cis1.0
computing: nature, power and limits—robotics applications
fall 2006
lecture # A.1
introduction

topics:

(0) introduction to the course

(1) what is a computer?

(2) what is a robot?

(3) what is a network?

instructor:

- Prof Elizabeth Sklar, *sklar@sci.brooklyn.cuny.edu*

course web page:

- `http://www.sci.brooklyn.cuny.edu/~sklar/cis1.0`

(0) introduction to the course

- about this course
  - parallel course to the new "lower tier core" cc3.12
  - uses **robotics** as a *context* (i.e., the basis for examples and some of the lab exercises)
  - this deviation means we have to call the course **cis1.0** instead of cc3.12, but you will get the same credit and can count this for your core requirement in the same way
- topics covered:

(A) Introduction to Computers and Networks
(B) Algorithms and Computer Languages
(C) Machine architecture, Data representation and storage
(D) Event-driven programming
(E) Programmer-defined functions
(F) Solvability and Feasibility
(G) Security, Privacy, Encryption and Plagiarism

(0) course structure

- 7 **units**
- each unit has:
  - one **lecture**
  - two **labs**
  - one **assessment**
- half of the labs will be hands-on sessions using the internet in the library multimedia classroom (room 383 Library)
- the other half will be hands-on sessions using **LEGO Mindstorms robots** (in 3214 N)
- the assessments will be:
  - written, take-home assignments ...OR
  - written, in-class quizzes ...OR
  - oral, in-class presentations ...OR
  - oral, in-class demonstrations
- your grade = 7 assessments (10% each) + final exam (30%)

(1) what is a computer?

- a device that can process data, store data and execute instructions
  - what is the difference between a computer and a calculator?
  - what everyday devices have computers inside them?

(1) hardware components of a computer

- processor (i.e., central processing unit, or CPU)
- memory
  - short-term: RAM (random access memory), goes away when you turn off the computer
  - long-term: permanent storage media, like a hard disk, USB drive, CD
- input devices
- output devices
- peripherals




(1) software components of a computer

- operating system
  - Microsoft Windows
  - Mac OS-X
  - Linux
  - UNIX
- applications
  - email (Outlook, MacMail, Eudora, pine, ...)
  - browser (Firefox, Internet Explorer, Safari, ...)
  - music (iTunes)
  - office tools (word processing, spreadsheets, presentations)
  - calculator
- drivers
  - printer
  - scanner

(1) how does software work?

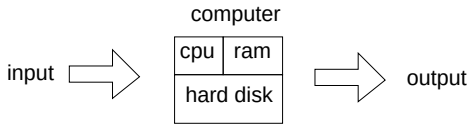
- a human writes instructions for the computer in a language that the computer can understand
  - low-level languages (e.g., assembly)
  - high-level languages (e.g., Java)
- high-level languages are **compiled** (translated) into **binary machine code**, i.e., a language that the computer's processor can understand
- instructions must be very specific!
- instructions are grouped into **programs**
- instructions are executed **sequentially** (one after another)
- what can go wrong?
  - user or "operator" errors
  - program errors: called **bugs**
  - hardware errors (or "faults")

(1) how are programs written?

- programs are written in high-level languages using a **text editor** (e.g., NotePad or TextEdit)
- this is different from a **word processor** (e.g., Microsoft Word), which stores extra formatting characters (besides what you see on the screen...)
- the **programmer** invokes a **compiler** to translate the program into code that the computer can execute
- the **user** runs the executable program
- the programmer's code and the executable program are stored on the computer's hard disk in **files**

(1) what is a computer again?

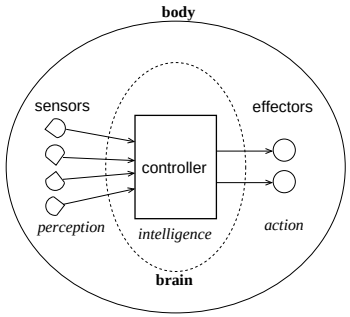
- a device that receives **input** from a human or another computer or another device, **processes** that input and produces **output**



- a computer **program** is what does the processing
- the program is stored on the computer's hard disk, and when the program runs, it is copied into the computer's memory (RAM) and the instructions contained in the program are executed by the computer's central processing unit (CPU)
  - it's like reading a book... you get the book from the shelf where it is stored (which is like fetching the program from the computer's hard drive), you open the book (which is like starting the program) and you read it, one word at a time (which is like running the program, one instruction at a time)

(2) what is a robot?

- *robot = autonomous embodied agent*
- has a *body* and a *brain* (a COMPUTER!)
- exists in the physical world (rather than the virtual or simulated world)
- is a mechanical device
- contains *sensors* to perceive its own state
- contains *sensors* to perceive its surrounding environment
- possesses *effectors* which perform actions
- has a *controller* which takes input from the sensors, makes *intelligent* decisions about actions to take, and effects those actions by sending commands to motors

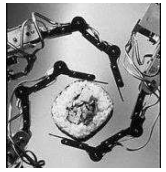


(2) a bit of robot history

- the word *robot* came from the Czech word *robota*, which means *slave*
- used first by playwrite Karel Capek, "Rossum's Universal Robots" (1923)
- human-like automated devices date as far back as ancient Greece
- modern view of a robot stems from science fiction literature
- foremost author: Isaac Asimov, "I, Robot" (1950)
- the *Three Laws of Robotics*
  1. A robot may not injure a human being, or, through inaction, allow a human being to come to harm.
  2. A robot must obey the orders given it by human beings except where such orders would conflict with the First Law.
  3. A robot must protect its own existence as long as such protection does not conflict with the First or Second Law.
- Hollywood broke these rules: e.g., "The Terminator" (1984)

