ERF2014 WORKSHOP

Dual-Arm Robots for Skilled Manufacturing Applications

13th March 2014, Roveretto, Italy

Dual arm robots programming

D. Surdilovic, Fraunhofer – IPK

Objectives

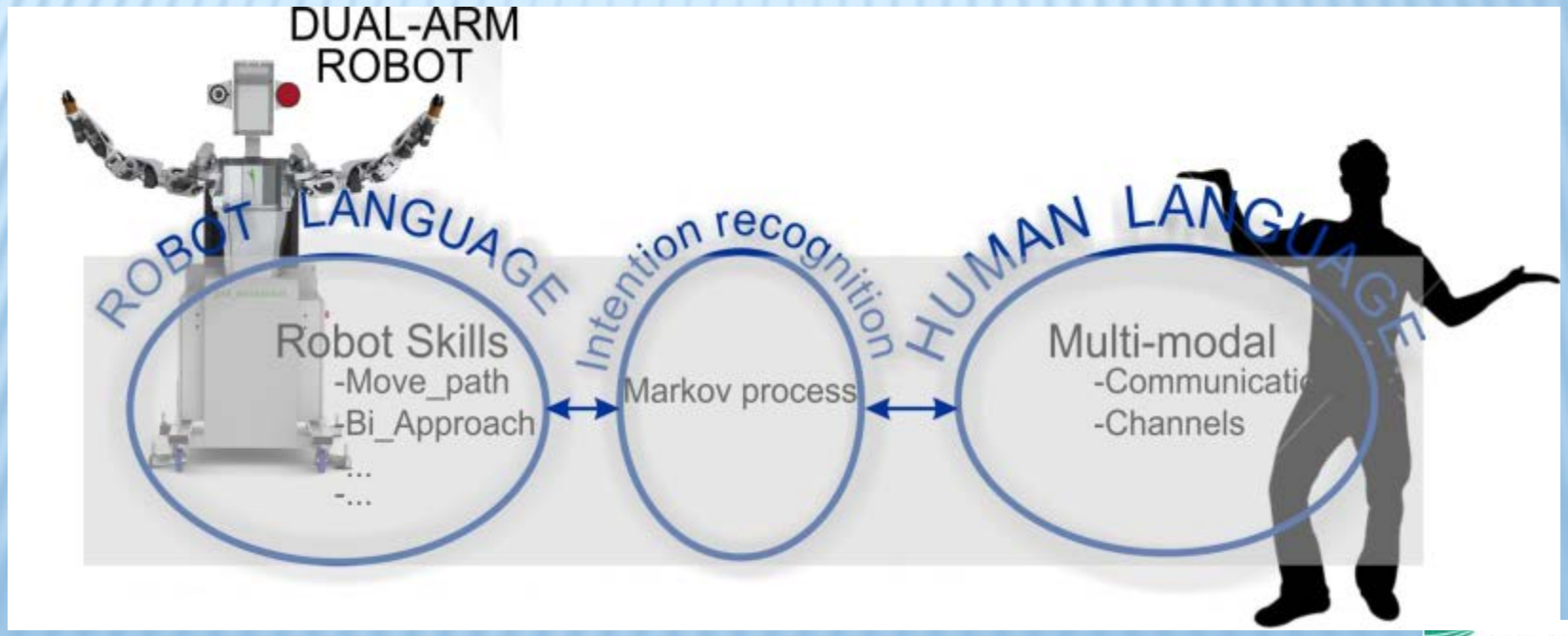
Programming is one of the two main bottlenecks for wider applications of dual-arm robots in industry

- **Ambituous framework** to support :
  - **Task-oriented** dual-arm robot programming (activities decomposition: order, jobs, tasks, operations)
  - **Sensor-based robot control** (**force/torque control, impedance**, vision – look and move etc., force-vision)
  - **Interactive programming** (gesture, voice, manual guidance……..)
  - Robust and flexible execution of complex assembly/disassembly tasks (including contact operations) in a structured, but uncertain environment (compensate for robot/environemnt errors/tolerances)

