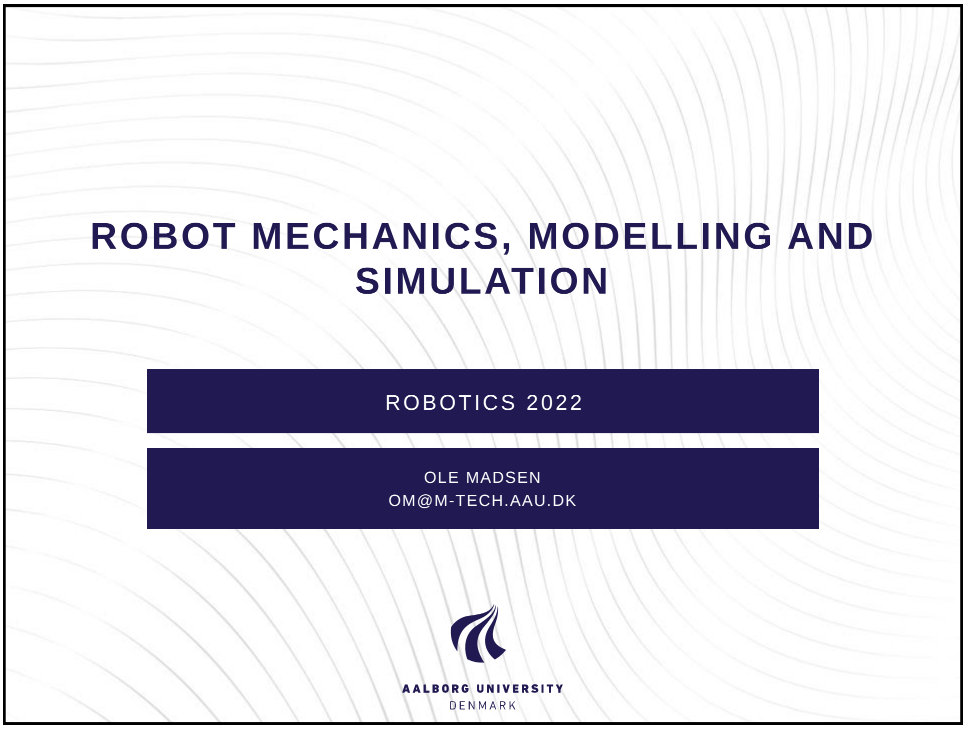
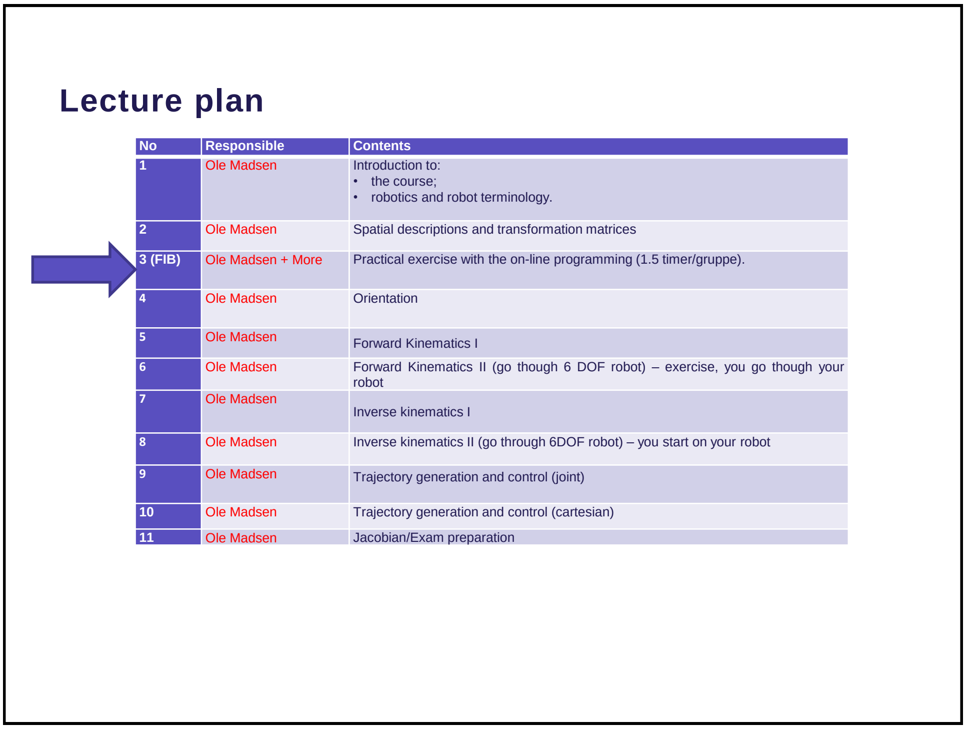


