Robotics
-
Industrial Robot Programming

Gaëtan GARCIA
Ecole Centrale de Nantes

Introductory remarks

- There are several meanings to « robot programming »
  - Robot operating system programming.
  - Robot programming language design.
  - **Robot task programming: how to program a robot to perform a given task, using industry standard equipment and languages.**
- We will concentrate on the latter.

Objectives

- See different types of applications and their characteristics.
- Understand that different applications require different programming tools.
- Grasp the strengths and weaknesses of current industrial robots.

Objectives

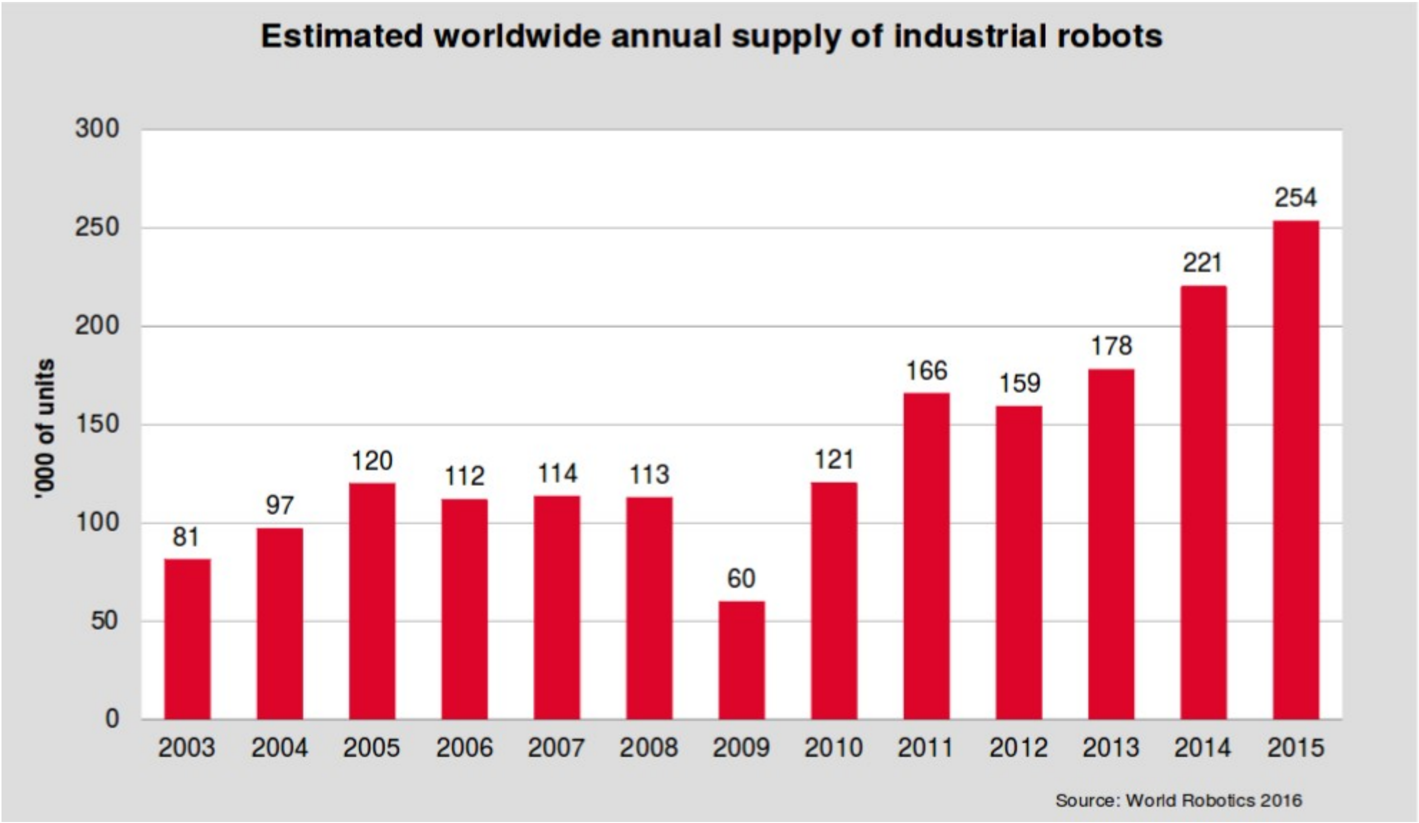
- Illustrate the role of sensors to overcome some robot weaknesses.
- Learn a state of the art robot language and be able to quickly use other robot programming languages.

Part I

Some statistics from the IFR

http://www.ifr.org

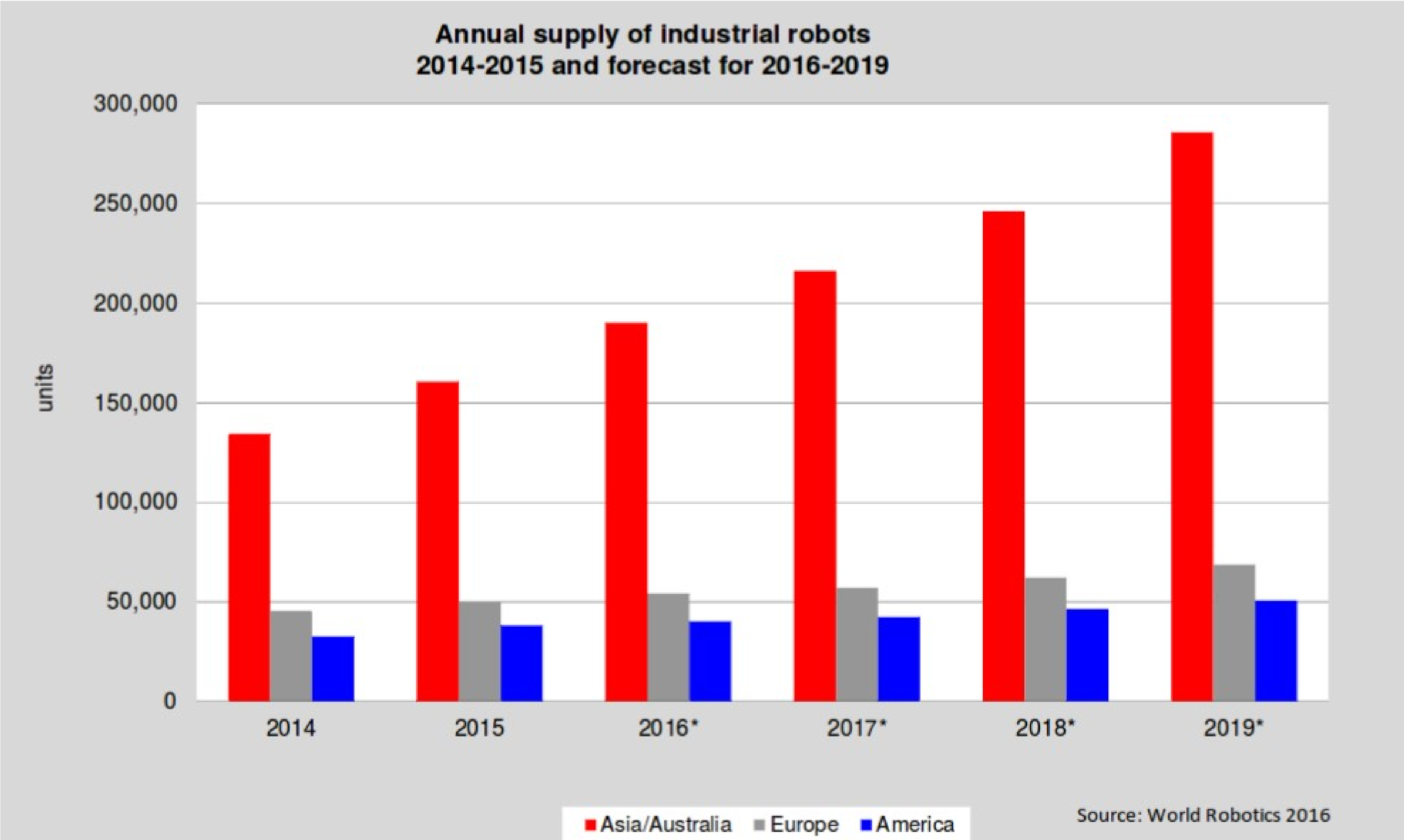
IFR statistics



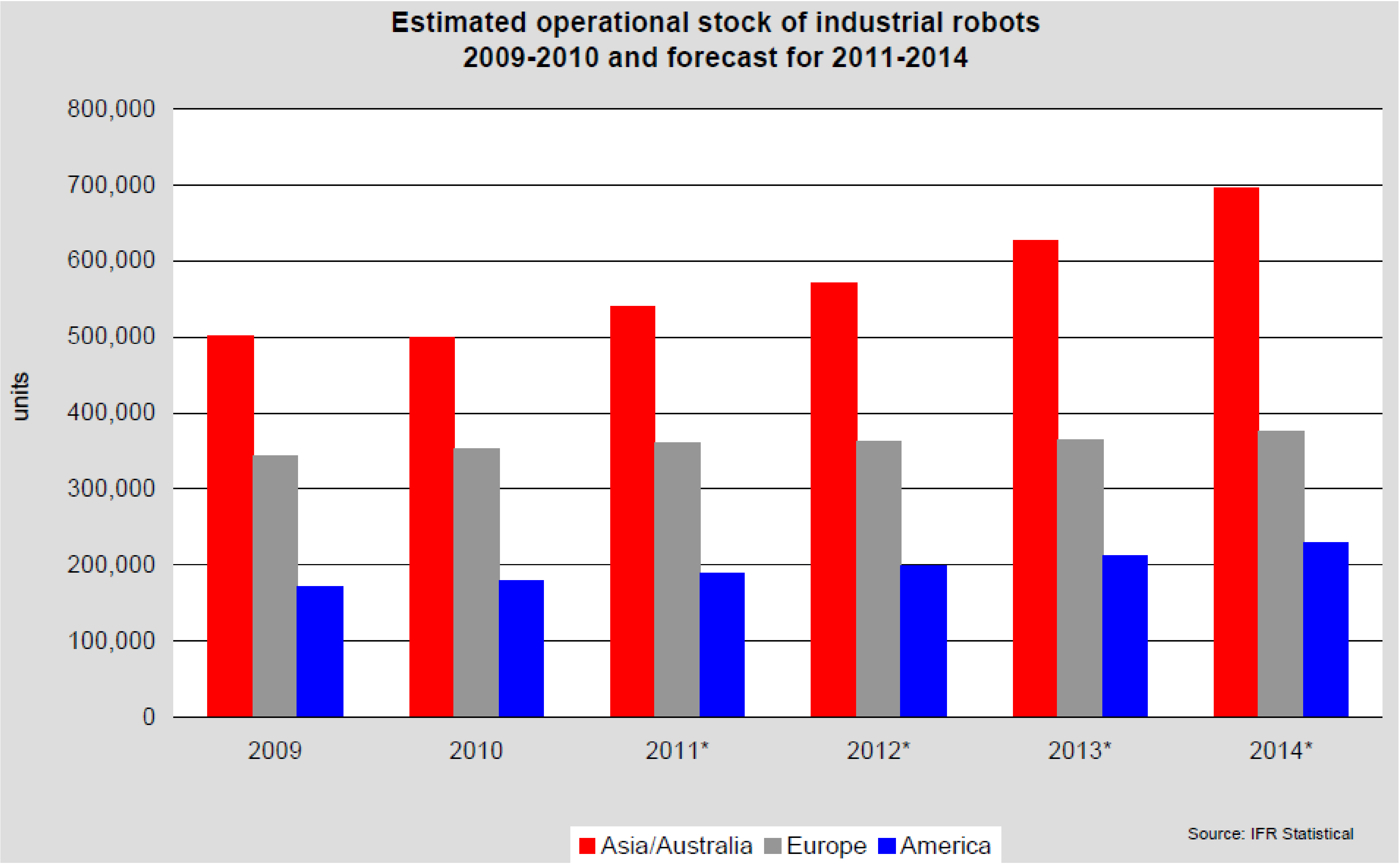
The last three years were all time records

