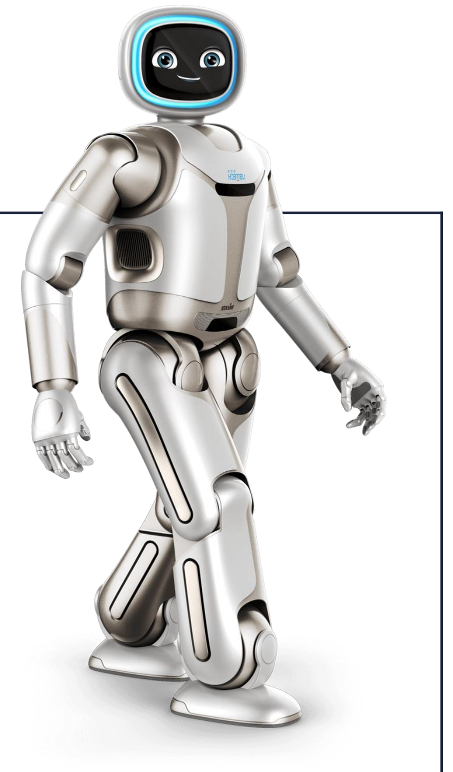
Advanced Robotics

ENGG5402 Spring 2023

Fei Chen

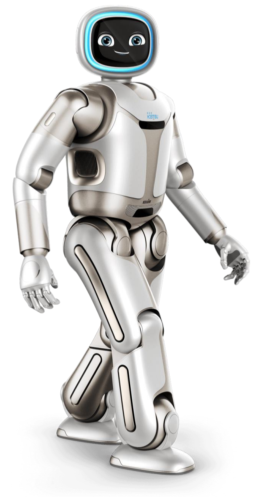


Topics:

- Introduction to Control

Readings:

- Siciliano: Sec. 8.1

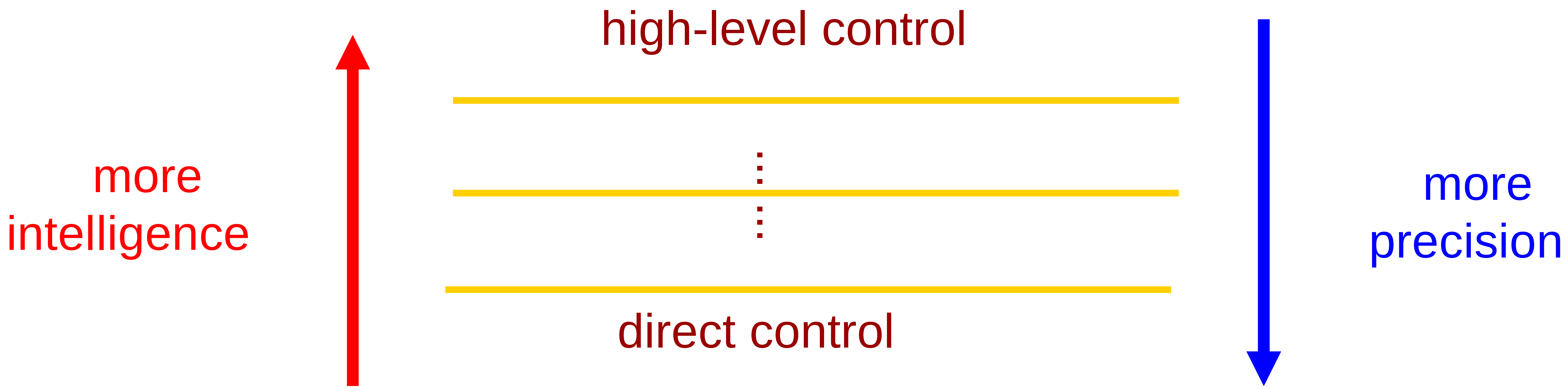


Robot Control

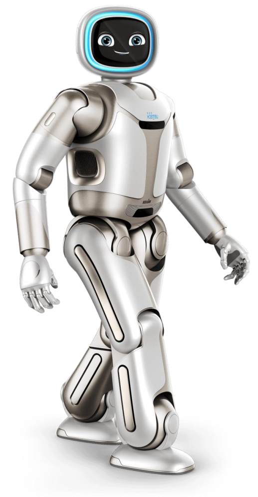
What do we mean by robot control?