ROBOT MECHANICS, MODELLING AND SIMULATION

ROBOTICS 2022

OLE MADSEN
OM@M-TECH.AAU.DK

AALBORG UNIVERSITY
DENMARK



Lecture plan

No	Responsible	Contents
1	Ole Madsen	Introduction to: • the course; • robotics and robot terminology.
2	Ole Madsen	Spatial descriptions and transformation matrices
3 (FIB)	Ole Madsen + More	Practical exercise with the on-line programming (1.5 timer/gruppe).
4	Ole Madsen	Orientation
5	Ole Madsen	Forward Kinematics I
6	Ole Madsen	Forward Kinematics II (go though 6 DOF robot) – exercise, you go though your robot
7	Ole Madsen	Inverse kinematics I
8	Ole Madsen	Inverse kinematics II (go through 6DOF robot) – you start on your robot
9	Ole Madsen	Trajectory generation and control (joint)
10	Ole Madsen	Trajectory generation and control (cartesian)
11	Ole Madsen	Jacobian/Exam preparation

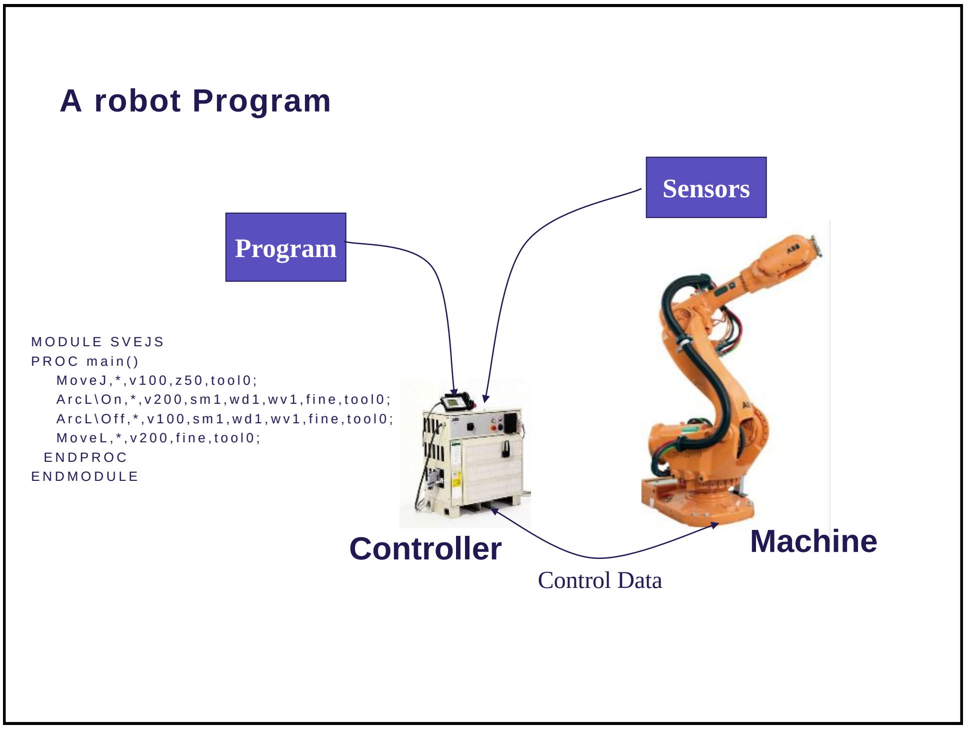
ROBOT PROGRAMMING

ROBOTICS 2022

OLE MADSEN
OM@M-TECH.AAU.DK

AALBORG UNIVERSITY
DENMARK

A robot Program



```
MODULE SVEJS
PROC main()
    MoveJ,*,v100,z50,tool0;
    ArcL\On,*,v200,sm1,wd1,wv1,fine,tool0;
    ArcL\Off,*,v100,sm1,wd1,wv1,fine,tool0;
    MoveL,*,v200,fine,tool0;
  ENDPROC
ENDMODULE
```

