Floor, Room: 27325
- Office **Hours**: each Sunday and Tuesday after 14:00 o'clock

The Lecturer

Some robots I worked with

SpaceClimber

CREX

The Lecturer

Some robots I worked with

LIMES (Mantis)

EOscc2

The Lecturer

Some robots I worked with

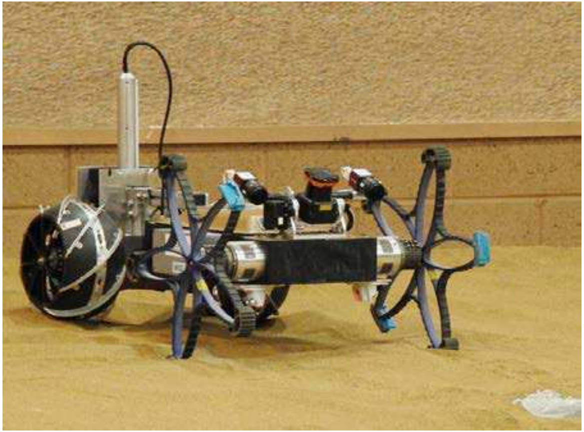
AILA

Mr.SemProm

The Lecturer

Some robots I worked with

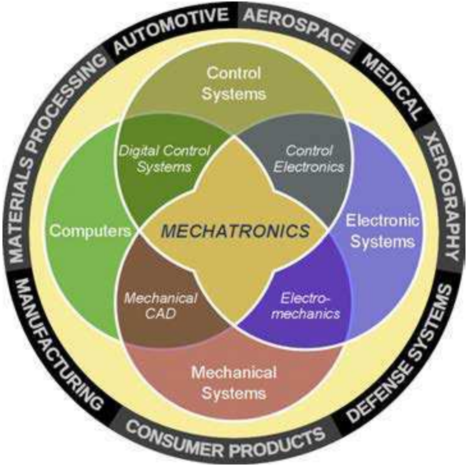
MIRA

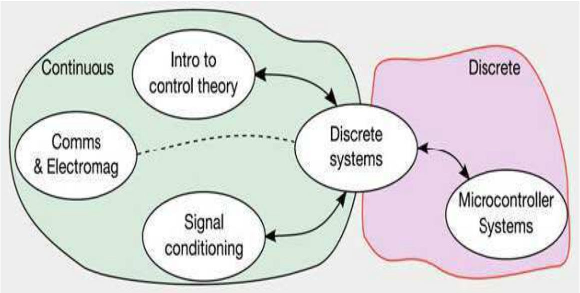


Coyote2

The Course

CSE421 Digital Control





https://en.wikipedia.org/wiki/Mechatronics

The Course

CSE421 Digital Control

The course introduces fundamental concepts in the **theory, analysis and design of discrete control systems**. It enables you to:

- Knowledge and understanding
  - **Model** and analyze discrete control systems
  - Evaluate the **performance** of discrete control systems
- Professional and practical skills
  - **Design and simulate** industrial and practical systems
  - Improve performances of discrete control systems
- General and transferable skills
  - Understand the requirements and operations of discrete control systems
  - Design and tuning techniques for performance improvement

CSE421 Digital Control

Topics to be covered

- Linear Discrete Systems
- Sampling and aliasing
- The Z -transform
- Block diagrams
- Stability
- Controller design using transfer functions
- State-space description of continuous & discrete systems
- State-space design of digital control systems

Assessment Methods

Method	Time	Weight
Assignments, Quizzes, $\cdots$	weekly	15
Midterm	week 6	25
Project (Project Report + Demo)	week 10	10
Final	week 12	100

Recommended Textbooks

- M. Sami **Fadali** and A. Visioli, *Digital Control Engineering Analysis and Design* (2^{nd} ed.), Elsevier, 2012.
- G F **Franklin**, J D Powell & M Workman, *Digital Control of Dynamic Systems* (3^{rd} ed.), Addison Wesley, 1998.
- R C **Dorf** & R H Bishop, *Modern Control Systems*, Pearson Prentice Hall, 2008.