- different level of definitions may be given to robot control
  - successfully complete a task or work program
  - accurate execution of a motion trajectory
  - zeroing a positioning error

⇒ control system unit has a hierarchical internal structure



- different but cooperating models, objectives, methods are used at the various control layers



Robot Control Evaluation

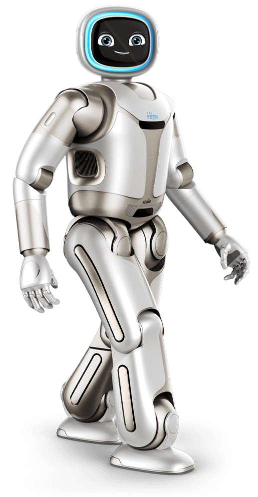
Evaluation of control performance

- quality of execution in nominal conditions
  - velocity/speed of task completion
  - accuracy/repeatability (in static and dynamic terms)
  - energy requirements

  ⇒ improvements also thanks to robot models (software!)
- robustness in perturbed/uncertain conditions
  - adaptation to changing environments
  - high repeatability despite disturbances, changes of parameters, uncertainties, modeling errors

  ⇒ can be improved by a generalized use of feedback, using more sensor information

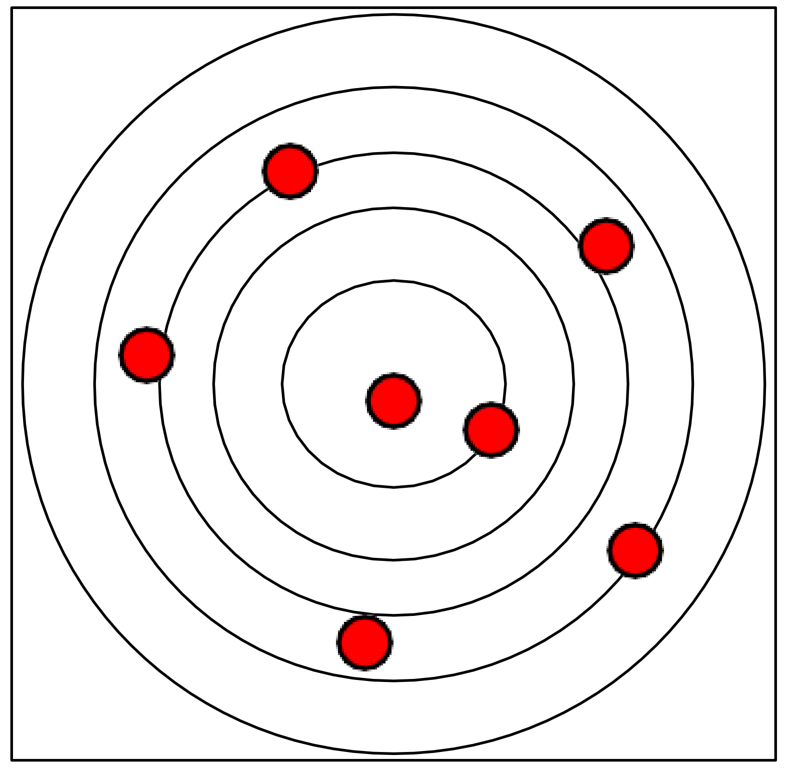
  ⇒ learn through repeated robot trials/human experience/demonstrations (*www.feichenlab.com* for more information)

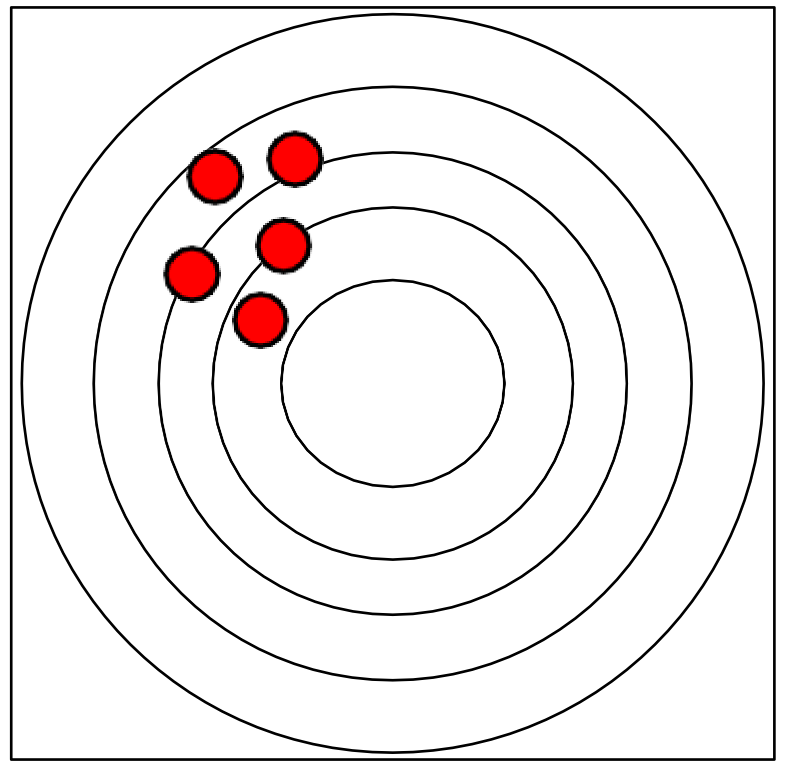


Robot Control Evaluation

Static positioning (accuracy and repeatability)