(2) effectors

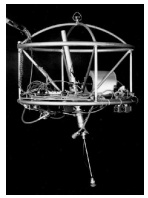
- comprises all the mechanisms through which a robot can *effect* changes on itself or its environment
- *actuator* = the actual mechanism that enables the effector to execute an action; converts software commands into physical motion
- types:
  - arm
  - leg
  - wheel
  - gripper
- categories:
  - *manipulator*
  - *mobile*



some manipulator robots

(2) mobile robots

- classified by manner of locomotion:
  - *wheeled*
  - *legged*
- stability is important
  - *static stability*
  - *dynamic stability*

(2) sensors

- convert physical properties into electronic signals which can be interpreted by the robot's brain (computer) in a useful way

property being sensed	type of sensor
contact	bump, switch
distance	ultrasound, radar, infra red (IR)
light level	photo cell, camera
sound level	microphone
smell	chemical
temperature	thermal
inclination	gyroscope
rotation	encoder
pressure	pressure gauge
altitude	altimeter

(2) autonomy

- automony = no remote control!
- the robot has to "think" by itself
- to be truly autonomous, a system must be able to accomplish *goals* and *solve problems*
- control architectures
  - *deliberative*
    - look-ahead; think, plan, then act
  - *reactive*
    - don't think, don't look ahead, just react!
  - *hybrid*
    - think but still act quickly
  - *behavior-based*
    - distribute thinking over acting

(2) the robots for our labs

- LEGO Mindstorms
- Hitachi h8300 microprocessor (computer) called **RCX**
- with an IR (infra-red) transceiver

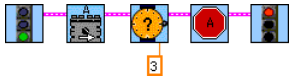
- and 3 input ports, for:
  - light sensor
  - touch sensor

- and 3 output ports, for:
  - motors
  - light bulbs

(2) programming the LEGO Mindstorms

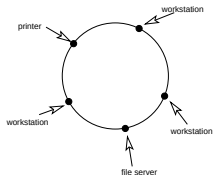
- you write programs on a computer and *download* them to the RCX using an IR transmitter ("communication tower")

- we will use **RoboLab** — a graphical programming environment



what is a network?

- when computers talk to each other, this is called a **network**
- the network can have different kinds of computers and peripherals attached to it
- the way in which the computers are connected to each other is called the network's **topology**



- networks in which computers are physically connected to each other in the close geographical proximity are called **local area networks** (LANs)
- other networks are called **wide area networks** (WANs)
- the **internet** is a wide area network

what is the internet?

- history
  - ARPAnet (circa 1971): used "NCP"
  - TCP (1974): hardware independent, open
  - internet was standardized in September 1981
- layered structure (Open System Interconnection (OSI) or "7-layer" model)
  1. application layer (displays data, communicates with lower layers via presentation layer)
  2. presentation layer (converts application layer data to forms understandable by other layers, and back; translates the "meaning" of the bits)
  3. session layer (exchange of data between applications – "dialog" – and synchronization between applications)
  4. transport layer (transfer of data through network; effects flow control; provides some error recovery)
  5. network layer (physical routing of data from one computer to another; facilitates sender finding receiver)
  6. data link layer (manages transmissions of low-level data; detects and corrects transmission errors)
  7. physical layer (sends electronic signals, or "bits" – 0's and 1's) (usually linked to above)

internet protocols

- *protocol* = set of rules for how computers communicate with each other
- TCP: transmission control protocol (computer ↔ computer)
- UDP: user datagram protocol
- IP: internet protocol
- HTTP: hypertext transfer protocol (computer ↔ browser)
- FTP: file transfer protocol
- SNMP: simple network management protocol (monitors network for problems)
- TFTP: trivial file transfer protocol (fast file xfer, lacks security)
- TCP: can re-transmit if errors
- UDP: user datano error checking, fast messaging
- IP: internet protocol, i.e., moving data via TCP or UDP
- ICMP = internet control message protocol (checks status of computers with other network devices)

some internet facilities

- the world wide web
  - HTML = hypertext markup language
  - *hyperlink*
  - *browser*
  - *web page*, *web site*, *web server*
- ftp (file transfer protocol)
  - *download*
  - *upload*
- email
- newsgroups
  - *posting*
  - *thread*
- mailing lists

internet addresses

- *IP address* = Internet Protocol address
- every computer on the internet has a unique address
- *dotted quad notation* = four numbers separated by dots (.); e.g., 146.245.250.131 (which is the address of the CIS dept web server...)
- address can be stored in 32 bits; there are four formats, depending on the size of the network (i.e., the size of each of the numbers in the dotted quad notation is defined by the format...)
- "subnetting" is a standard (defined in 1985) to divide a large network into a number of smaller networks (this is what a *router* does)
- example:

	network prefix	subnet number	host number
IP address 130.5.5.25 →	130.5.	.5	.25

domain names

- provides a more convenient way to address a computer on the internet than the numeric IP address
- structured hierarchically
- example: `www.sci.brooklyn.cuny.edu`
- common *domain names*: `com`, `edu`, `gov`, `uk` and other country-based domains
- *name server*: maintains correspondance between numeric IP address and domain names;
- DNS = internet domain name system = group of domain name servers

how do robots use networks?

- tele-operation
  - over the internet
  - over LANs
- for...
  - communicating with other robots
  - communicating with computers
  - communicating with humans (who are using computers)
- examples
  - the telegarden
  - Alvin