Worldwide stock of operational industrial robots in 2015: 1.600.000

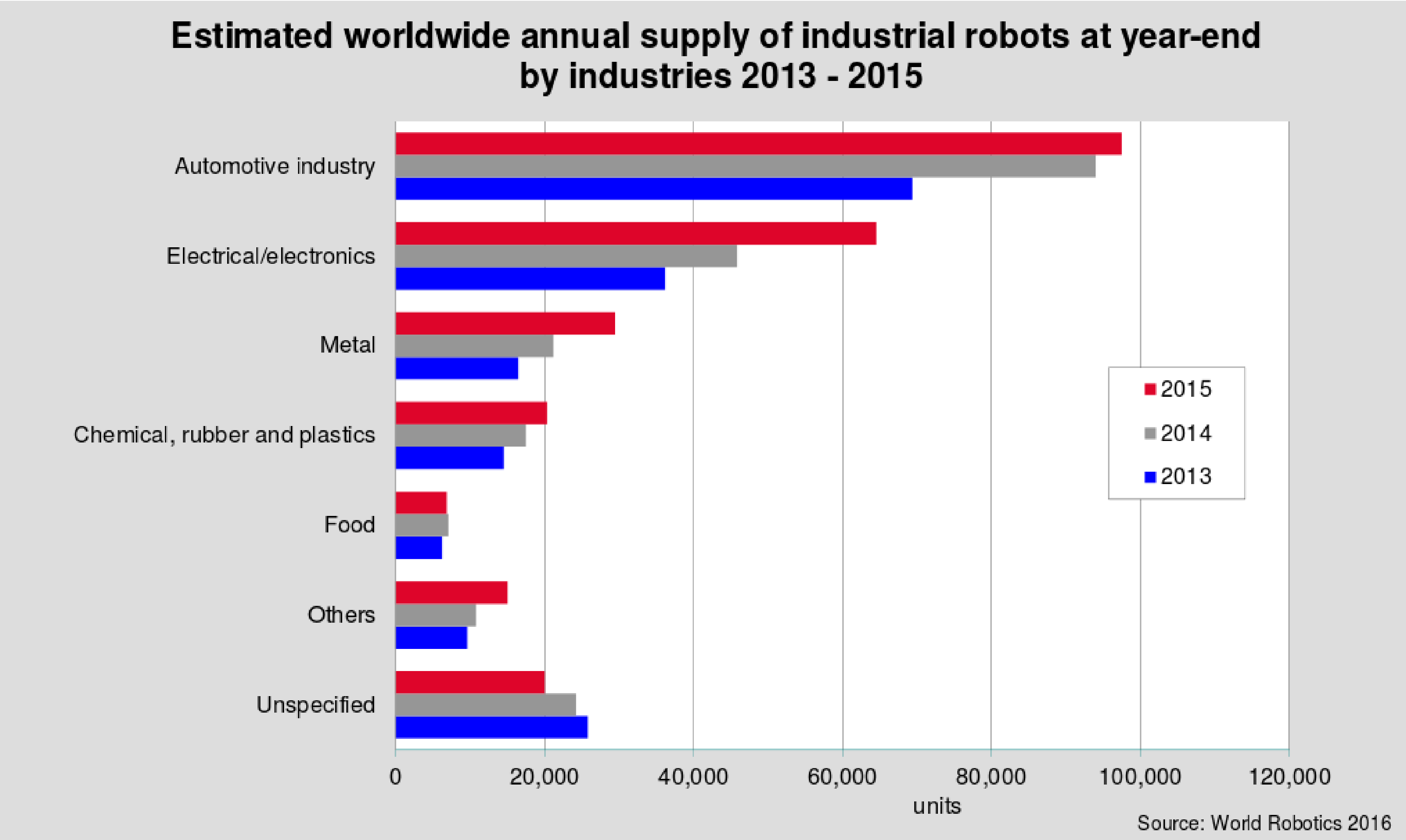
IFR statistics



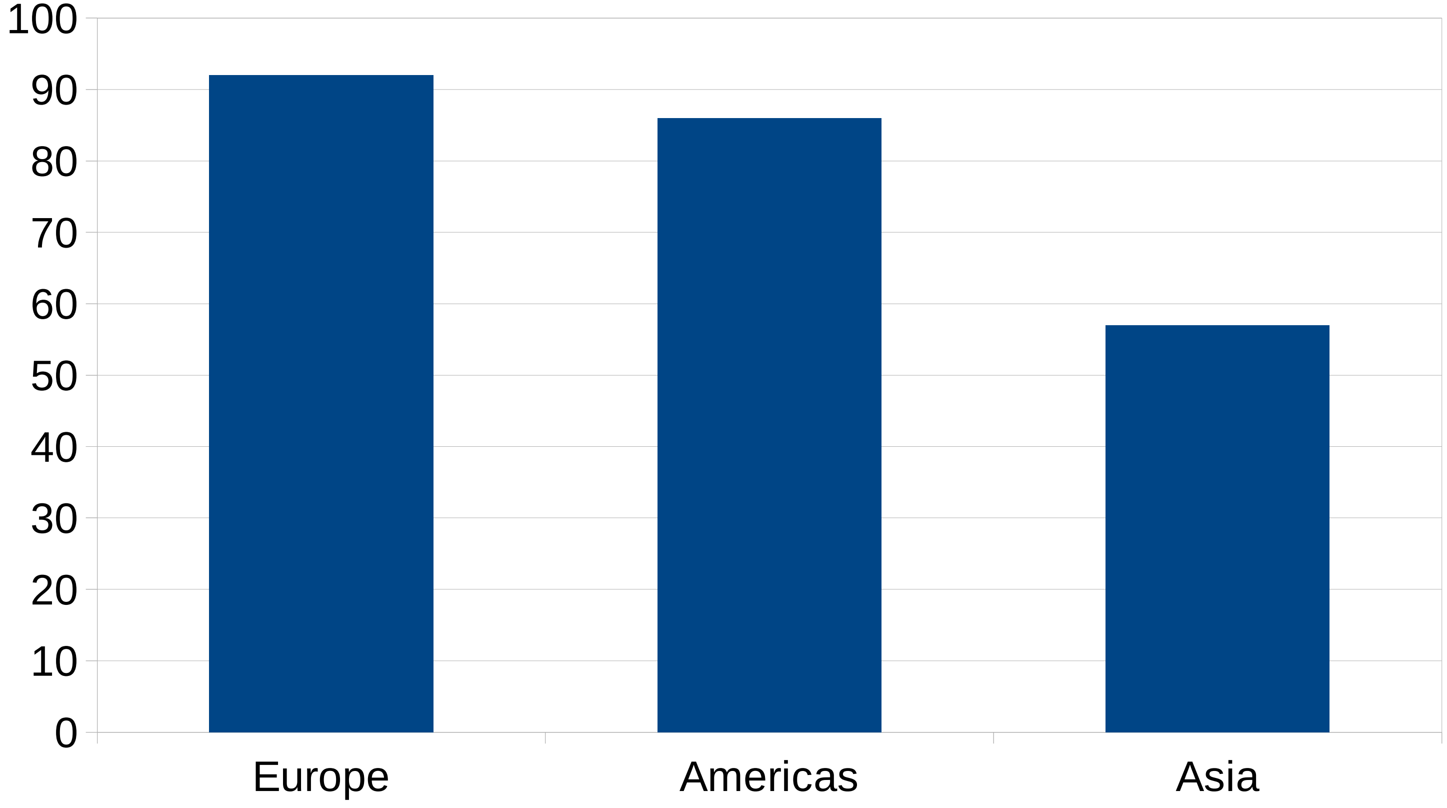
IFR statistics



IFR statistics

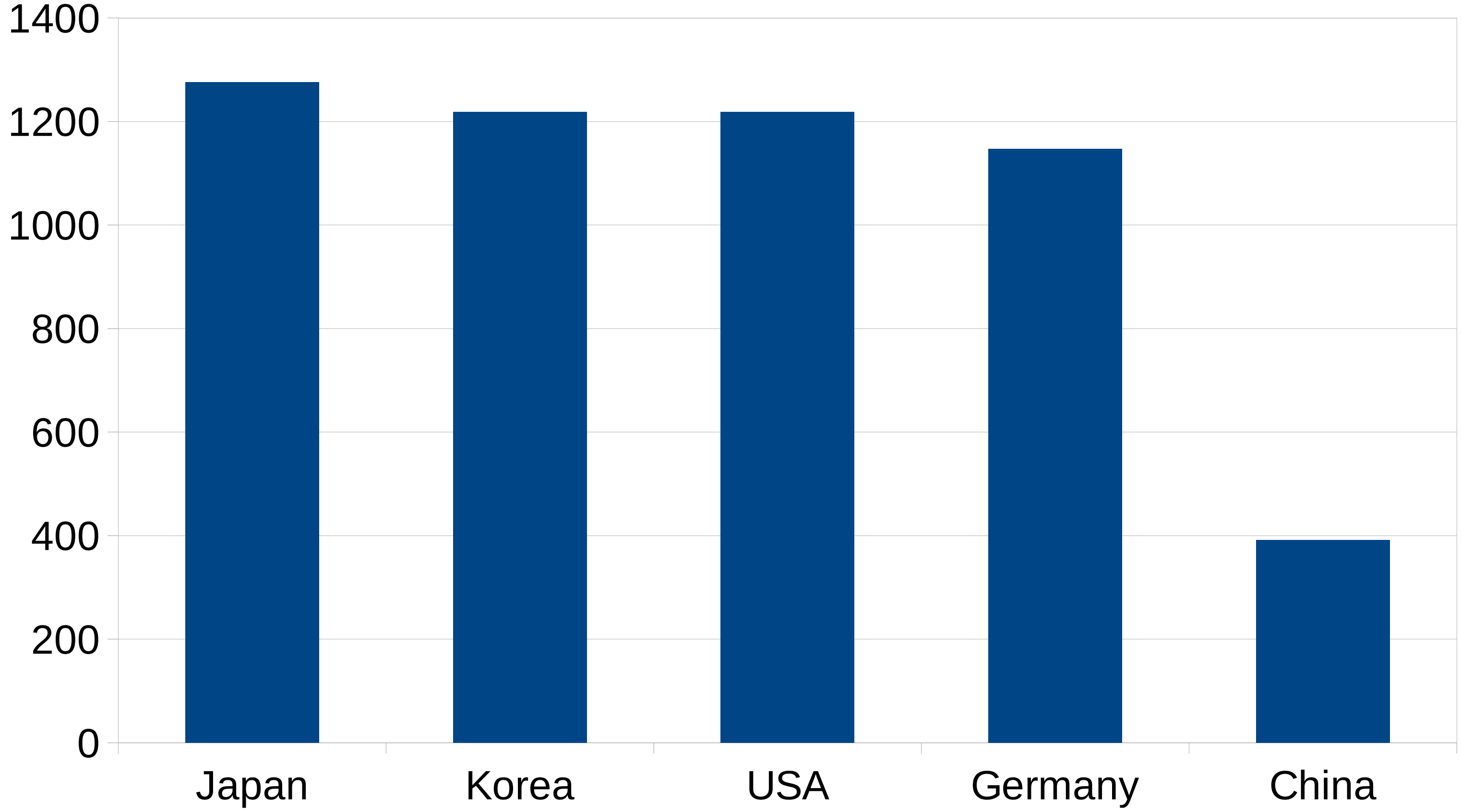


IFR statistics



Robot density in the industry for 10,000 workers

IFR statistics



Robot density in the **automotive** industry for 10,000 workers

Part II

Application examples

and what they tell us about
the variety of robot programming problems

Why robots are attractive for automating tasks

The well-known reasons

- Reduce labor costs
- Withstand repetitive tasks with high speed and constant quality.
- Can cope with tasks that humans cannot perform.
- Improve safety.
- Available at a comparatively low cost.
- …

Other reasons we are interested in

- A demand of greater flexibility:
  – Robots are more flexible than fixed automation.
  – Development of limited series production.
  – Shorter product life cycle.
- Robots can be reprogrammed and redeployed many times.
- Robots can have rapid payback.
- Shorter time to market.
- « Automate or evaporate »…

Robots are « universal » …

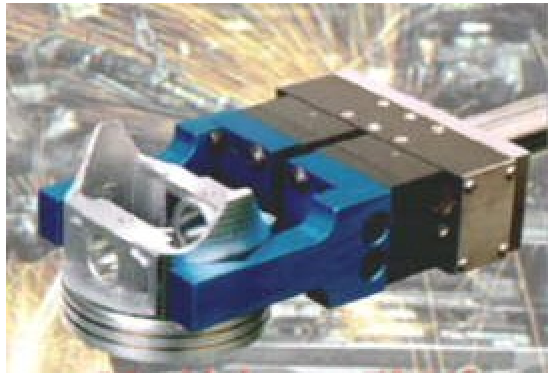
- Universal: applicable everywhere or in all cases.