Programming levels

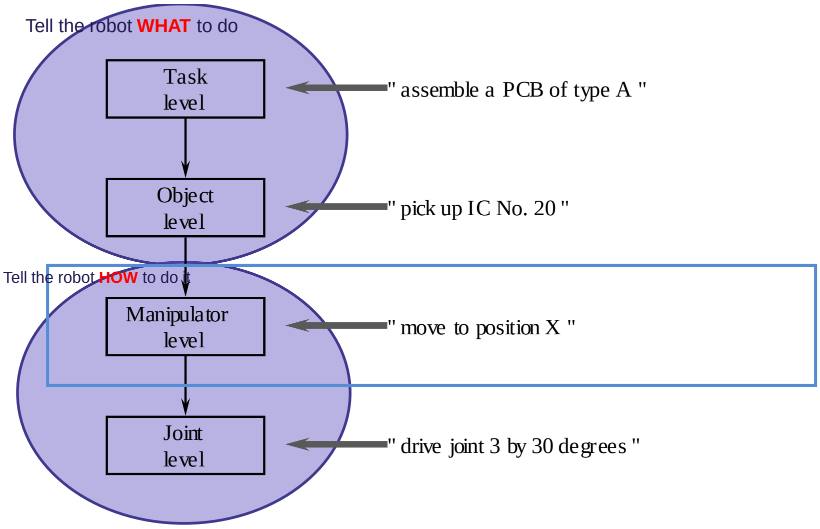
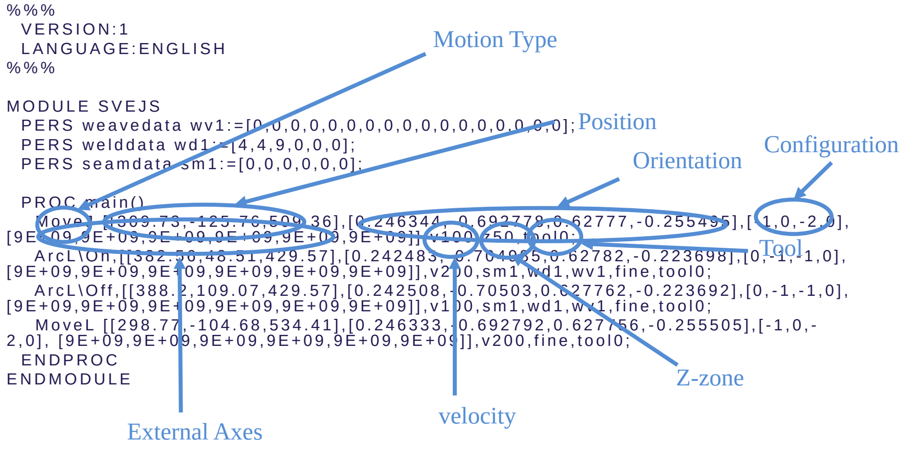

Tell the robot WHAT to do
Task level
Object level
Tell the robot HOW to do it
Manipulator level
Joint level
" assemble a PCB of type A "
" pick up IC No. 20 "
" move to position X "
" drive joint 3 by 30 degrees "