Fraunhofer IPK

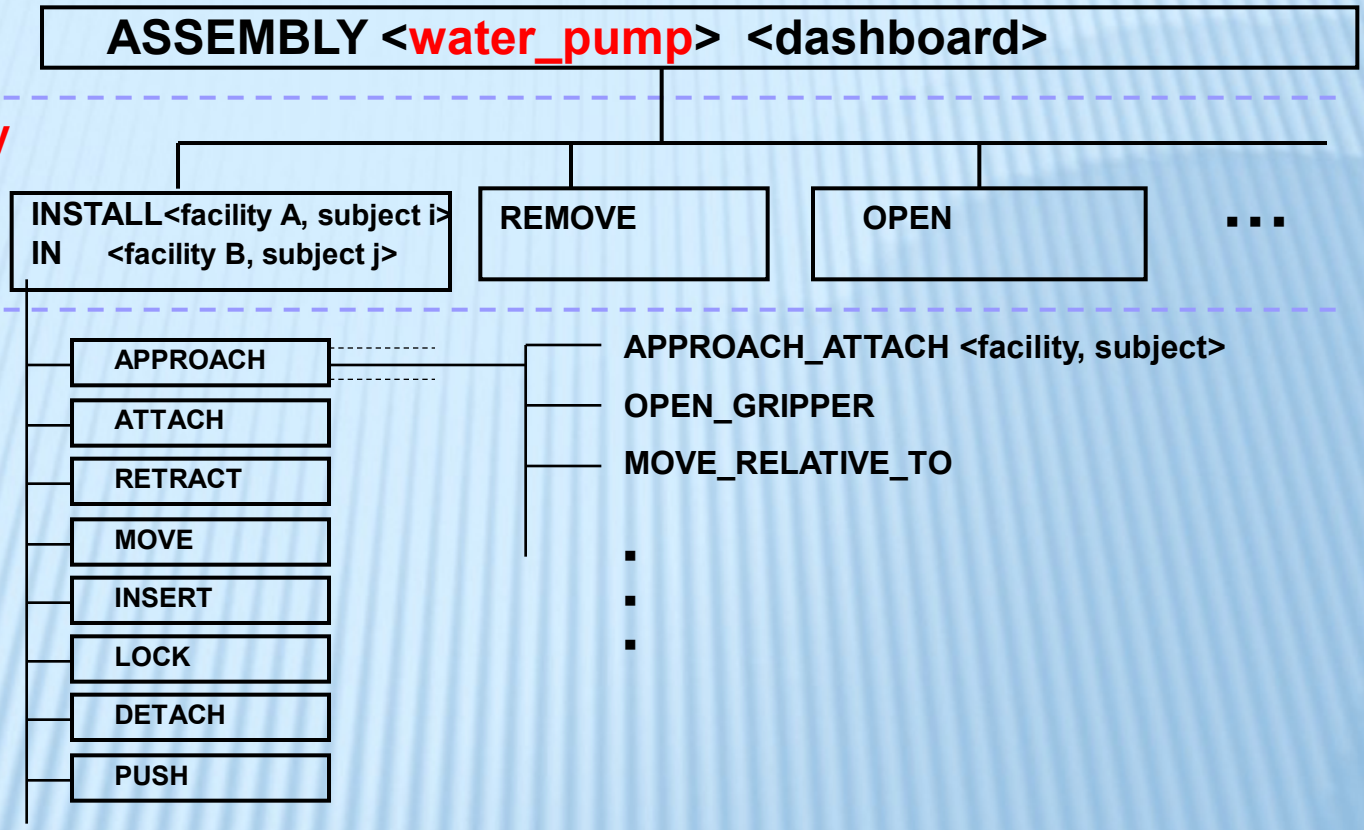
Interactive dual arm robot programming

- Define robotic skills in terms of activities:
  - Supports hierarchical decomposition,
  - General purpose vs. specific activities (avoid teaching)
  - Scaling
  - Explicit/implicit, object-oriented programming
  - Robotic-Language DARL - key approach