  Machinery: noting any of various machines, tools, or devices widely adaptable in position, range of use, etc.
- A robot is a universal tool:
  - Its kinematic structure allows:
    - A variety of trajectories
    - Access to a large workspace in position and orientation.
  - It is *a priori* adapted to many tasks.

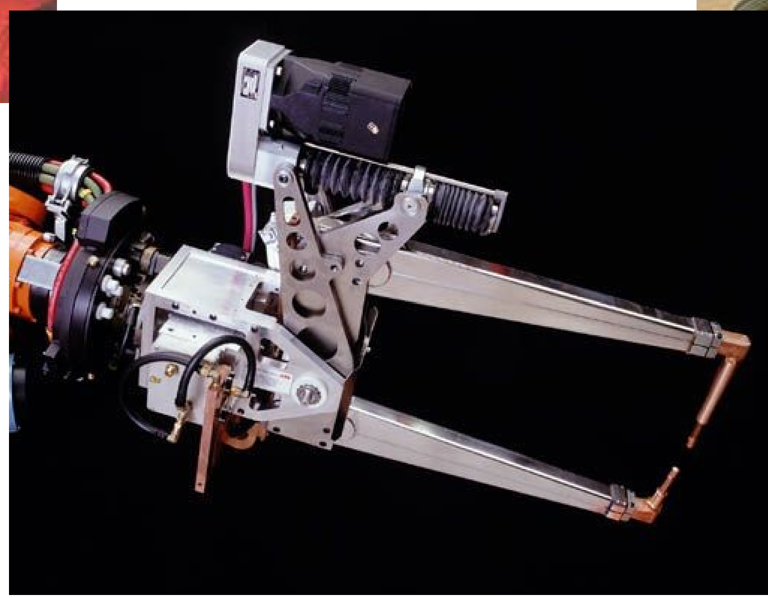
… But not that flexible!

- Robots are more flexible than fixed automation, but …
- … not as much as one would like. Improvements are still necessary in:
  - Reconfiguring for a new task.
  - Reprogramming for a new task.

Examples show that problems are often due to end effectors and insufficient sensory capabilities.

End effector examples





We are yet to create a tool as flexible as the human hand …

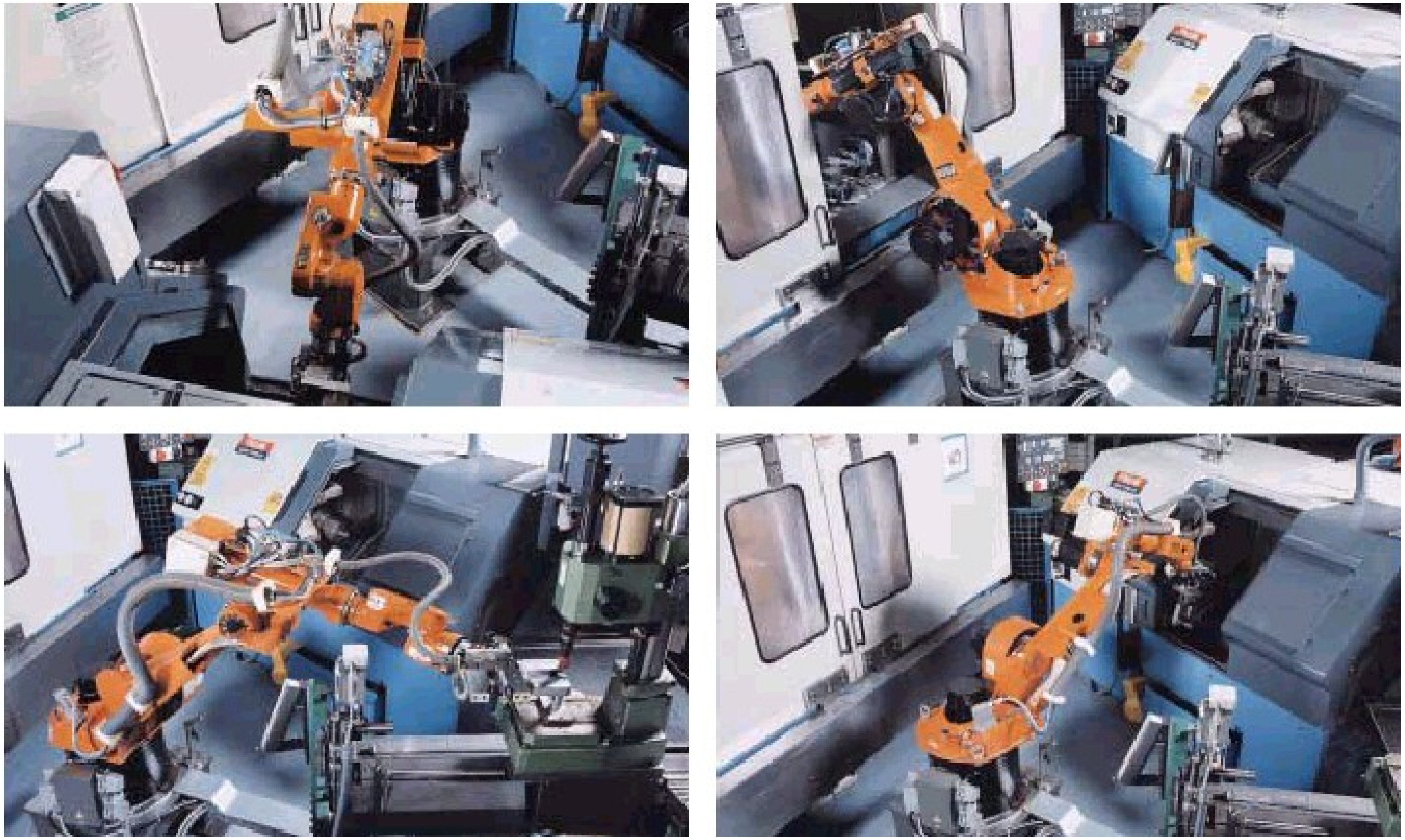
End effector examples

- 2- and 3-finger parallel grippers + dual double gripper: https://www.youtube.com/watch?v=KItgcTCjK34
- Bag gripper:
  https://www.youtube.com/watch?v=nxDN7xzlM1U
- Car body gripper (manipulates only two types!):
  https://www.youtube.com/watch?v=R0-8zysCe30
- Spot welding gun:
  https://www.youtube.com/watch?v=Jx6SjiJUhUA
- Grinding tool (tool controls applied force, not robot)
  https://www.youtube.com/watch?v=NQt2eEu6Y6M