ABB program

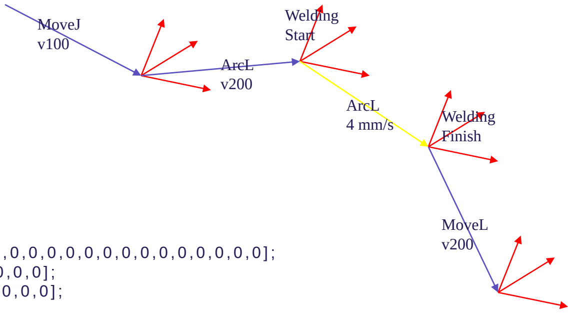


```
%%%
  VERSION:1
  LANGUAGE:ENGLISH
%%%

MODULE SVEJS
  PERS weavedata wv1:=[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
  PERS welddata wd1:=[4,4,9,0,0,0];
  PERS seamdata sm1:=[0,0,0,0,0,0];

  PROC main()
    MoveJ [[309.73,-125.76,509.36],[0.246344,-0.692778,0.62777,-0.255495],[-1,0,-2,0],
[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v100,z50,tool0;
    ArcL\On,[[382.56,48.51,429.57],[0.242483,-0.704965,0.62782,-0.223698],[0,-1,-1,0],
[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v200,sm1,wd1,wv1,fine,tool0;
    ArcL\Off,[[388.2,109.07,429.57],[0.242508,-0.70503,0.627762,-0.223692],[0,-1,-1,0],
[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v100,sm1,wd1,wv1,fine,tool0;
    MoveL [[298.77,-104.68,534.41],[0.246333,-0.692792,0.627756,-0.255505],[-1,0,-
2,0], [9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v200,fine,tool0;
  ENDPROC
ENDMODULE
```

ABB program



```
%%%
  VERSION:1
  LANGUAGE:ENGLISH
%%%

MODULE SVEJS
  PERS weavedata wv1:=[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
  PERS welddata wd1:=[4,4,9,0,0,0];
  PERS seamdata sm1:=[0,0,0,0,0,0];

  PROC main()
    MoveJ [[309.73,-125.76,509.36],[0.246344,-0.692778,0.62777,-0.255495],[-1,0,-2,0],
[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v100,z50,tool0;
    ArcL\On,[[382.56,48.51,429.57],[0.242483,-0.704985,0.62782,-0.223698],[0,-1,-1,0],
[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v200,sm1,wd1,wv1,fine,tool0;
    ArcL\Off,[[388.2,109.07,429.57],[0.242508,-0.70503,0.627762,-0.223692],[0,-1,-1,0],
[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v100,sm1,wd1,wv1,fine,tool0;
    MoveL [[298.77,-104.68,534.41],[0.246333,-0.692792,0.627756,-0.255505],[-1,0,-
2,0], [9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v200,fine,tool0;
  ENDPROC
ENDMODULE
```

ABB program

```
PROC main()
   writeToLog:=TRUE;
   TPWrite "Initializing...";
    WaitForStart;
    IF startIndicator<>5 THEN
        RETURN;
    ENDIF
    InitVars;
    TPWrite "Connected to ipac";
    WHILE continue DO
              WaitTime 0.1;
              IF currentInstruct=3 THEN
            HandleError;
              ENDIF
              IF currentInstruct=1 THEN
            MoveTo;
              ENDIF
     ENDWHILE
ENDPROC
```

```
PROC MoveTo()
    TPWrite "start moveTo";
    nextDest.trans.x:=moveToData{1};
    nextDest.trans.y:=moveToData{2};
    nextDest.trans.z:=moveToData{3};
    nextDest.rot.q1:=moveToData{4};
    nextDest.rot.q2:=moveToData{5};
    nextDest.rot.q3:=moveToData{6};
    nextDest.rot.q4:=moveToData{7};
    ConfJ\Off;
    ConfL\Off;
    MoveL
    nextDest,v1000\V:=moveToData{8},fine
             ,tiwa\WObj:=wobj0;
    TPWrite "Finnished moveTo";
ENDPROC
```

On-line programming

Takes place at the site of production itself and involves the work cell.

On-line programming

Advantages

- Cheap
- Easily accessible
- Programs made in concordance with the actual position of equipment and pieces.