BACKGROUND FOR CONTROL/PROGRAMMING DEVELOPMENT

Higher complex activity
JOB, ***TASK***



Lowest level of activity

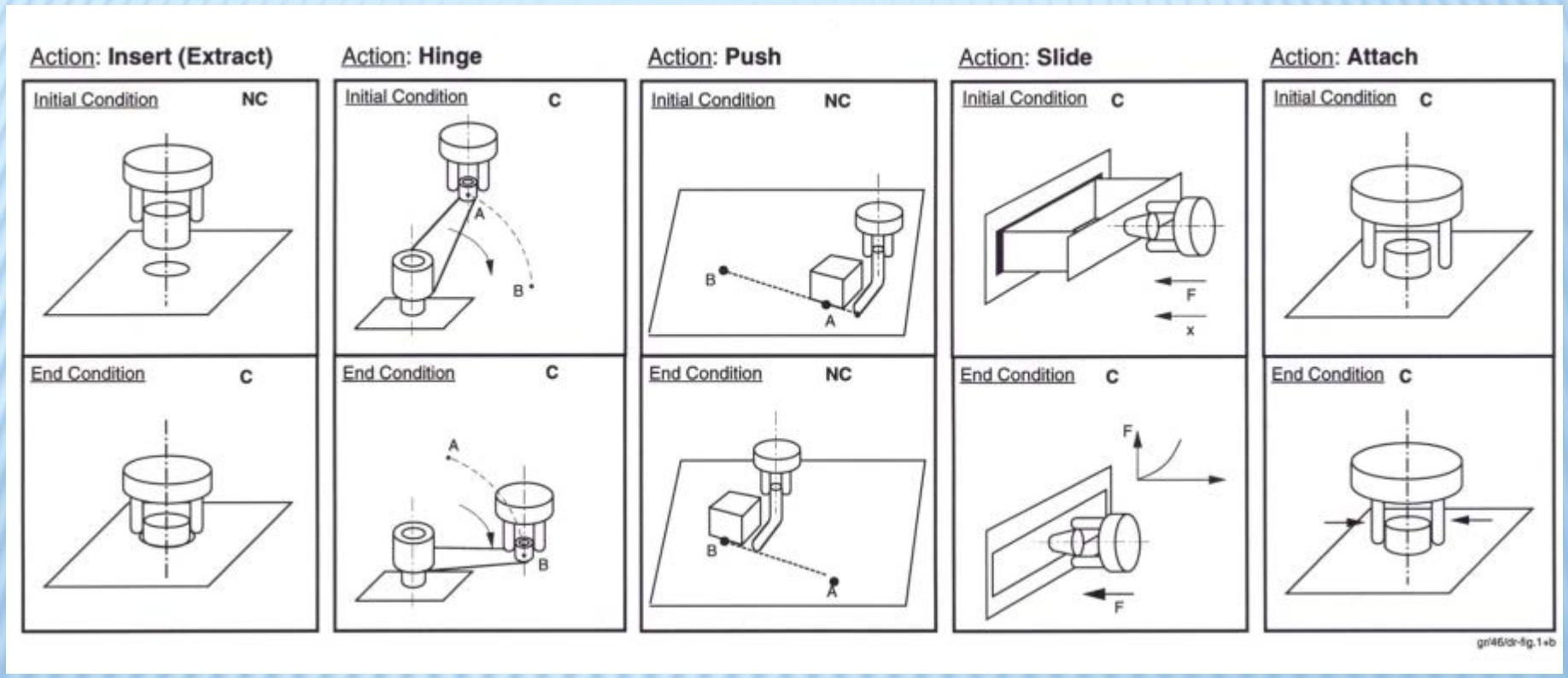
- Assigned to a sub-system (device)
- Mapped to a system ability (algorithm)

OPERATIONS *(ACTIONS)*

Atomic activity PDL2 Robot/Motion Commands, New EXT-Commands

Fraunhofer IPK

Single Arm - Contact Operations (Actions)



HOLD, YIELD, LOCK, UNLOCK etc. (GET_CONTACT, APPLY_FORCE)

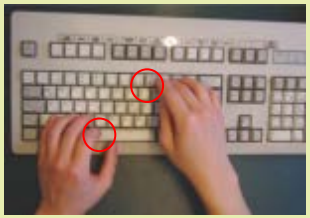
Fraunhofer IPK

Dual-arm operations

coordinated

goal-coordinated

non-coordinated



Jigless operations:

L: Action ⟷ **R: Action**

HOLD **INSERT**

bimanual

symmetric **asymmetric**

congruent **non-congruent**

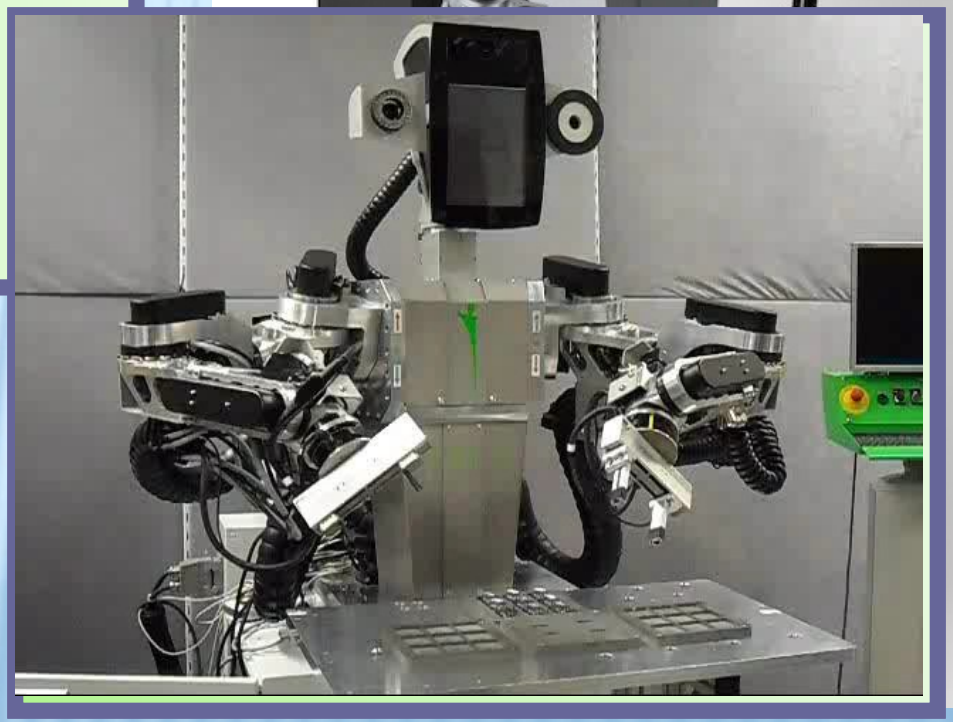
Why human performs mostly simplified dual-arm motion?