CENTRALE NANTES

Examples of simple applications of robots - Perfectly repeatable tasks

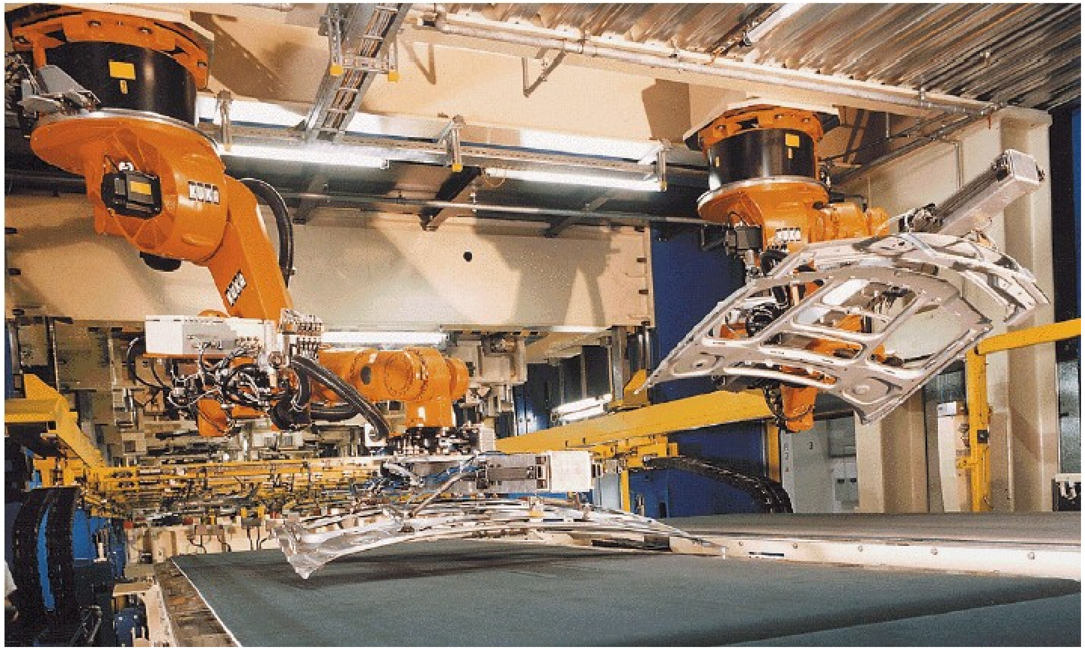
Machine tool servicing



https://www.youtube.com/watch?v=c8WNUC2etCw

In this example, observe the pallets (will be discussed later)

Unloading machines

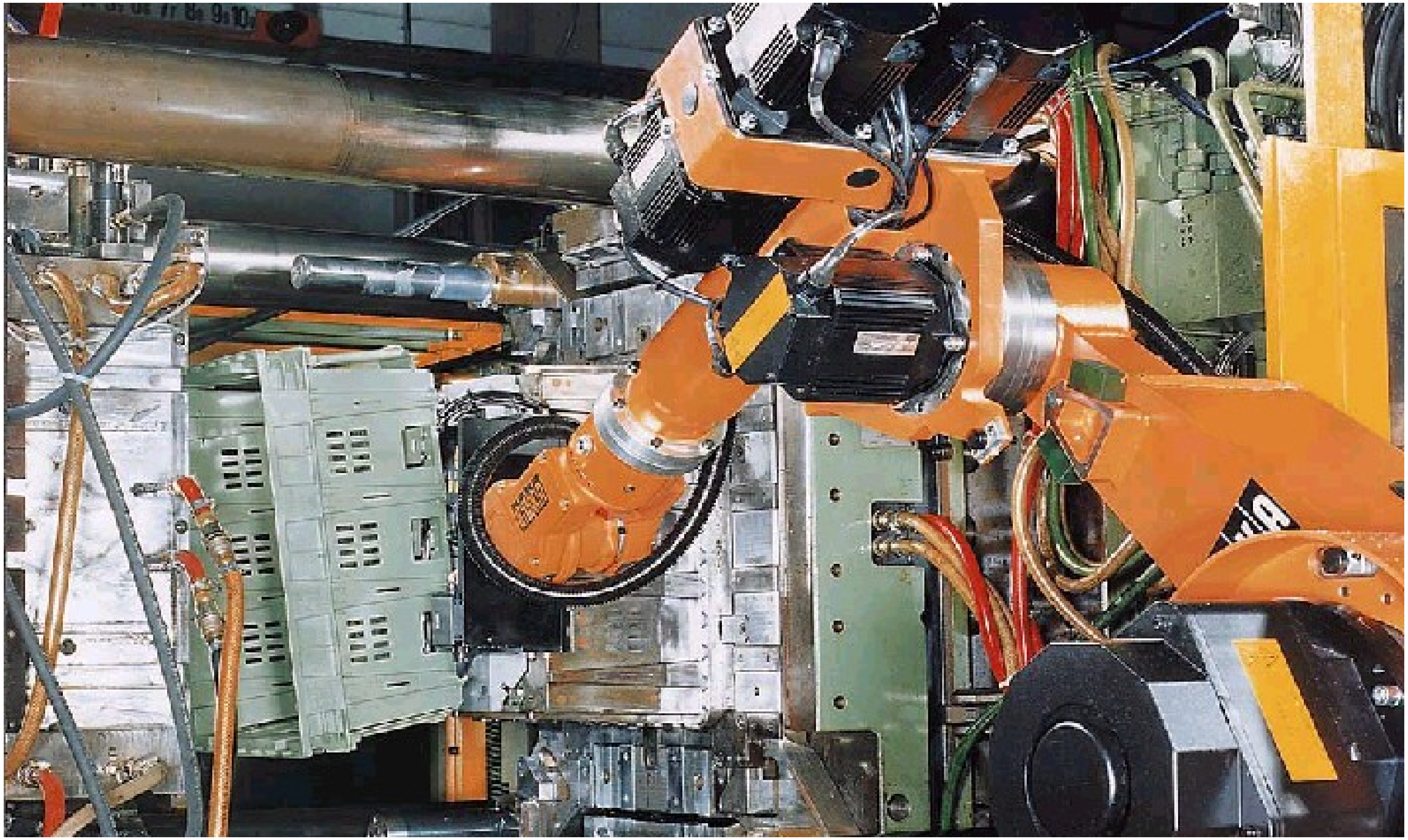


Note that end effectors must be adapted when parts change

https://www.youtube.com/watch?v=TiQJdPfLPJM

Notes: Extremely noisy environment. Short cycle time. Robot in the background takes a part from one press to put it in the next for further processing (presses organized in a line).

Unloading a plastic moulding machine



https://www.youtube.com/watch?v=c07dZP2cGRQ

Characteristics of these applications

- The work locations are the same at each task cycle: the machines do not move.
- To perform the task, the robot merely needs to be able to come back to the same locations at each cycle (requires a certain ***repeatability***).
- The tasks are defined by few locations.

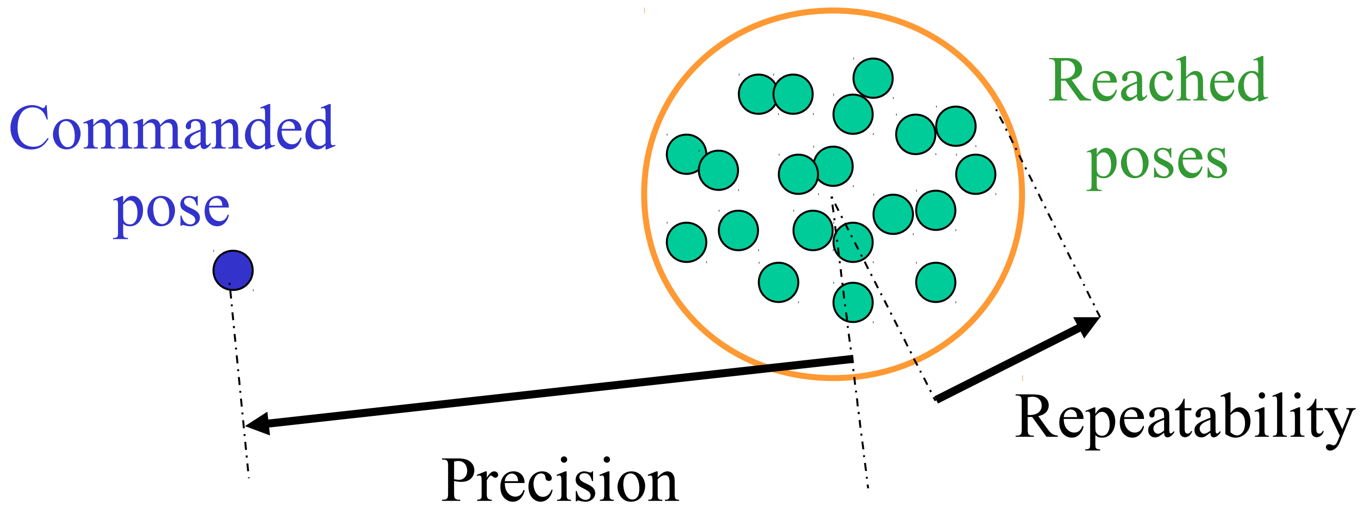
Basically, such tasks can be performed by playing back pre-recorded locations

(more on this later)

Side note…

Precision vs. repeatability

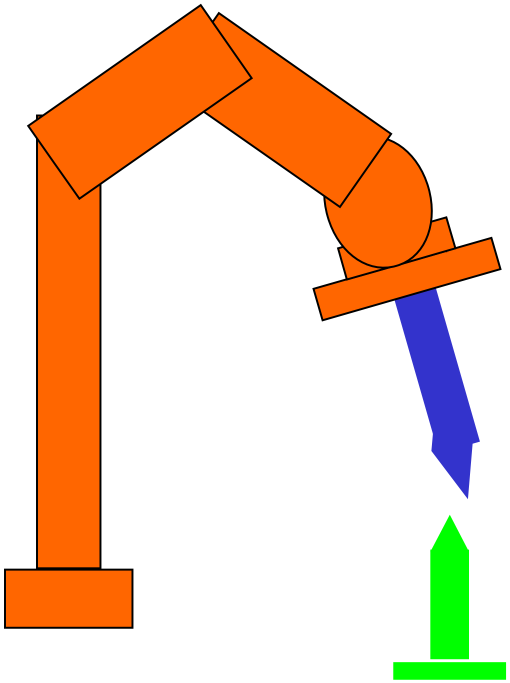
Make a robot move several times to a point (commanded pose) and measure the locations (reached poses).



- Precision: distance between commanded and reached poses.
- Repeatability: agreement between reached poses.