Disadvantages

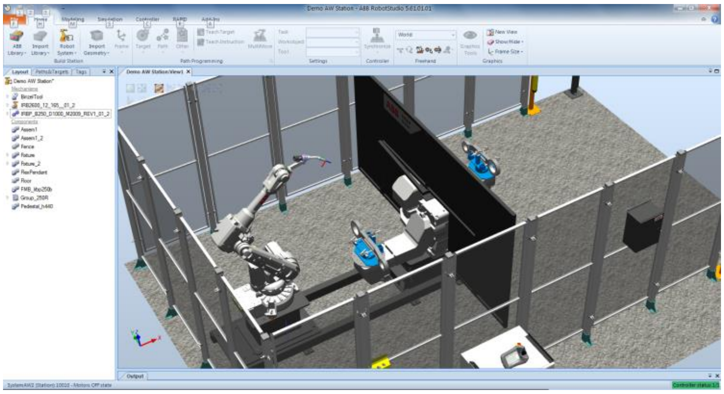
- Suspension of production while programming.
- Slow movement of the robot while programming.
- Program logic and calculations are hard to program.
- Poorly documented

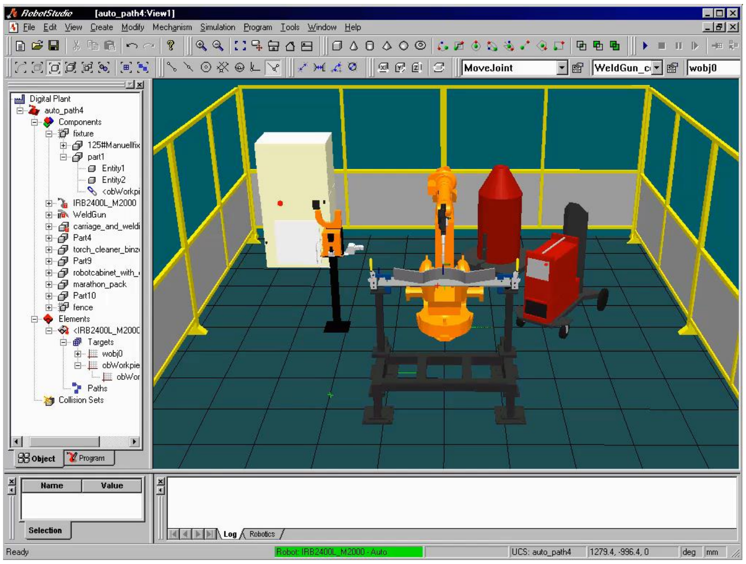
On-line programming

Process	Programming time	Process time	Ratio
Welding (complex)	9.600 min (20 days)	20 min	480
Welding (simpel)	40 min	1 min	40
Bending	480 min (8 hours)	5 min	96

Off-line programming

- Takes place on a computer without use of the physical robot
- Uses models of the work cell with robot, work pieces, process and surrounding.