Relevant Websites

- Lecture slides, notes and others on course webpage: https://mnourgwad.github.io/CSE421

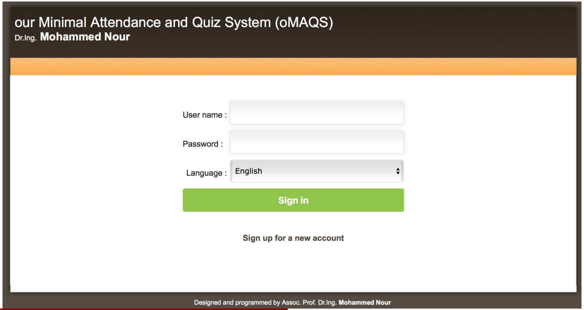
Sign up to the System

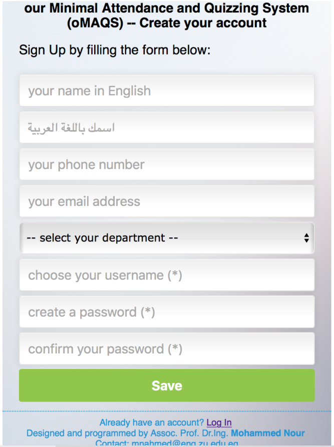
In your **smart** phone:

1. connect to WiFi network **Nour**
2. password: **12345678**

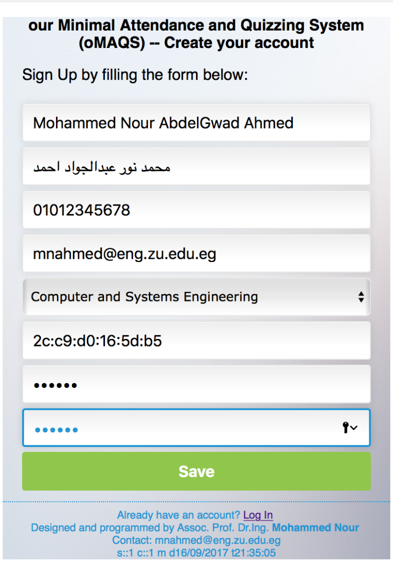
in phone Internet browser:

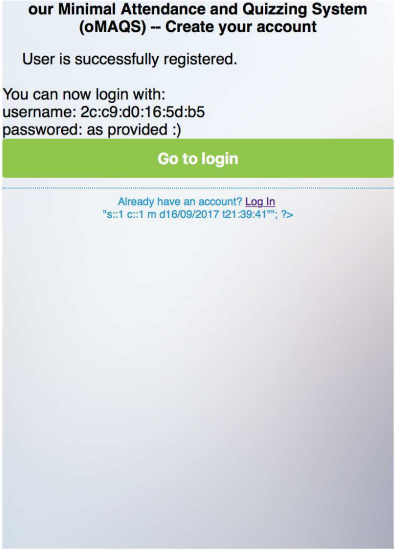
1. navigate to the address: **192.168.1.2**



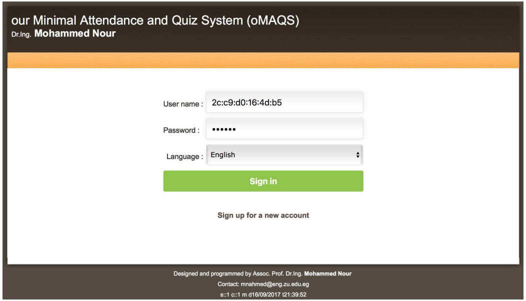


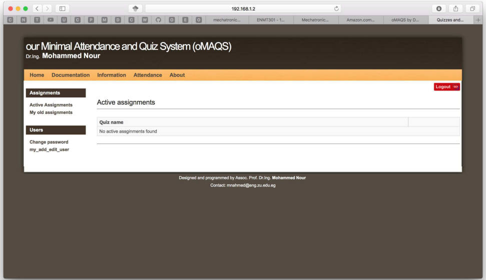
Sign up to the System





Sign up to the System





Lecture: 1

Introduction

- Basic Definitions
- Linear Discrete Systems
  - Major Components
  - Basic Operations
  - Advantages and Disadvantages
- Sampling and Aliasing

Control systems

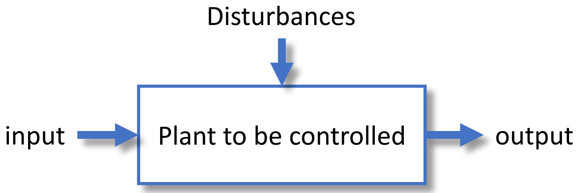
Control system

mechanical, optical, or electronic device, or set of devices, that manages, commands, directs or regulates the behavior of other devices or systems **to maintain a desired output**.