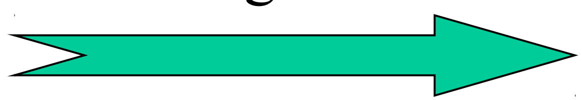
Industrial robots have excellent repeatability but much poorer precision

Side note… How to check that robots are not precise

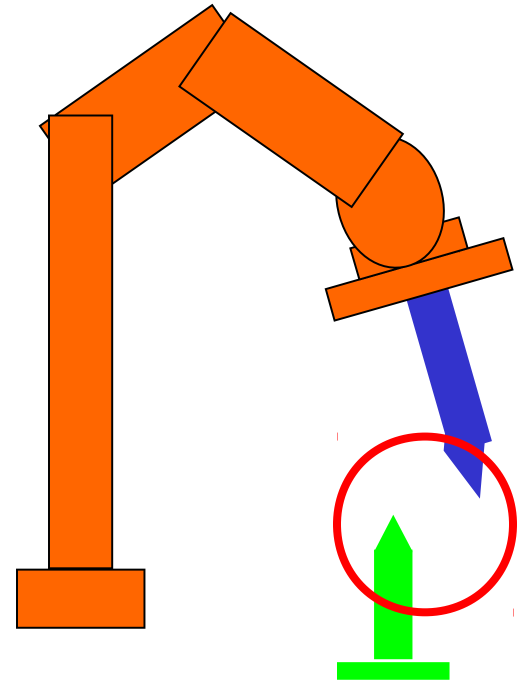


Fixed point

Move to the same location A, but in left shoulder configuration:



Here A
Lefty
Move A



Fixed point

Record robot location in right shoulder configuration as a cartesian point A

The robot should reach the exact same point, but does not !

Side note…

How to check robot precision

- Changing shoulder configuration is the worst case.
- Order of magnitude of the error: in mm.
- This does not **measure** robot precision. But if the error is too large, then the robot definitely needs **calibration**.
- Other notions of precision are also of interest (*e.g.* in a local frame, without configuration changes). More on this later…

How to teach locations to a robot

What a recorded location is

- A ***joint*** location: vector of (generally six) joint coordinates (angles, lengths…)

For most robots, it is also possible to use:

- A ***cartesian*** location (requires Direct and Inverse Geometric Models):
  - 3 ***position*** coordinates in a given frame (the robot base frame is the default)
  - 3 ***orientation*** coordinates (wide variety of orientation representations)

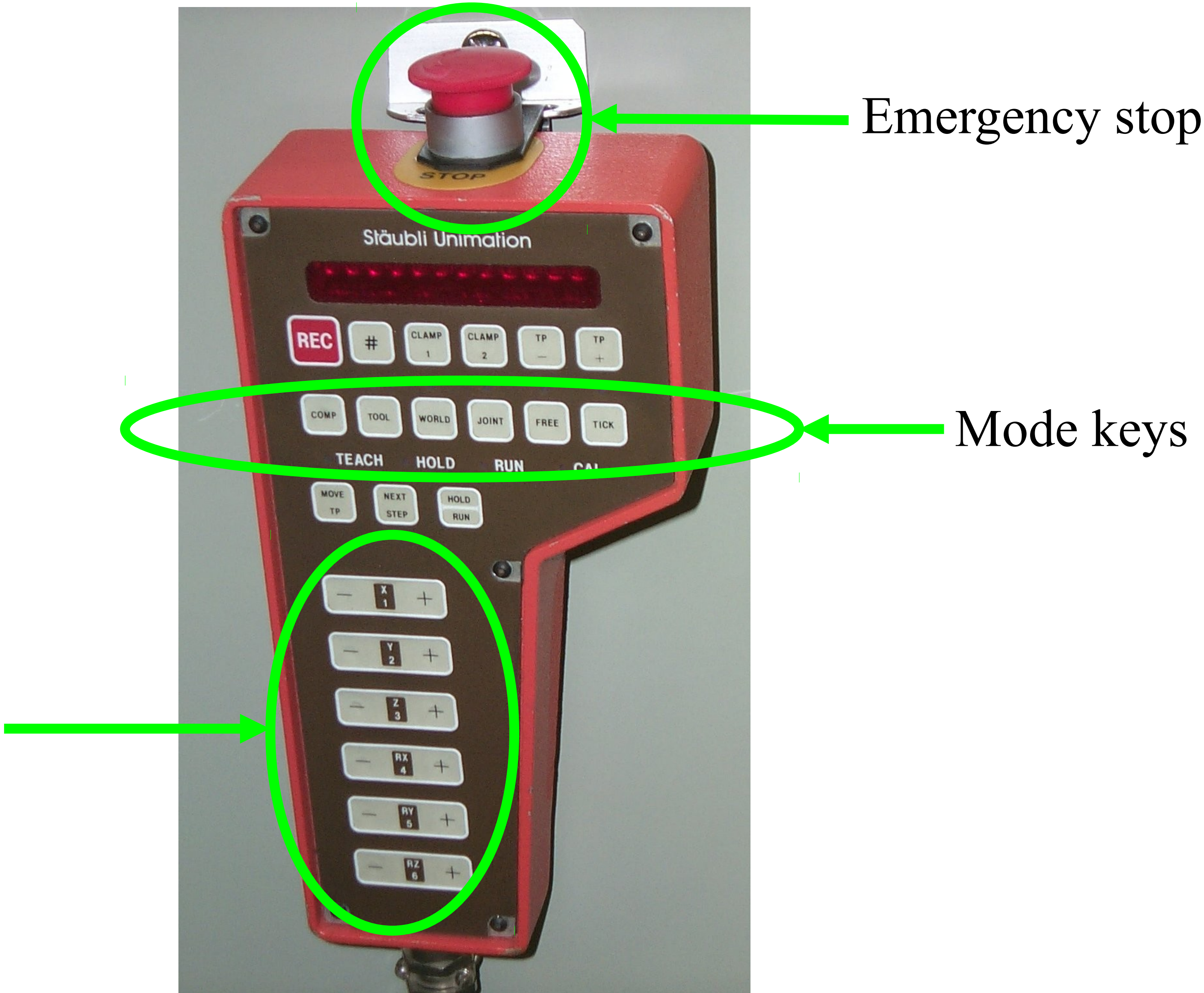
Location recording process

- Manually move the robot, equipped with the proper tool, to the desired location using the **teach pendant**.
- Record and name the location either as a cartesian or as a joint location
- Sometimes, you have to specify whether it is a **fly-by point** or a **stop-point**.
  - Speed is non zero at a fly-by point (typically not a functional point).
  - Speed and position error are nulled at a stop-point.