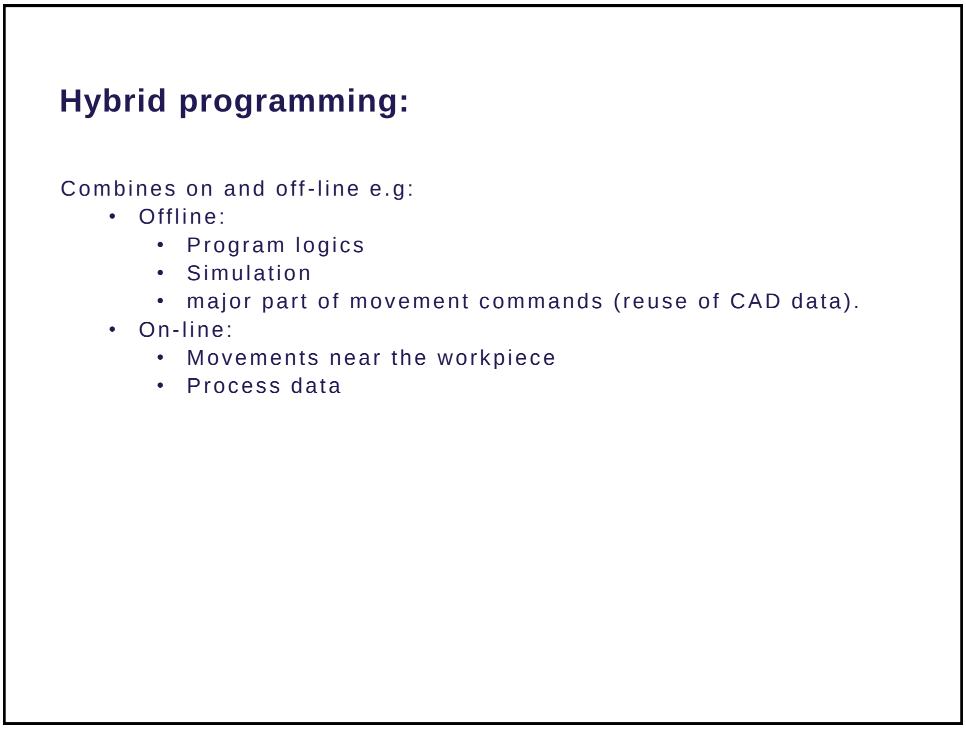
Off-line programming

Advantages:

- Production can continue while programming.
- Effective programming of program logics.
- Effective programming of locations.
- Verification of program through simulation and visualization.
- Well documented through simulation model.
- Reuse of existing CAD data.

Disadvantages

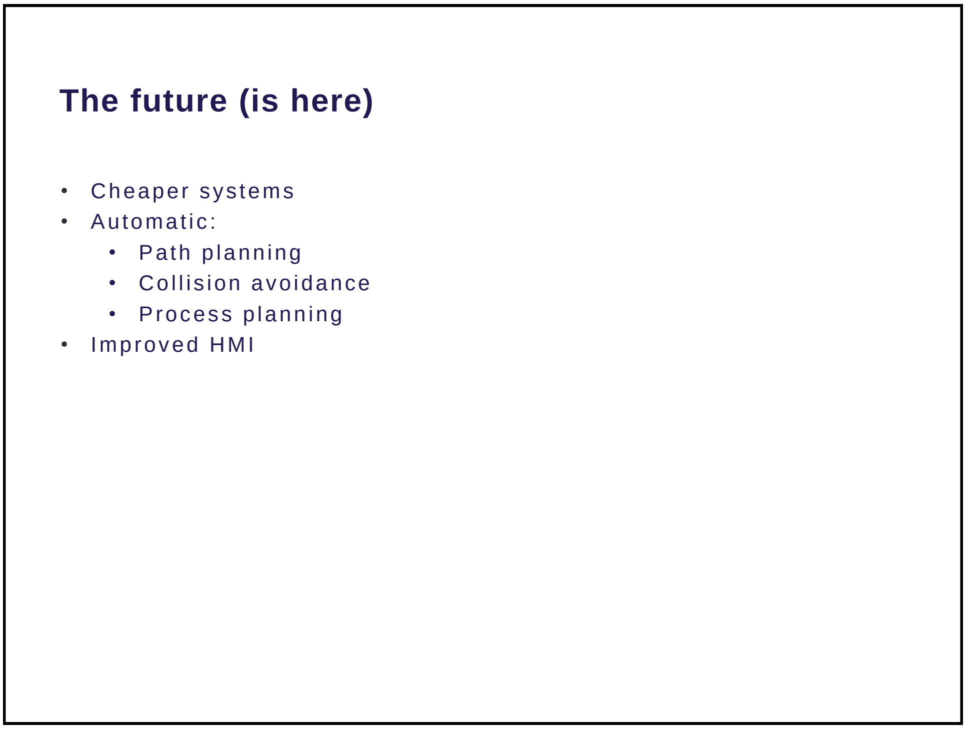
- Often expensive off-line programming system.
- Models are needed and has to correspond to reality
- Takes time to learn



Hybrid programming:

Combines on and off-line e.g:

- Offline:
  - Program logics
  - Simulation
  - major part of movement commands (reuse of CAD data).
- On-line:
  - Movements near the workpiece
  - Process data



The future (is here)

- Cheaper systems
- Automatic:
  - Path planning
  - Collision avoidance
  - Process planning
- Improved HMI

AALBORG UNIVERSITY

DENMARK