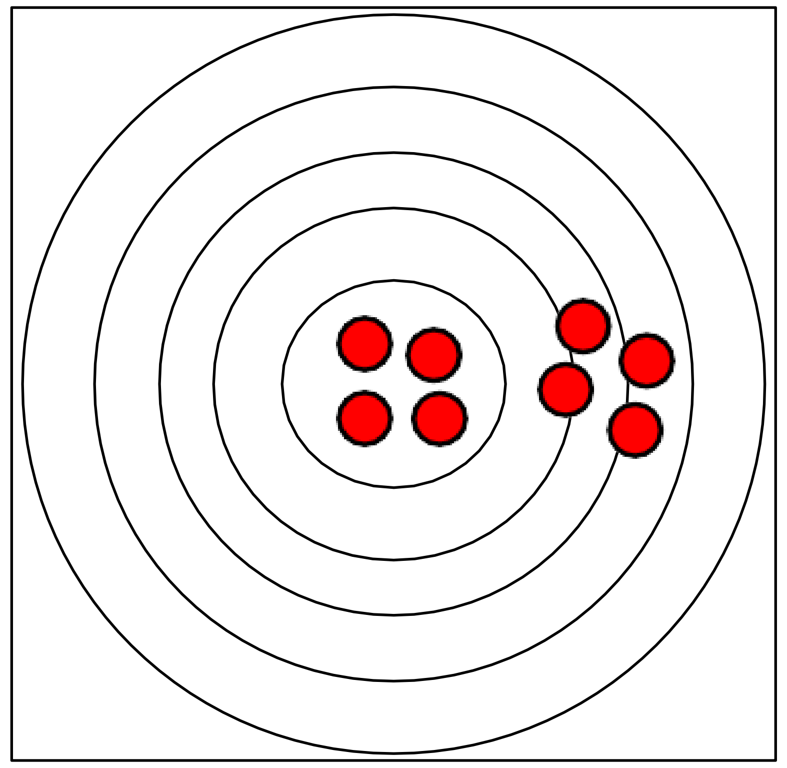
poor accuracy
poor repeatability

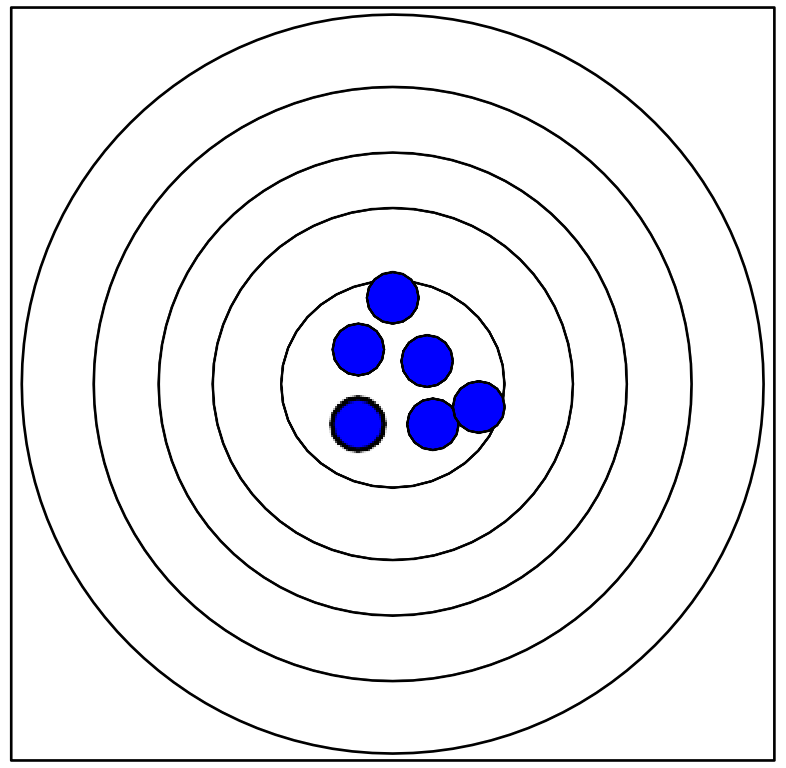




poor accuracy
good repeatability

good accuracy
poor repeatability