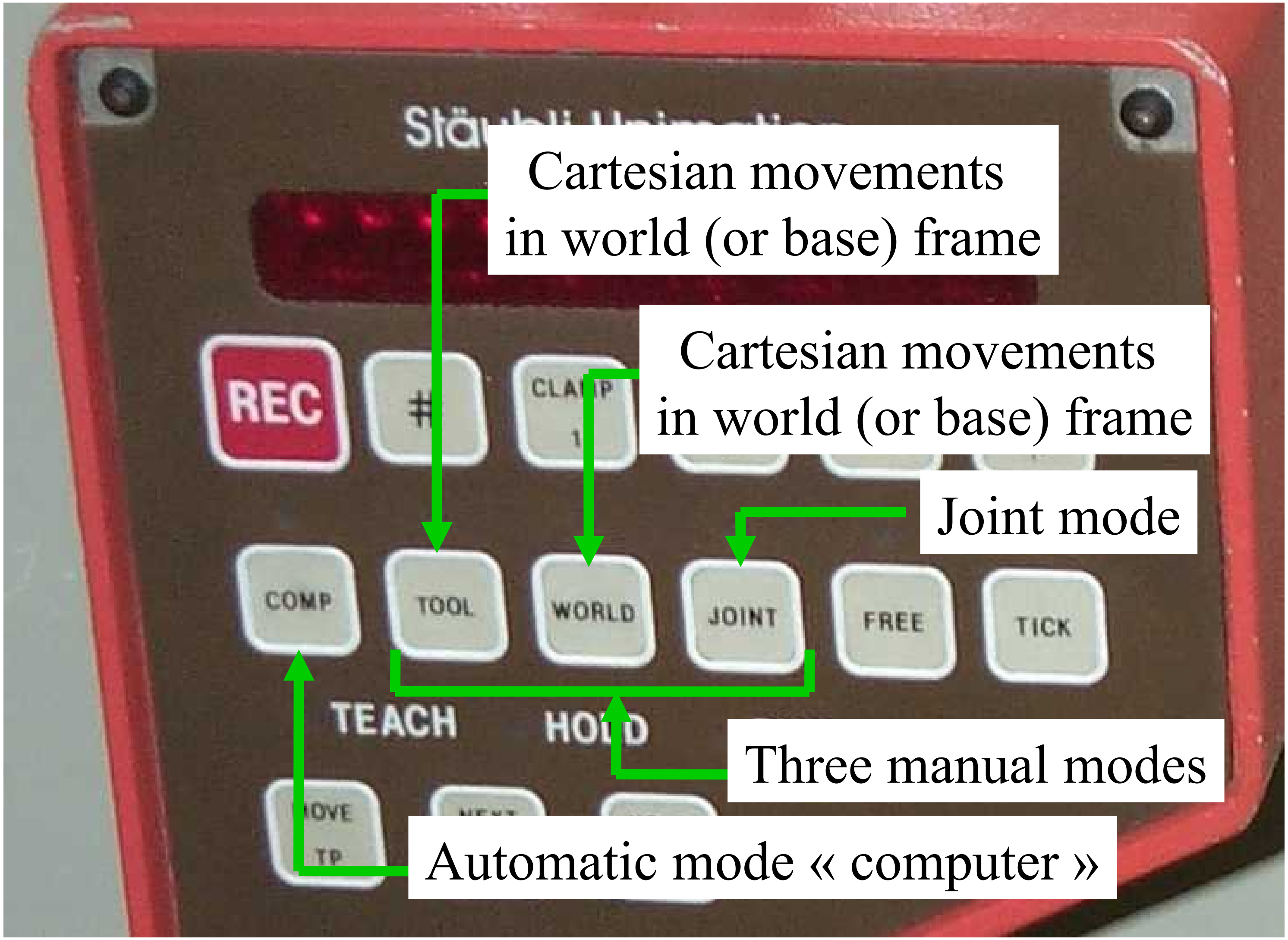
Examples of teach pendants

790-7

Teach pendant - Puma 560

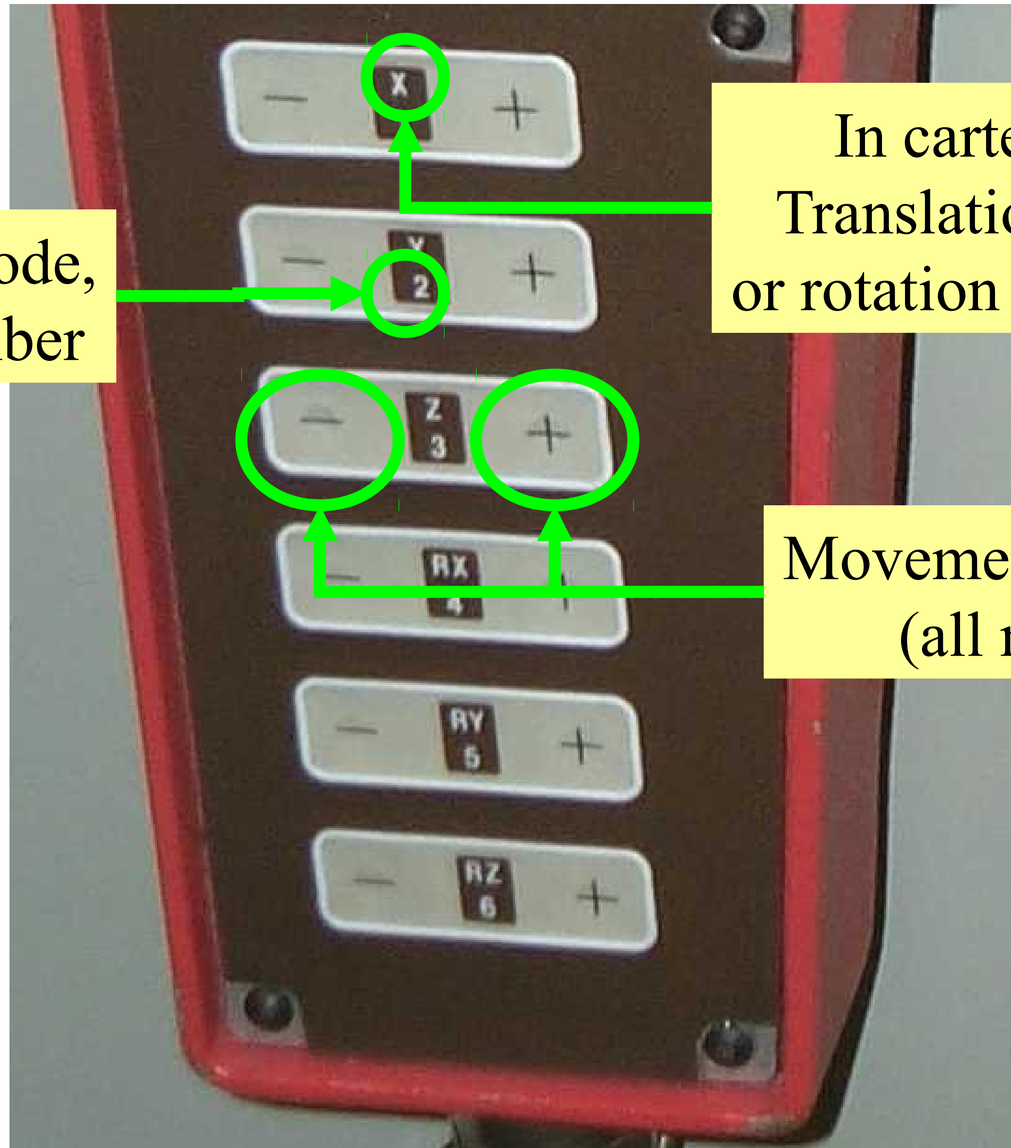


Mode keys



Cartesian movements in world (or base) frame
Cartesian movements in world (or base) frame
Joint mode
Three manual modes
Automatic mode « computer »
REC
#
COMP
TOOL
WORLD
JOINT
FREE
TICK
TEACH

Motion keys



In joint mode,
Joint number

In cartesian mode,
Translation axis (x,y,z)
or rotation axis (Rx,Ry,Rz)

Movement direction
(all modes)

Teach pendant - Stäubli RX90

Manual mode indicators

Manual mode selection

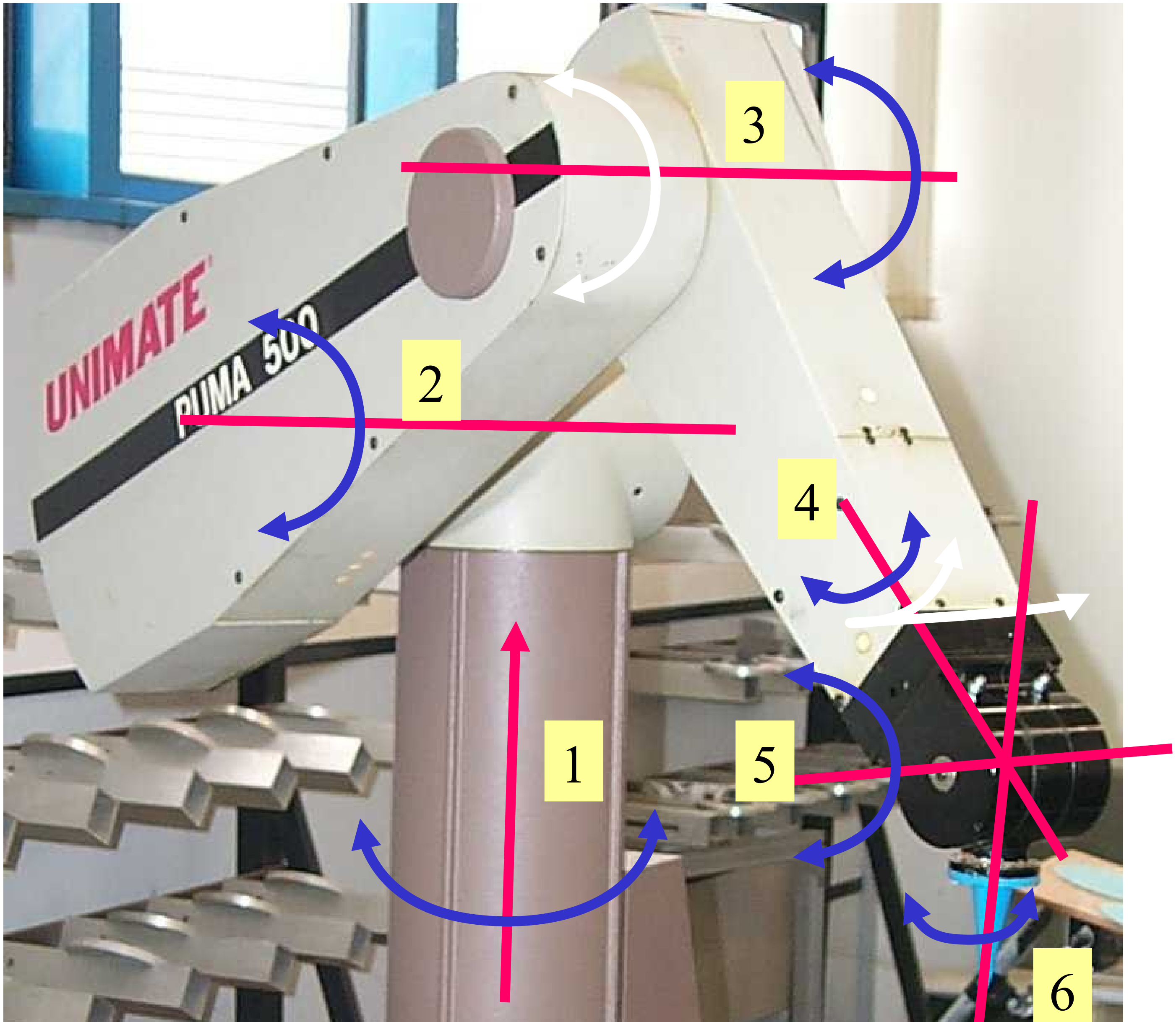
Direction and speed

Displacements

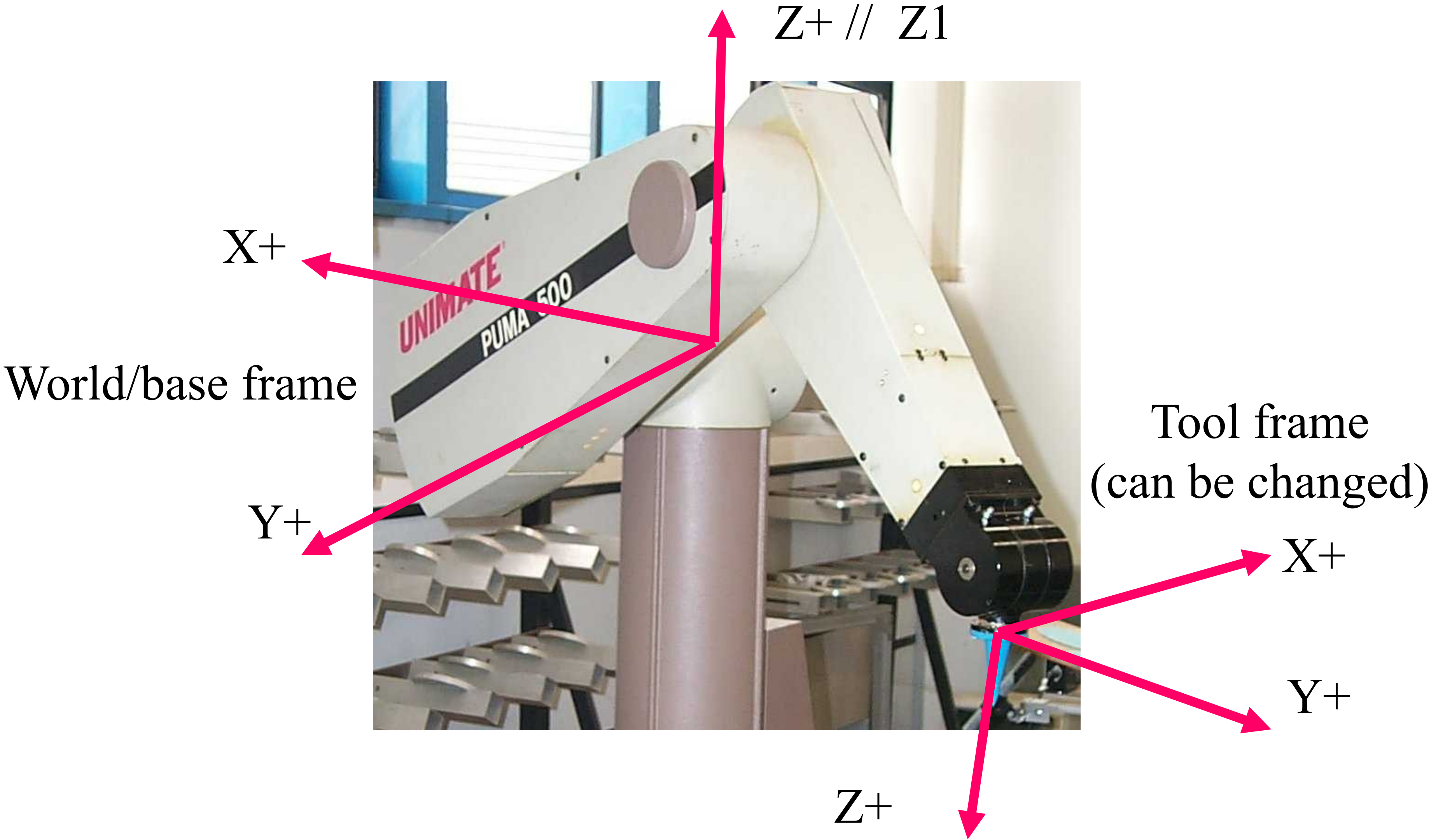
Dead-man security

Teach pendants may look very different, but you will always find the same basic manual modes to move the robot