Bottleneck : planning (monitoring), not control

Fraunhofer IPK

Simple planning and programming of human-like bimanual motion („Callosum“ – control)

-Symetric/asymetric, congruent/non-congruent motions
-Arms collision monitoring and avoidance



Fraunhofer IPK

Bimanual Operations

Jigless Operation

Pure Bimanual Operation

Fraunhofer IPK

Dual-Arm Actions (Mimic Human Motion)

Bi-Approach (Retract)

Bi-Grasp (Release)

Bi-Insert (Extract)

Bi-Hinge

Bi-manual operations performed on a common object

Fraunhofer IPK

Dual-Arm Actions (Mimic Human Motion)

Bi-Slide

Bi-Hold

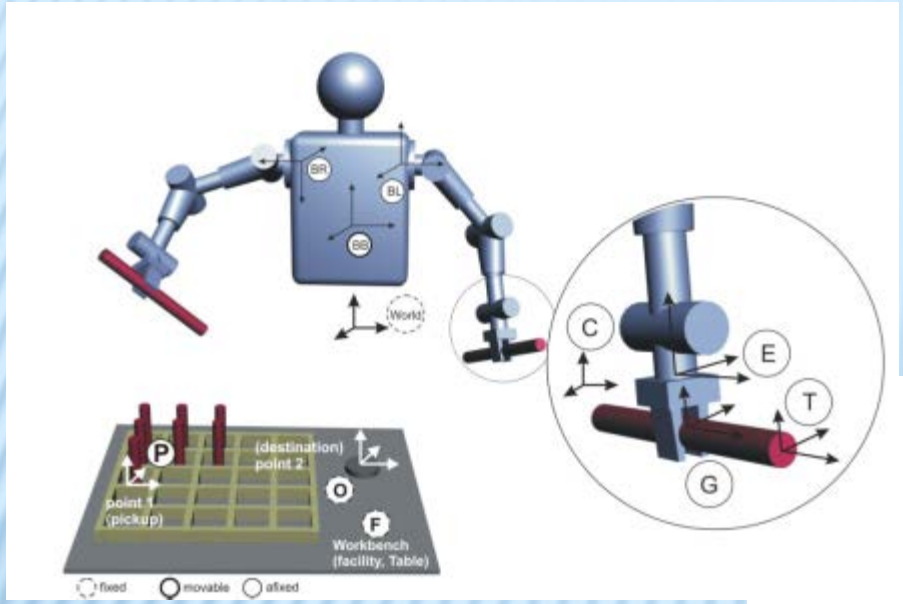
Bi-Move

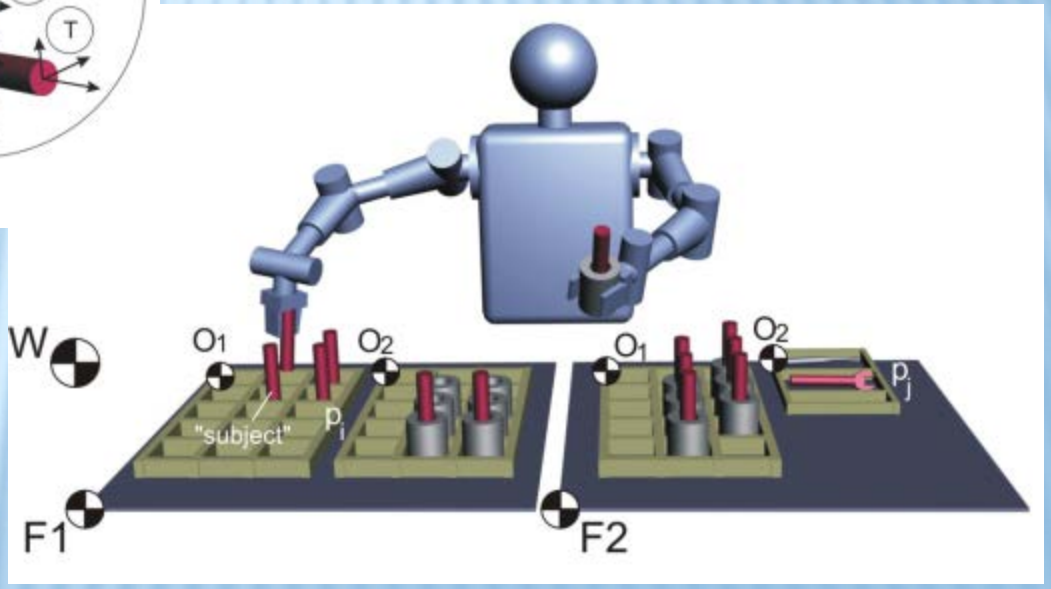