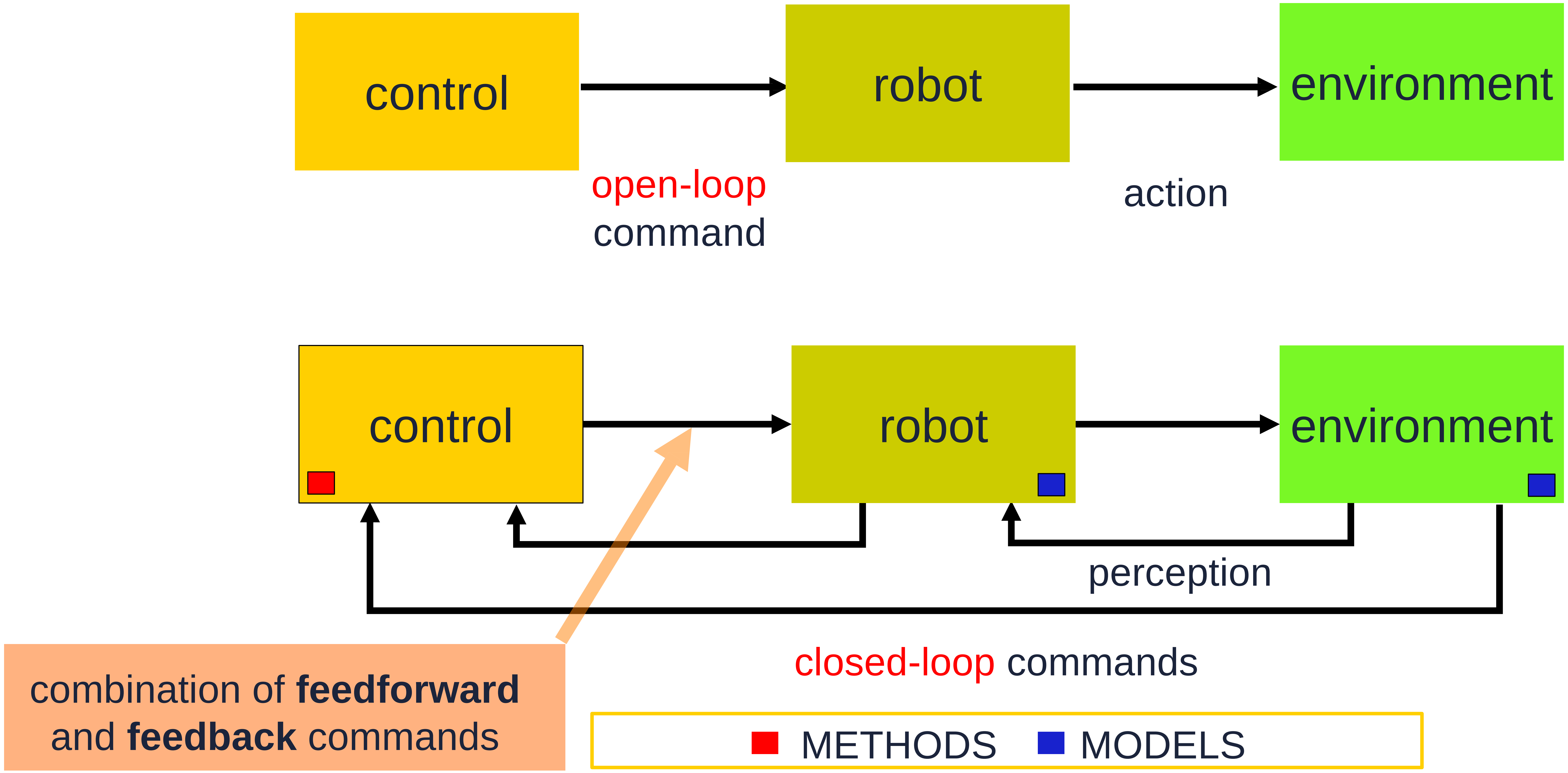


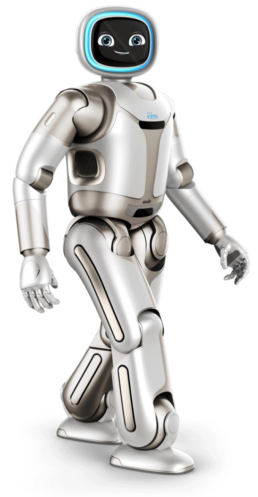


good accuracy
good repeatability

what about “dynamic” accuracy on (test or selected) motion trajectories?