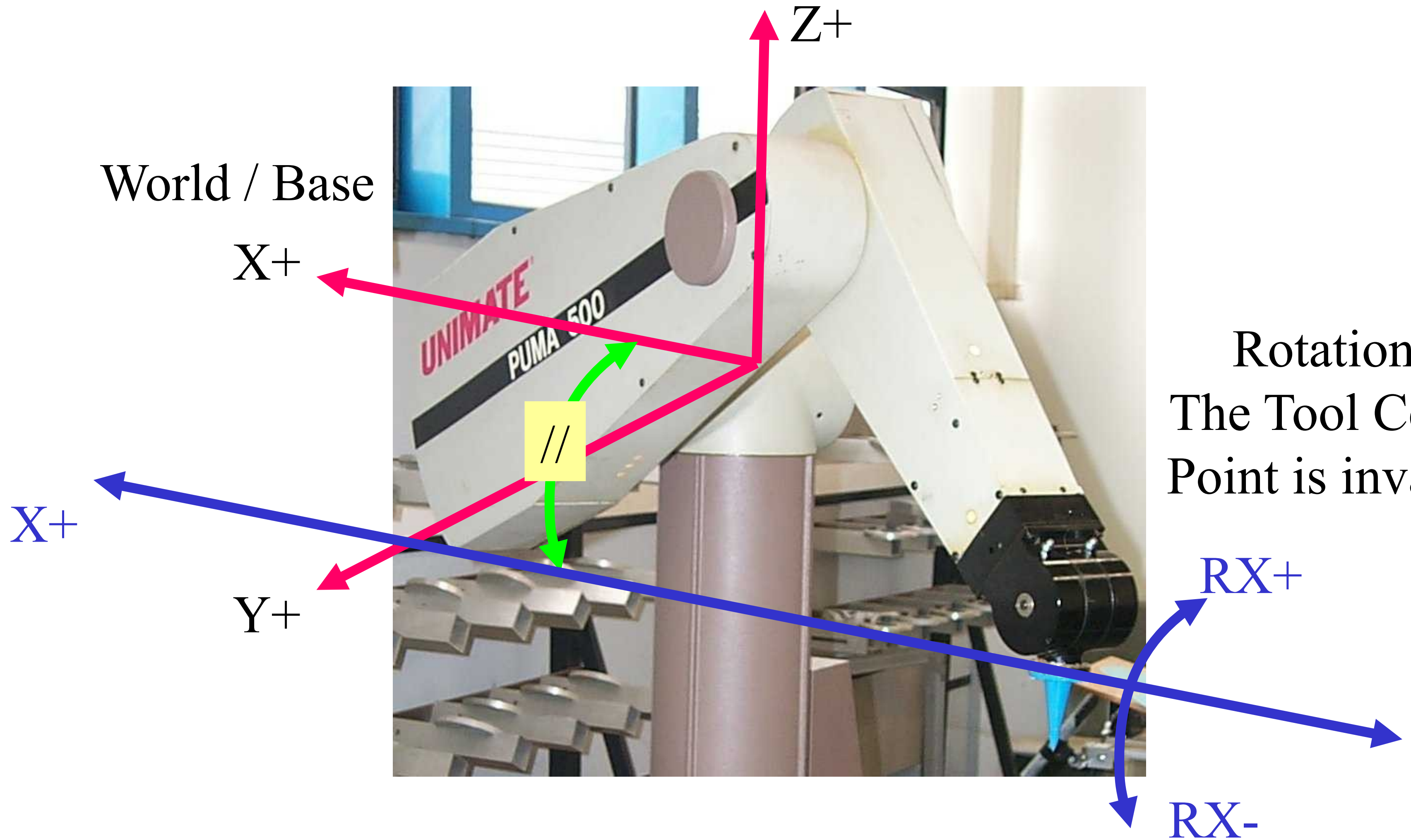
The joint mode



Cartesian modes: World and tool frames

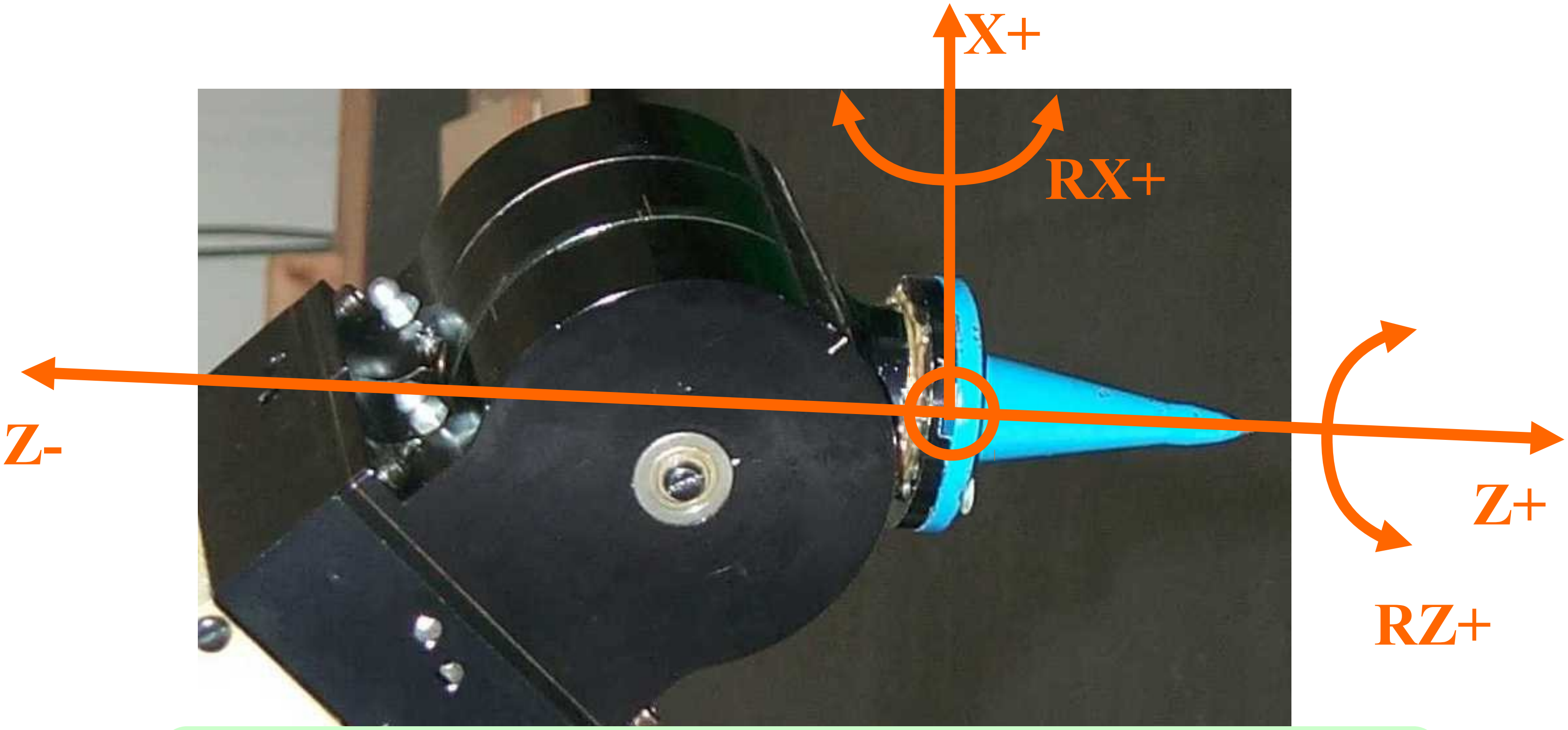


Displacements in world mode



Rotations:
The Tool Center
Point is invariant

Displacements in tool mode



Use of the teach pendant

- Someone (who clearly has moderate but not zero experience) struggling with a task requiring alignment of the tool with an arbitrary direction. Observe that the clearance is generous...

  https://www.youtube.com/watch?v=A9M1zX3GzX0

- Traditional vs joystick based teach pendant:

  https://www.youtube.com/watch?v=KTKAsZaEmjs

- Teach pendant vs active compliance base teaching (suggestion: switch off audio!)

  https://www.youtube.com/watch?v=jzR5NZrZSu0

  Note : the operator is an experience teach pendant user.

Still simple applications of robots

-

Tasks made repeatable

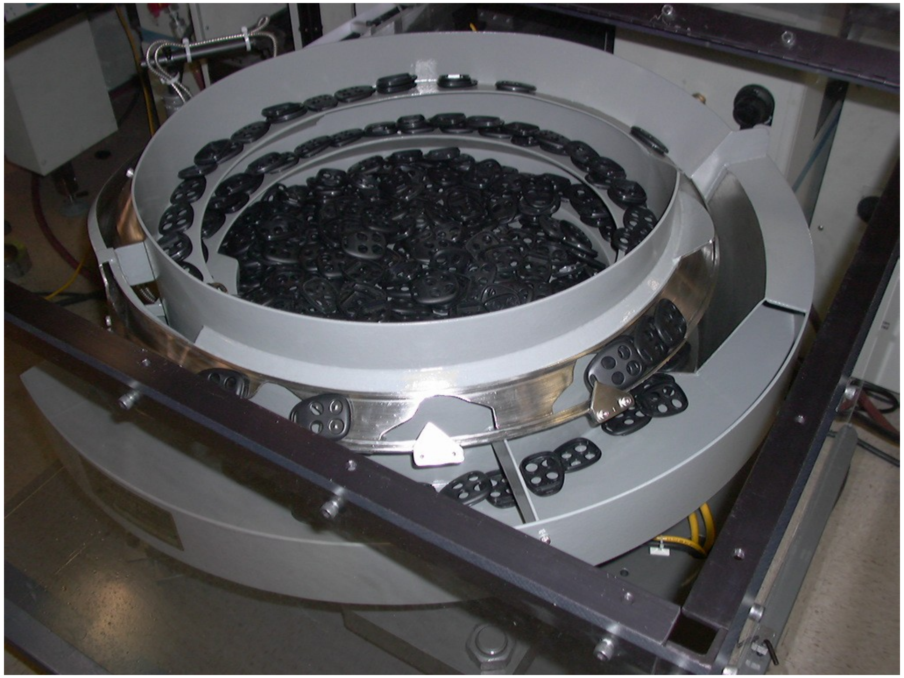
with some efforts … and cost

What if the task is not in essence repeatable?