Bi-Yield

Bi-manual operations performed on a common object

Fraunhofer IPK

WORLD MODEL : ROBOT AND ENVIRONMENT FRAMES

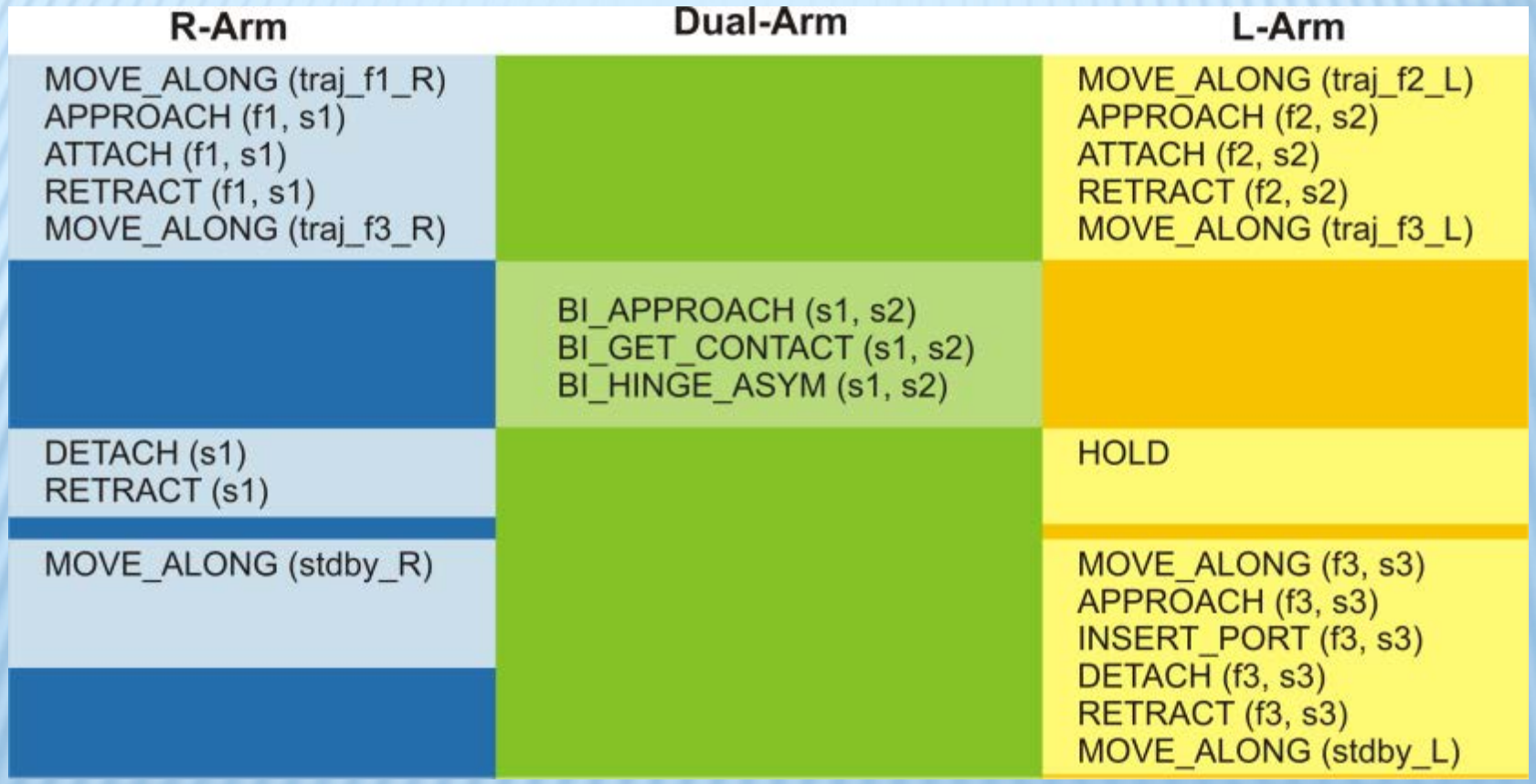




Structured but uncertain environment

Fraunhofer IPK

Action and Tasks Bi-Manual (Lego-Like) Programming



High-Level Object-orinted dual-arm robot programmimg language (C++)– *Advanced frame for dual-arm robot programming*

```
lArm.setIMCOStatusBlocking(SYSTEM_STATUS_MONITORING);
lArm.setComplianceGains( gLowStiff );
lArm.setComplianceFrame( CartPose() );
lArm.setIMCOStatusBlocking(SYSTEM_STATUS_RUNNING);

biMovePTP(lOverFacility, rOverFacility);
biExecuteBlocking();
```