Basic Control Schemes





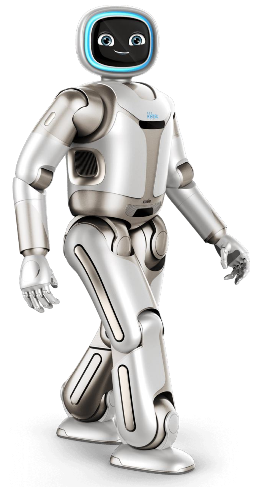
Control Schemes

Control schemes and uncertainty

- feedback control
  - insensitivity to mild disturbances and small variations of parameters
- robust control
  - tolerates relatively large uncertainties of known range
- adaptive control
  - improves performance on line, adapting the control law to a priori unknown range of uncertainties and/or large (but not too fast) parameter variations
- intelligent control
  - performance improved based on experience: LEARNING
  - autonomous change of internal structure for optimizing system behavior: SELF-ORGANIZING

uncertainty on parametric values ➡ IDENTIFICATION

… on the system structure ➡ …



Limits

Limits in control of industrial robots - 1

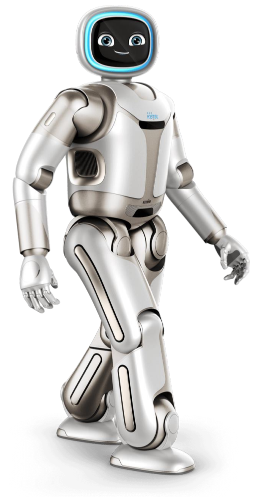
- from a functional viewpoint
  - "closed" control architectures, relatively difficult to interface with external computing systems and sensing devices

  ⇒ especially in applications where hard real-time operation is a must
- at the higher level
  - open-loop task command generation

  ⇒ exteroceptive sensory feedback absent or very loose
- at the intermediate level
  - limited consideration of advanced kinematic and dynamic issues

  ⇒ e.g., singularity robustness: solved on a case-by-case basis

  ⇒ task redundancy: no automatic handling of the extra degrees of freedom of the robot



Limits

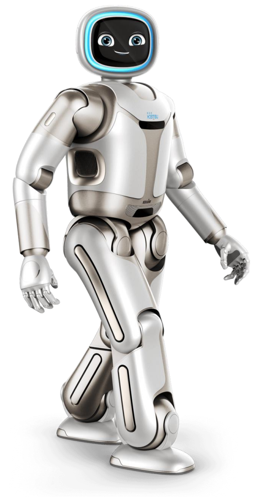
Limits in control of industrial robots - 2

- at the lower (direct) level
  - reduced execution speed ("control bandwidth")
    ⇒ typically heavy mechanical structure ■
  - reduced dynamic accuracy on fast motion trajectories
    ⇒ standard use of kinematic control + PID only ■
  - problems with dry friction and backlash at the joints ■
  - compliance in the robot structure
    ⇒ flexible transmissions ■
    (belts, harmonic drives, long shafts)
    ⇒ large structures or relatively lightweight links ■

now **desired** for safe **physical Human-Robot Interaction**

■ need to include better dynamic models and model-based control laws

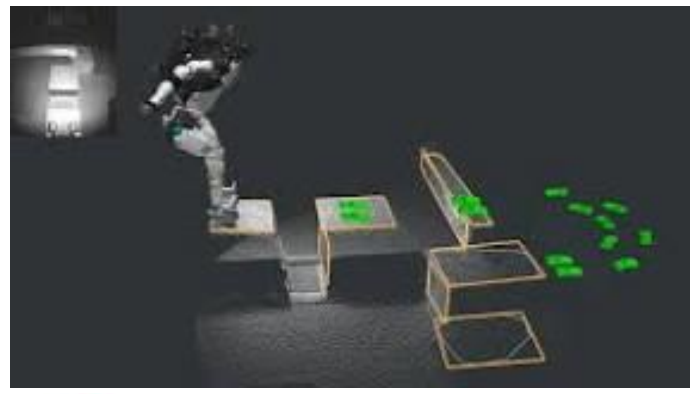
■ handled, e.g., using direct-drive actuators or online friction compensation