Two **approaches** for control:

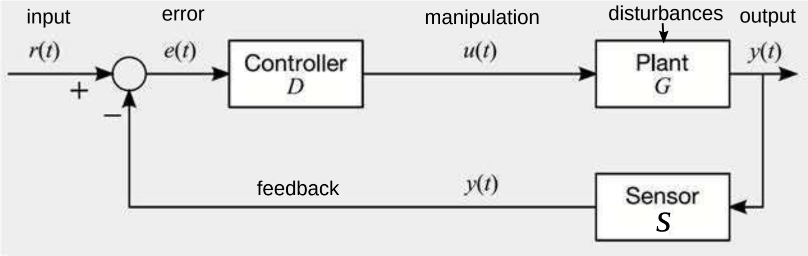
- Open loop control.
- Closed–loop (feedback) control.

Open-loop control



- adjust input to keep the output as close as possible to some desired value.
- However, because of the unknowns in the system model and the effects of external disturbances open-loop control is not accurate.

Closed-loop (Feedback) control



- measurements of plant output is used to modify its input.
- controller receives the error signal, then generates a suitable value of the plant input, hence closing the loop.

Advantages of closed-loop control

- Remove (isolate or **reject**) the unwanted **disturbance** signal(s)
- **Reduces sensitivity** of output to variations in plant parameters.
  - plant model is not required to be exactly known.
- Can **stabilize** the system (if unstable)
  - Open-loop control can not be used in this case!
- **Command Tracking**: cause the output to track the reference input closely

Continuous control systems

Generally, plant inputs and outputs are continuous signals both in time and in amplitude.

- e.g., consider the plant to be controlled is a motor. Its input (current or voltage) and output (speed) are defined and may change at every instant in time (continuous in time).
- these variable can take any value within certain range (continuous in amplitude).

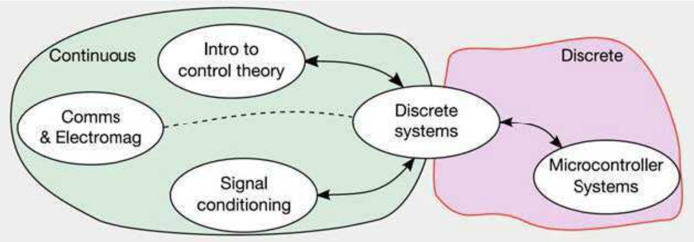
in continuous closed-loop control, **at each time instant**, the output is fed back, the error is calculated, the controller generates a control signal

- Methods of designing $D(s)$:
  - **Time domain**: set damping ratio ζ and natural frequency ω
  - **Frequency domain**: set gain and phase margins of $D(s)\,G(s)$

Digital control system

Discrete system

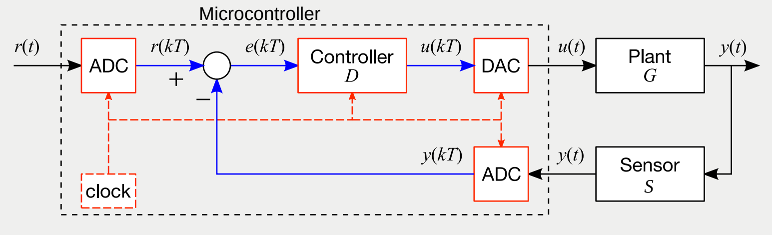
a dynamic system with at least one discrete (quantized) variable.



- This course is on discrete systems in which time is the discrete variable. Digital computers and microcontrollers are widely used in control systems.
  - Computers perceive the world (sensors) & interact with it (actuators) as if it was a discrete system.

Discrete-Time Control Systems

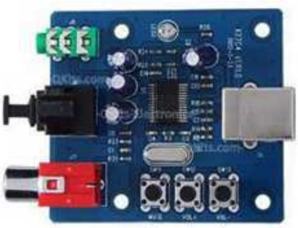
- To control a physical system or process which is analog using a digital controller,
  - e.g., continuous-time systems controlled by a digital computer with interfaces.



Note: Controller input $e(kT)$ & output $u(kT)$ are discrete variables.

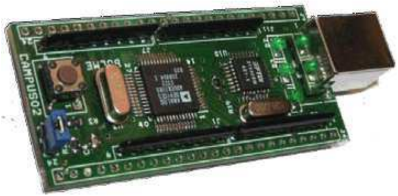
- Such (**Discrete**-Time Control/Digital Control) system consists of **four major parts**:
  1. **Plant**: a continuous-time dynamic system.
  2. Analog-to-Digital Converter (**ADC**).
  3. **Controller**: a microprocessor (μP) with a real-time OS (RTOS).
  4. Digital-to-Analog Converter (**DAC**) .