BI-MANUAL-ACTIONS SUPPORTED BY DARL

For example:

- **biMovePTP**(**const** JointPose& pl, **const** JointPose& pr, JointPose jspd_l, JointPose jspd_r)
- **biMoveTo**(CartPose cl, CartPose cr, **double** spd_l, **double** spd_r)
- **biApproachInsert**(Subject& hl, Subject& hr, **double** spd=APPROACHINSERT, Blend* blend=NULL, **bool** openBlend=**false**)
- **biInsert**(Subject& hl, Subject& hr, **double** spd=INSERT)
- **biPutSubjectTo**(Subject& hl, Subject& hr, Blend* blend=NULL, **bool**
- openBlend=**false**)
- **biMove2Approach**(Subject& sl, Subject& sr, **double** spd_app=APPROACH, Blend* blend=NULL, **bool** openBlend=**false**)

Fraunhofer IPK

BI-INSERT(SUBJECT& HL, SUBJECT& HR, DOUBLE SPD=INSERT)

biInsert is resolved to:

```
CartPose cl = hl.getPose();
lArm. setComplianceGains(ENGAGE_CONFIG);
lArm.setComplianceFrame(CartPose())
lArm << MoveLin(cl);
CartPose cr = hr.getPose();
lArm. setComplianceGains(ENGAGE_CONFIG);
rArm.setComplianceFrame(CartPose())
rArm << MoveLin(cr);
```

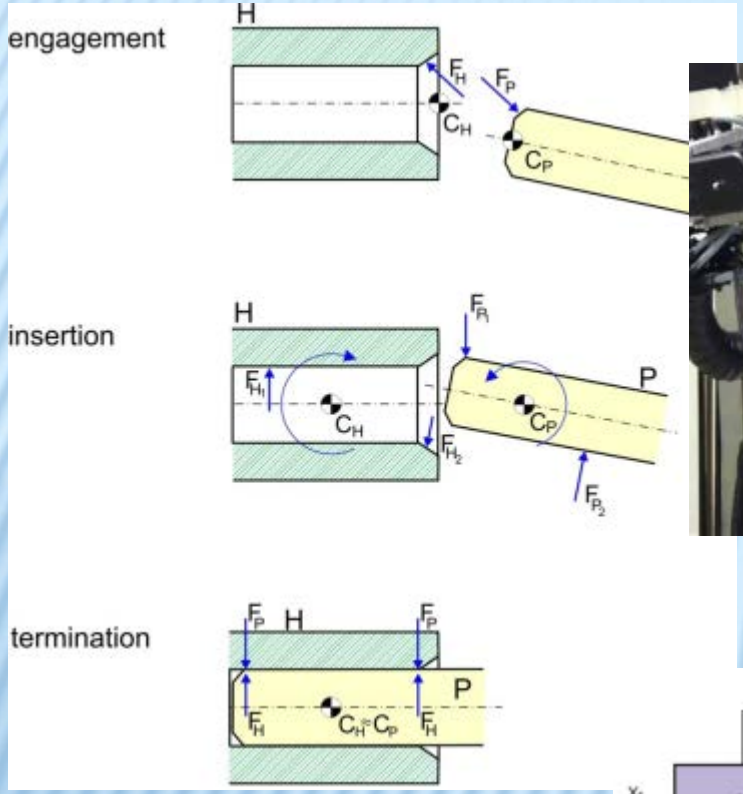
Engage

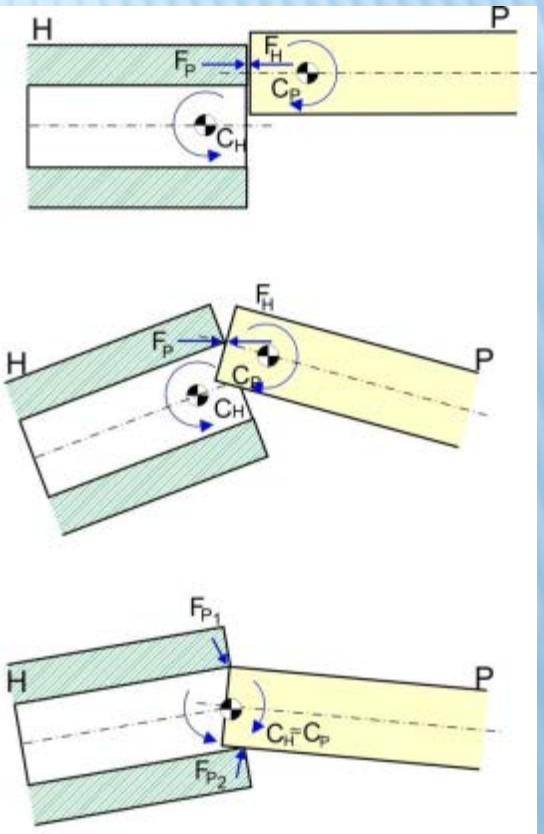
```
CartPose cl = hl.getPose();
lArm. setComplianceGains(INSERT_CONFIG);
lArm.setComplianceFrame(CartPose() +Vector3(0,0,-d/2)))
lArm << MoveLin(cl);
CartPose cr = hr.getPose();
lArm. setComplianceGains(INSERT_CONFIG);
rArm.setComplianceFrame(CartPose()+Vector3(0,0,-d/2))
rArm << MoveLin(cr);

lArm.execute();
rArm.execute();
```