Advanced Robot Control Laws

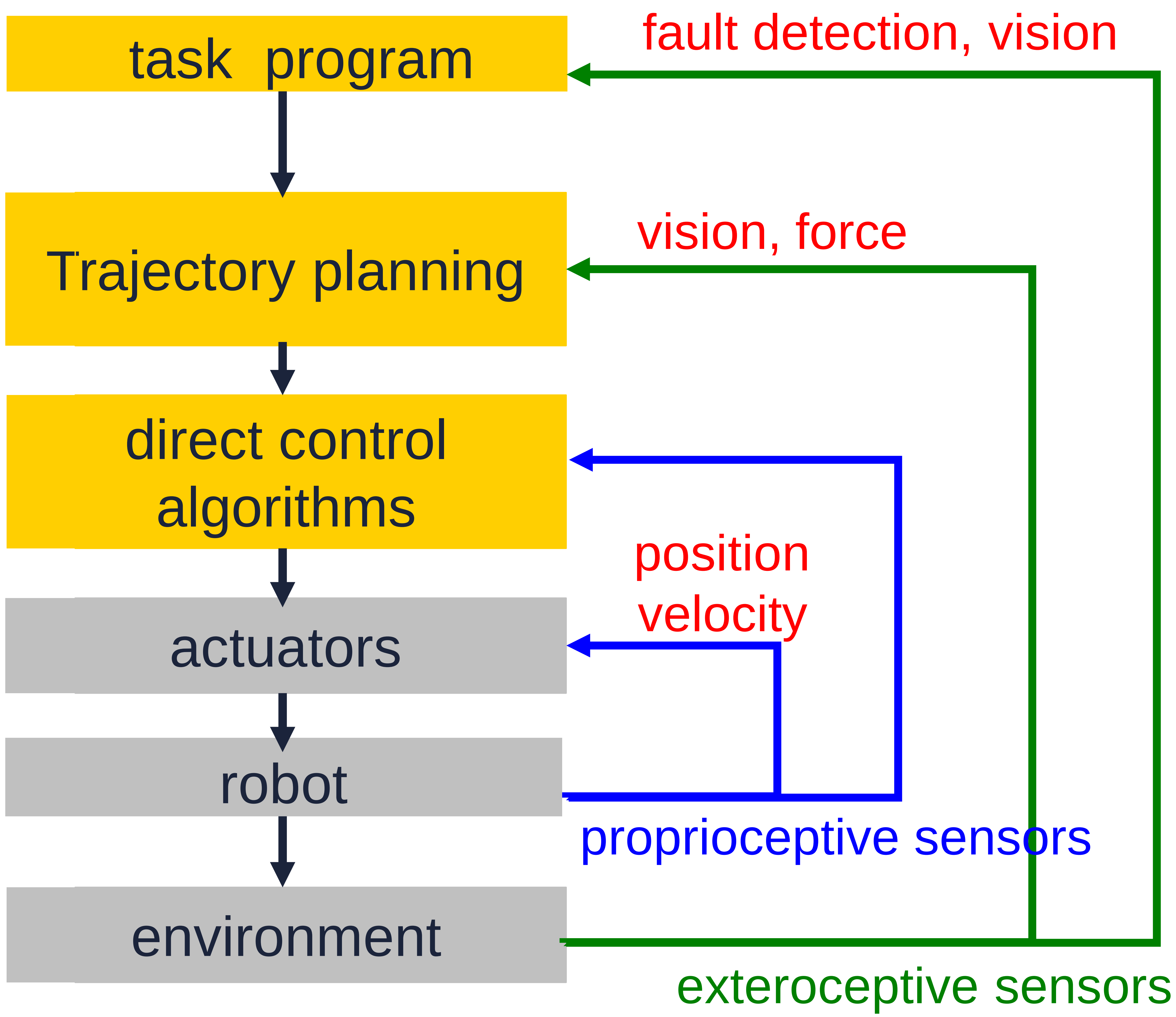
We look for

- deeper mathematical/physical analysis and modeling of robot components (model-based approach)
- schemes using various control loops at different/multiple hierarchical levels (feedback) and with additional sensors
  - visual servoing
  - force/torque sensors for interaction control
  - ...
- “new” methods
  - integration of (open-loop/feedforward) motion planning and feedback control aspects (e.g., sensor-based planning)
    - fast (sensor-based) re-planning
    - model predictive control (with preview)
  - learning (iterative, by imitation, skill transfer, ...)