There are several ways to solve the problem:

- Give the robot sensing capabilities to determine exact locations at execution time (more on this later)
- Use tools which naturally adapt to small variations of object locations (compliance).
- Easiest solution: force task repeatability using perirobotic systems.

Examples of perirobotic systems



Vibrating part feeders: from unstacked parts to perfectly positionned and aligned parts

Example: http://www.youtube.com/watch?v=gBBOb1ZitGE

Examples of perirobotic systems

Part orientation by a centrifugal feeder

Examples: http://www.youtube.com/watch?v=qrid3PiRtLE

A number of applications are shown here:

http://www.youtube.com/watch?v=9pWtjwkrPCs

Remarks about such systems

- They are used to **compensate for the robot's inability to sense its environment**.
- They are **costly**.
- They must be **adapted to the type of parts** to be manipulated.
- Removing part feeders would reduce implementation costs and downtime.

In terms of flexibility, perirobotics is often the weak link of robotized applications

CENTRALE NANTES

Examples of cases when teaching locations is not satisfactory

Spot welding

Spot welding

The number of welding points can be very high

Video: https://www.youtube.com/watch?v=KEQdn57Kz1Q

Spot welding

- If the robot is to is to use the same locations at each cycle, the task must be repeatable…
- Repeatability of the task requires efforts:
  - To stop the car body at the same position at each cycle.
  - To keep the dimensional variations of the **car body** within narrow bands.
  - To keep the dimensional variations of the **tool** within narrow bands (electrodes must be changed regularly …)

Spot welding

- Even if the task is made repeatable, it is still unsatisfactory, because:
  - The task requires a large number of locations.
  - You have to teach the left robot its locations, even though they are symmetric to those used by the right robot (a common situation).
  - Teaching locations would be performed on site, resulting in a long downtime (no production during teaching time).
- These are the reasons for developping robot simulation and offline programming (see later…)

Very few points, and yet…

Here, the problem is to choose the right robot and to position it correctly for the task. Simulation is the key!

In this video: https://www.youtube.com/watch?v=8_lfxPI5ObM you will see the robotized seat mounting around 3'30.

Many regularly spaced objects

4x15 element pallet

Oups! Stackable!

Suppose a robot uses 10 such pallets as a stock …

- Will the user have to teach the robot 600 locations?
  If he (or his robot) knows only about teaching locations, he definitely will!
- What if he realises that it would be better to slightly shift the pallets to reorganize the workcell? A nightmare!

Video showing an example with a very large number of locations:

http://www.youtube.com/watch?v=qTu_jclavAE

Applications where the trajectory matters

Painting applications

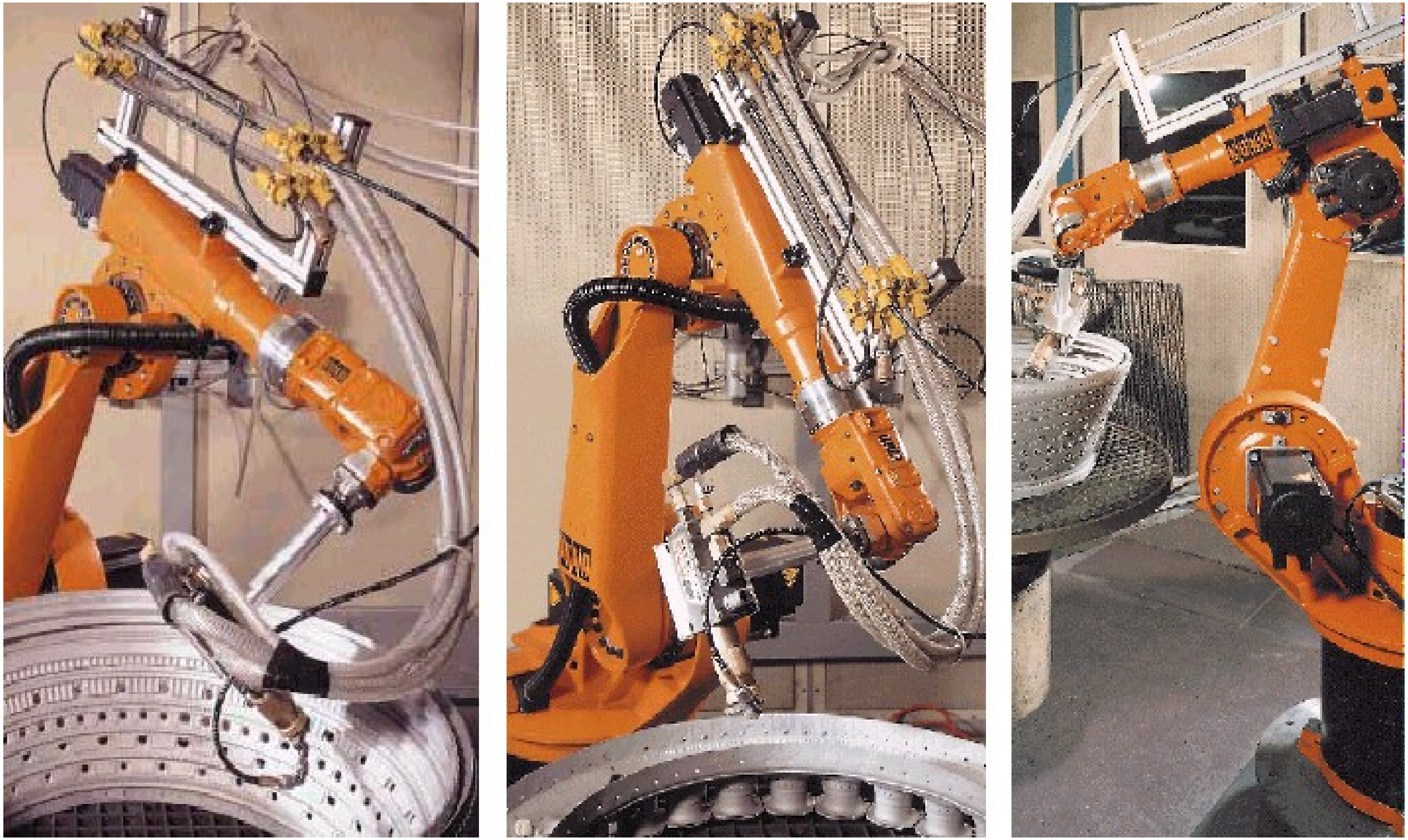
Back to a previous video
(painting is seen at 2'40):

https://www.youtube.com/watch?v=8_lfxPI5ObM

Painting applications

- Users don’t care too much about:
  - Exact tool position
  - Exact tool orientation
- They care about:
  - Homogeneous thickness
  - No dripping
- Quality is related to trajectory « smoothness ».
- Painters know how to move the tool for a good quality result.

Similar cases

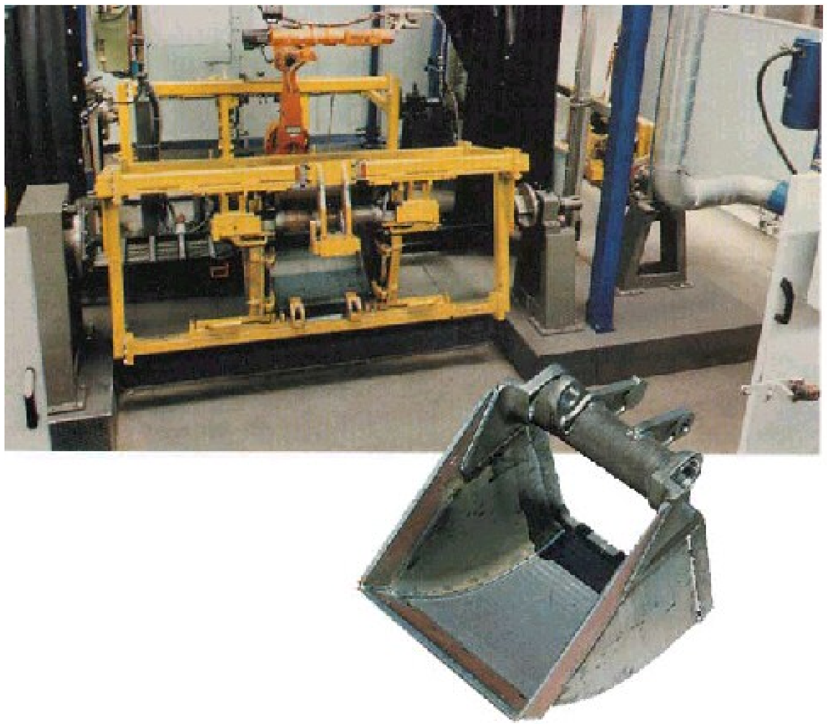


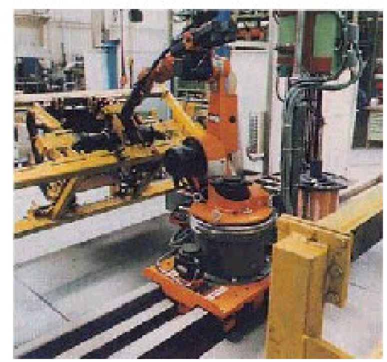
Some process applications have similar characteristics to painting

How to teach trajectories

- Locations are recorded along the path (played by a painter) as a function of time.
- Using an active compliance robot:
  - The operator manually leads the end effector
  - Robot actuators « cancel » operator efforts
  - Generally uses a force/torque wrist sensor
- Using a mechanical simulating device
  - Lightweight structure with encoders
  - The kinematic structure need not be the same as the robot's kinematic structure.
- There are now dedicated CAD modules for painting.

Trajectory *and* exact position *and* exact orientation can matter





- Dedicated seam following sensors.
- Necessary to manage more than six axes in a synchronized way.

Partial conclusions

Remember

- Robots are attractive to automate tasks but they lack (at least)
  - A universal end effector
  - Sensory capabilities
- Robots have excellent repeatability but fairly poor absolute precision
- Locations and trajectories can be taught to a robot but it’s generally a fairly slow process.

Remember

- Task repeatability makes things easier.
- Task repeatability can generally be forced, but at a cost.
- Purely teaching locations is not satisfactory if there are many locations and because it increases downtime.
- Some tasks could not be solved satisfactorily without CAD based simulation (see later).