Analog to Digital Converter (ADC)



- **samples** analog signal (typically a voltage) and then converts these samples into an integer number (**quantization**) suitable for processing by digital computer.
- typically has ranges: $0 \to 5$ V, $0 \to 10$ V (unipolar) or ± 5 V, or ± 10 V (bipolar).
- has **quantization error** given by the converter resolution in bits.
  - Common resolutions are 8 bits (256 levels), and 12 bits (4096 levels).
  - 12-bit ADC of range ± 10 V would have a conversion quantum of $q = 20/4096 = 4.88$ mV.
- usually approximated as a sampler (a switch).

Digital to analog Converter (DAC)



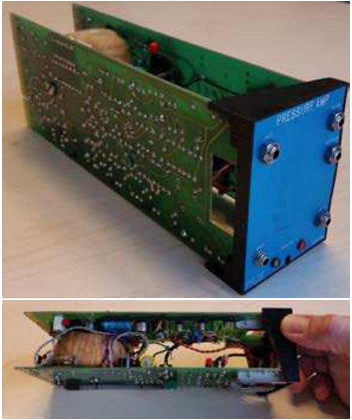
- converts the digital (integer) number calculated by the computer into a voltage so as to drive the output of the plant as desired.
- The voltage ranges and converter resolutions are the same as for the ADC.
- functions as a zero-order hold (**ZOH**), holding its output at a constant value until it receives the next discrete input.

Many microcontrollers incorporate built-in ADC and DAC circuits. These microcontrollers can be connected directly to analog signals.

Motivation

why Discrete-Time Control Systems?

Proportional amplifier (part of a continuous time PID loop for controlling a hydraulic actuator)



Data acquisition PCI card

Microcontroller board

Motivation

why Discrete-Time Control Systems?

Reasons for the prevalence of digital control & signal processing

- **Reliability**: processing digital signals avoids noise and uncertainty that affects analogue signal processing
- **Flexibility**: limited only by processing power and storage
- **Cost**: advances in technology make microcontrollers economical even for small, low cost applications
- **Accuracy**: digital signals usually represented using at least 12 bits

Discrete-Time Control System

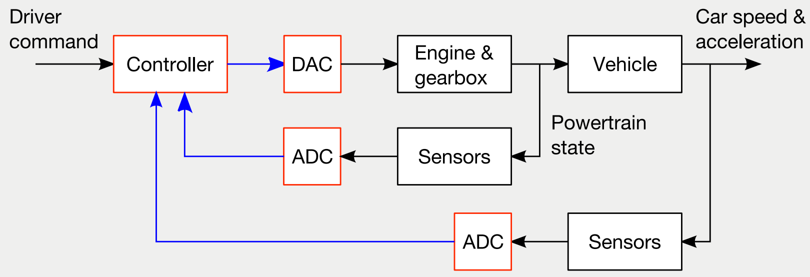
Engine control unit (ECU)

Microprocessor regulating fuel, engine timing, gearbox, brake operation

- ECU **samples** the outputs of numerous sensors (throttle position, crankshaft position, car acceleration)
- uses a **control algorithm** to compute the required control signals (air and fuel flow-rates, ignition & valve timings)

Discrete-Time Control System

Engine control unit (ECU)

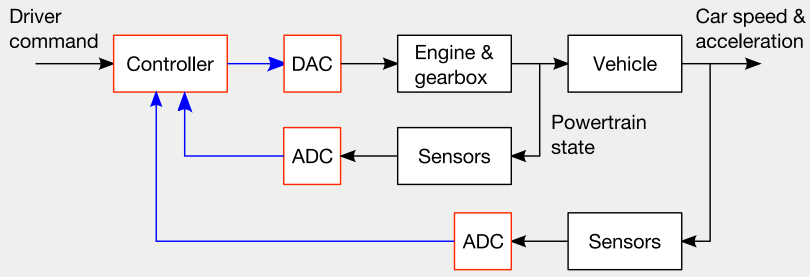


At each sampling instant, the ECU:

1. reads in sensor measurements from ADCs
2. uses these to compute the required control signals
3. outputs control signals to actuators via DACs

Discrete-Time Control System

Engine control unit (ECU)



- controller is a discrete time system (μP) which interacts with continuous time systems (car, driver, etc)
- Either design continuous time controller using a continuous time system model and implement it approximately in discrete time
- Or design a controller directly in discrete time using discrete time models

Computer Control Algorithms

The control algorithm in a computer is implemented as a **program** which runs continuously in a loop.