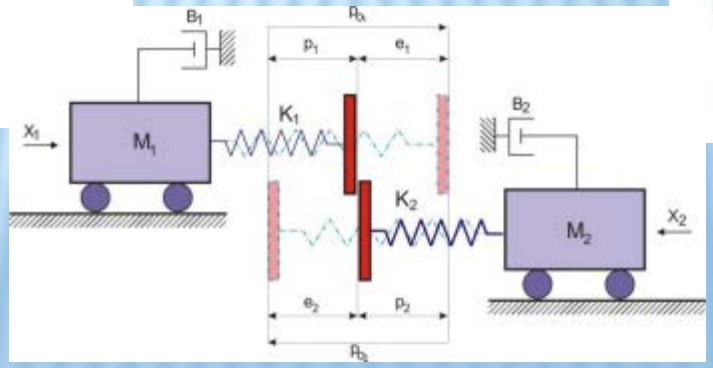
Insert

Fraunhofer IPK

EFFICIENT DUAL-ARM INSERTION (INTERACTION BETWEEN TWO COMPLIANT ARMS)



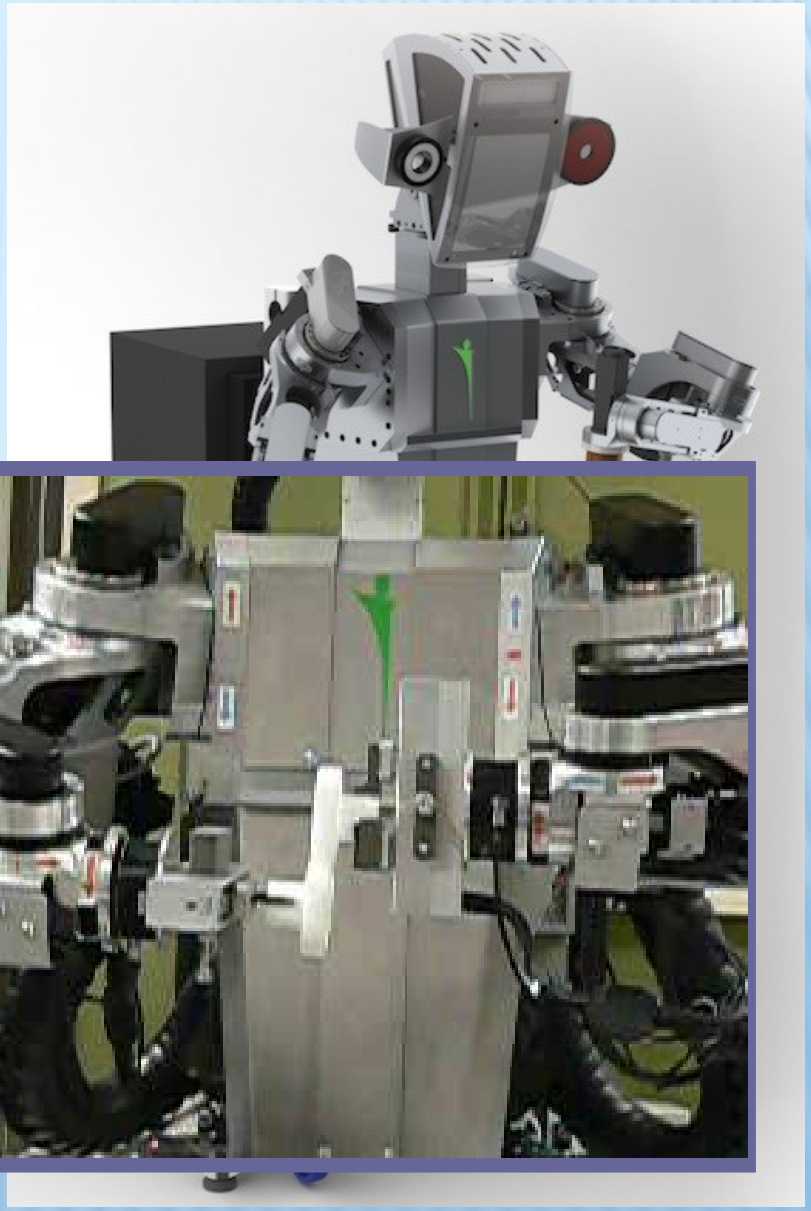


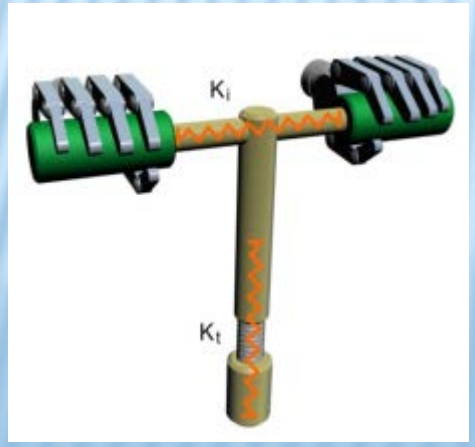


Fraunhofer IPK

Bimanual Contact Tasks Control

Single Arm/External (common object)/
Internal – impedance control





Fraunhofer IPK

DUAL ARMS SYNCHRONIZATION

Sync A, B;
right_arm << move (p1) << A << move (p2) << move (p3) << B;
left_arm << move (p4) << B << move (p5) ;
l e f t_g r i p p e r << A << c l o s e ( ) << B;

Fraunhofer IPK

1) Short set-time in a for human designed work-environment – *Half-our commissioning*