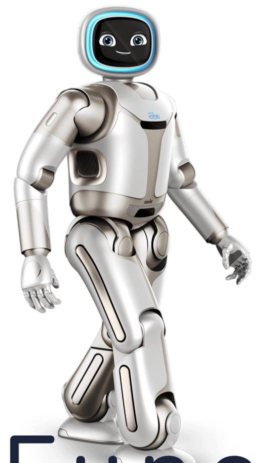


Structure

Functional structure of a control unit (sensor measurements)



SENSORS:
optical encoders, velocity tachos, strain gauges, joint or wrist F/T sensors, tactile sensors, micro-switches, range/depth sensors, laser, CCD cameras, RGB-D cameras
…



Structure

Functional structure of a control unit (programming languages)

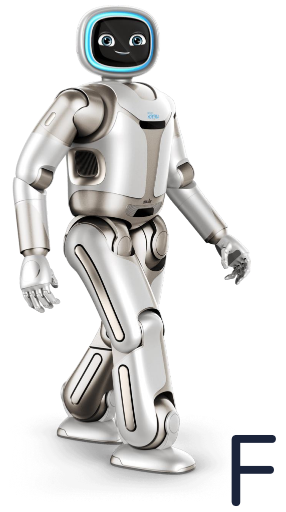
Task program	Java, Lisp, expert-and rule-based systems
↓ Trajectory planning	Matlab, C++, Python
↓ direct control algorithms	Assembler (PICs), C, C++
↓ actuators	
↓ robot	
↓ environment	

dedicated programming languages provided by arm manufacturers

Task-
Object-
Robot-
Oriented

T-O: insert P1 into H5
O-O: move APPR frame #13
R-O: rotate joint 3 by -45°

often "addressed" using the manual TEACH BOX in conventional industrial robots



Structure

Functional structure of a control unit (modeling issues)

Task program	modeling of tasks
trajectory planning	geometric and kinematic models coordinate transformations
direct control algorithms	nonlinear methods dynamic control
actuators	(electrical and mechanical) dynamic models
robot	
environment	structured and unstructured world modeling (and acquisition)

Robot Control/Research software

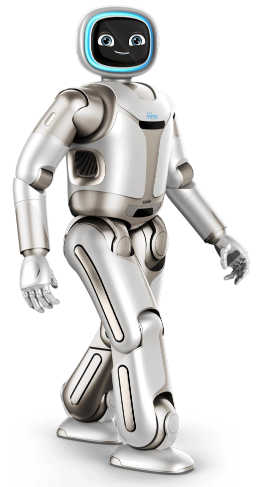
- a (partial) list of open source robot software
  - for simulation and/or real-time control
  - for interfacing with devices and sensors
  - research oriented

Player/Stage playerstage.sourceforge.net ⇒ github.com/rtv/stage ⇒ retired

- Stage: in origin, a networked Linux/MacOS X robotics server serving as abstraction layer to support a variety of hardware ⇒ now a 2(.5)D mobile robot standalone simulation environment
- Gazebo/Pybullet/Isaac: 3D robot simulator (ODE physics engine and OpenGL rendering), now an independent project ⇒ gazebosim.org

CoppeliaSIM (ex VREP; edu version available) www.coppeliarobotics.com

- each object/model controlled via an embedded script, a plugin, a ROS node, a remote API client, or a custom solution
- controllers written in C/C++, Python, Java, Matlab, ...



Robot Control/Research software

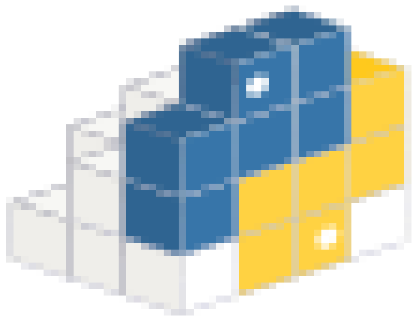
Robot control/research software (cont'd)

Robotics Toolbox (free addition to Matlab) petercorke.com

- study and simulation of kinematics, dynamics, trajectory planning, control, and vision for serial manipulators and beyond ⇒ R2023a

ROS (Robot Operating System) ros.org

- middleware with: hardware abstraction, device drivers, libraries, visualizers, message-passing, package management
- "nodes": executable code (in Python, C++) running with a publish/subscribe communication style
- drivers, tools, state-of-the-art algorithms … (all open source)

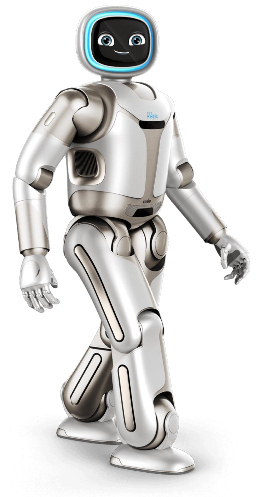


PyRobotics (Python API) pypi.org/project/pyRobotics (v1.8 in 2015) ⇒ retired

MuJoCo is a physics engine that aims to facilitate research and development in robotics, biomechanics, graphics and animation, and other areas where fast and accurate simulation is needed. – acquired by DeepMind!