- computer performs these calculations once every T .
  - synchronization and the computation delays are important.
- programming is done in a high-level computer language (C is often used).
  - Hardware drivers provided by the ADC and DAC manufacturers.

```
initialize system
    repeat
        wait for interrupt
        data input
        compute controller output
        update shift registers
        data output
        terminate task
    until done
shut down system
```

Computer Control Algorithm

- The algorithm, once starts, runs continuously and can only be stopped manually by an operator or if some abnormal condition or event occurs.
- Note that the loop is run exactly at the sampling instants. Two approaches to achieve this:
  1. **timer interrupt**
  2. **ballast coding**

Ballast Coding

- This method is so **simple**. It involves finding the execution time of each instruction inside the loop and then adding **dummy code** to make the loop execution time equal to the required sampling interval.

```
Do Forever
    R = Read (desired value);
    Y = Read (actual plant output);

    E = R - Y;          % Calculate the error signal
    U = CtrAlg (E);  %Calculate controlle output

    SendToDAC(U);
    Add dummy code
    ...
    ...
    Add dummy code
End
```

- **Disadvantage**: if the code inside the loop and/or CPU clock rate is changed, then it will be necessary to readjust the execution timing of the loop.

Timer Interrupts

- A popular way to perform accurate sampling with constant sampling period is to use timer interrupts
  - available on most microcontrollers.
- The controller algorithm is written inside the timer interrupt service routine (ISR), and the timer is programmed to generate interrupts at regular intervals, equal to the sampling time.
- At the end of the ISR algorithm, control returns to the main program, which either waits for the occurrence of the next interrupt or performs other tasks (e.g. displaying data on an LCD).
  - The use of interrupts has the advantage that the computer can do other tasks between the sampling instants.

Timer Interrupts

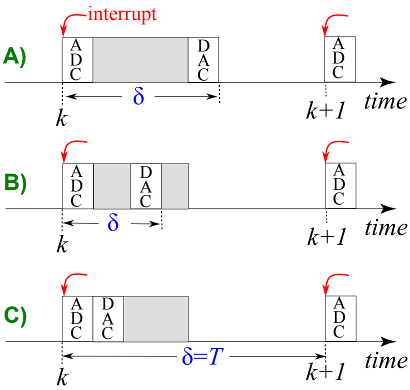
```
MainProgram (){
    % Wait for a timer interrupt
    % (or perform some other tasks)

    End;
}

ISR (){     %interrupt service routine
    R = Read ( desired value );
    Y = Read ( actual plant output );

    % Calculate the error signal
    E = R - Y;
    U = CtrAlg (E);  %Calculate controlle output
    SendToDAC(U);

    return; %from interrupt
}
```



Digital vs. analog control

- The processing speed of computer hardware makes it possible to sample signals at very high speeds (i.e. very small sampling periods).
  - Therefore, digital controllers achieve performance that is essentially the same as that based on continuous monitoring of the controlled variable.
- Digital controller is implemented in **software** and so is **easy to modify**. **Analog** control is difficult to modify once implemented in **hardware**.
- **Complex controller** structures such as adaptive control are easily **realizable** using digital control.
  - This is different from analog control where the structure of the controller is restricted to simple forms such as PID controllers.
- Digital control is **economical** even for small, low-cost applications.

Homework

MATLAB Tutorial

Carefully read and *exercise* on the provided **MATLAB Tutorial** from beginning up to:
B.2.1 Continuous-Time Systems
(pp.285–298)

Administrative Stuff

Mini–Projects

Rules

- Students are organized in groups
- each group will submit
  1. **technical** report (both soft and hard copy)
  2. presentation and demo
  3. hardware and fully commented software

Topics[1]

- Arm Motion Analyzer ★
- 3D Object Tracker ★
- DC Motor Torque Control
- Small Size Robot Arm
- Self Tutor V2

I stronglly recommend each student to have a github account

[1] Other projects can be considered

★ externally funded projects

Thanks for your attention.

Questions?

Assoc. Prof. Dr.Ing.
Mohammed Ahmed
mnahmed@eng.zu.edu.eg
goo.gl/GHZZio

Robotics Research Interest Group (zuR^2IG)
Zagazig University | Faculty of Engineering | Computer and Systems Engineering Department | Zagazig, Egypt