2) Redundant arms programming and control – *Efficient handling of two arms redundancy*

3) **Impedance and force control over all control layers** – *First implementation of programmable and configurable impedan and force control for industrial robots applications*

4) Task-Level Programming – *An old id becomes reality in dual-arm robots*

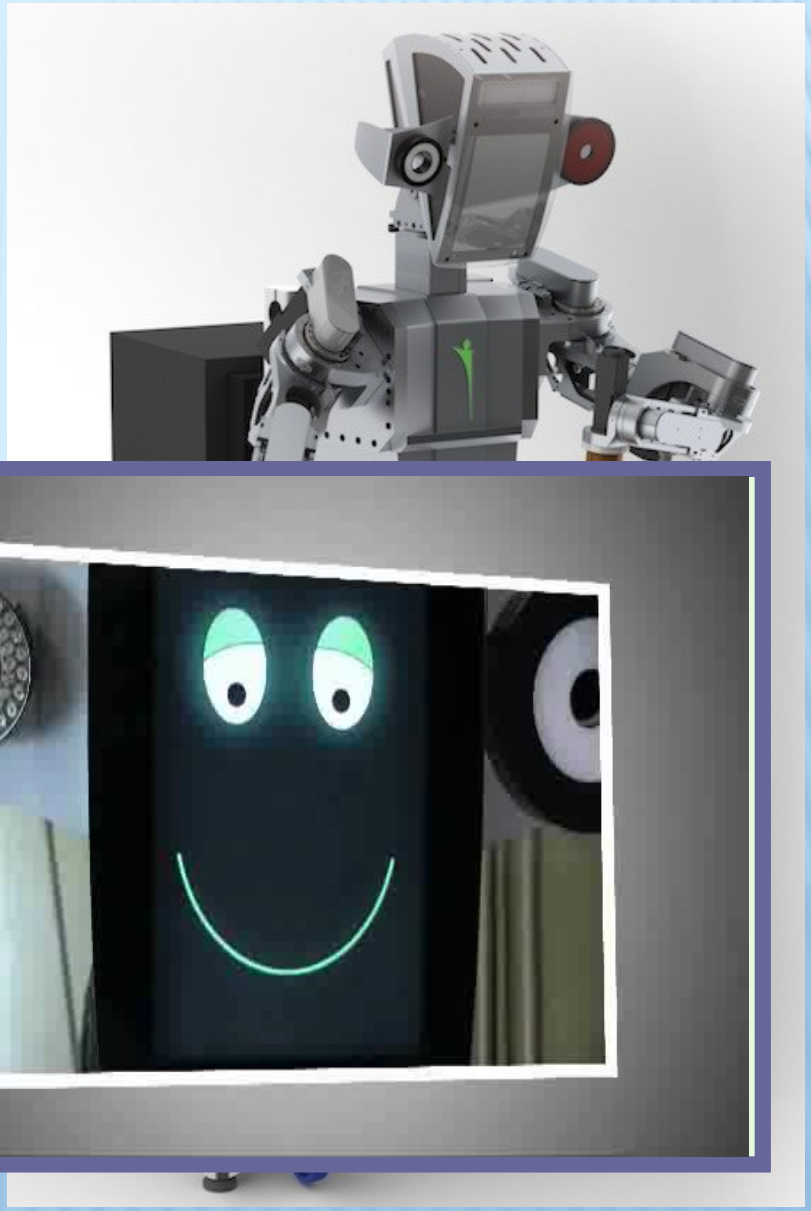
Fraunhofer IPK

DARL – Flexible framework for programming dual-arm robots in C++

ROS – package (generalization)

Connection to DB (ontological)

To be published soon (April 2014)



Fraunhofer IPK