MuJoCo

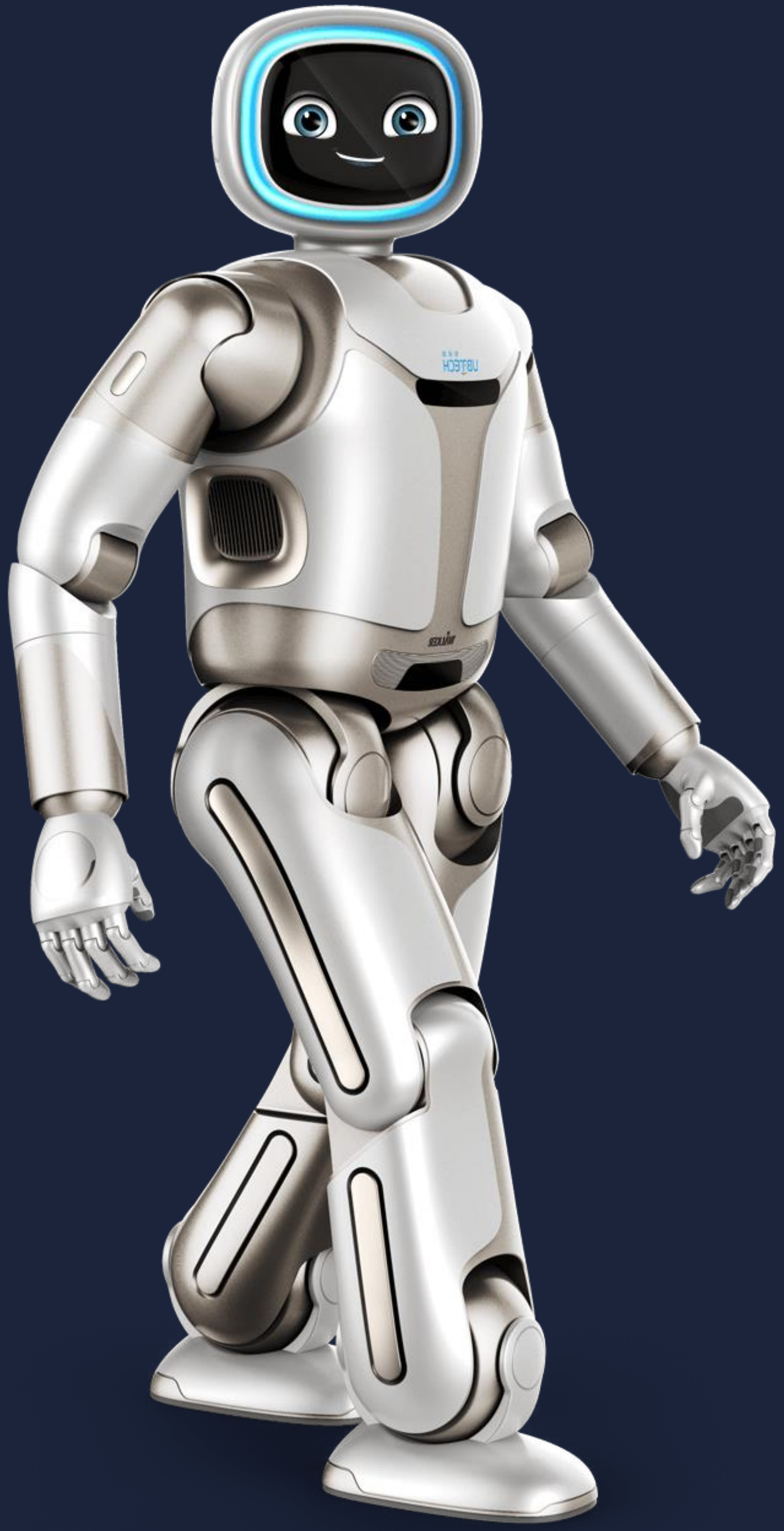
Advanced physics simulation



Summary

- to improve performance of robot controllers
  1. more complete modeling (kinematics and dynamics)
  2. introduction of feedback throughout all hierarchical levels
- dynamic control at low level allows in principle
  1. much higher accuracy on generic motion trajectories
  2. larger velocity in task execution with same accuracy
- interplay between control, mechanics, electronics
  1. able to control accurately also lightweight/compliant robots
  2. full utilization of task-related redundancy
  3. smart mechanical design can reduce control efforts (e.g., closed kinematic chains simplifying robot inertia matrix)
  4. actuators with higher dynamic performance (e.g., direct drives) and/or including controlled variable stiffness

advanced applications should justify additional costs
(e.g., laser cutting with 10g accelerations, safe human-robot interaction)



Q&A