Introduction to Industrial Robot Programming

Sarper ALKAN

sarper@cankaya.edu.tr

What is an industrial robot?

- Designed for:
  - High movement precision
  - High speed
  - High repeatability
- Easy to program
- High safety
- Low maintenance costs
- Flexible

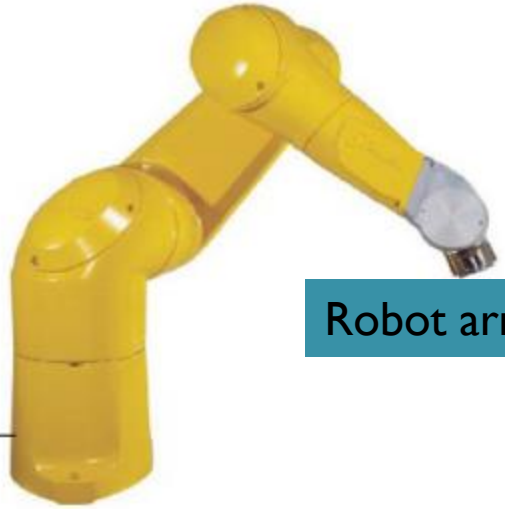
Our robot: Stäubli TX90XL

- 6 axis articulated robot arm
- High-speed, high-resolution servo motors with absolute position sensors at each axis
- Nominal payload: $5\ kg$
- Repeatability: $\mp 0.04\ mm$
- Weight: $116\ kg$
- Arm reach: $1.5\ m$

Our robot: Stäubli TX90XL

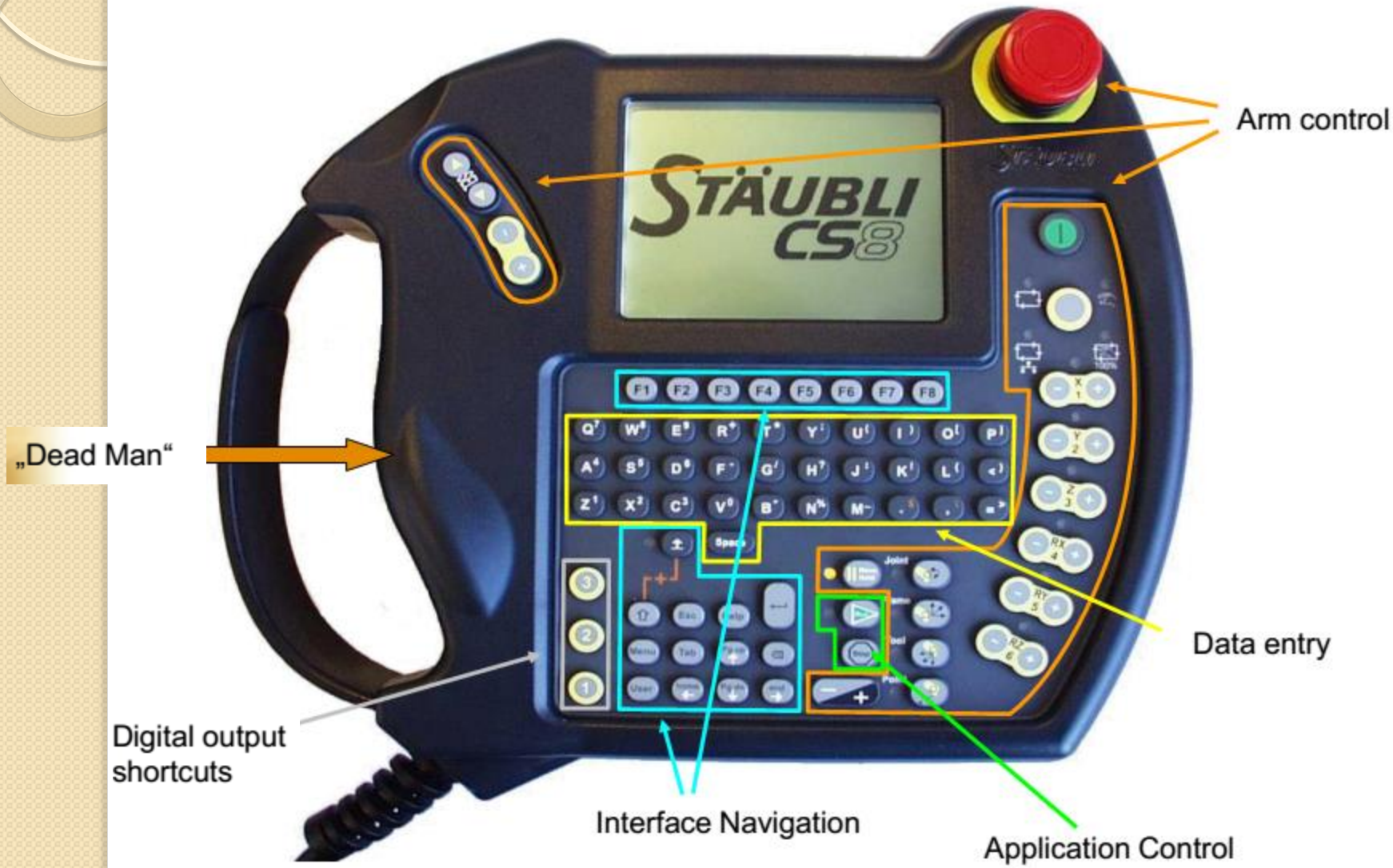
Manual control: Pendant

Cabinet



Robot arm

Pendant


Arm control
„Dead Man“
Data entry
Digital output shortcuts
Interface Navigation
Application Control

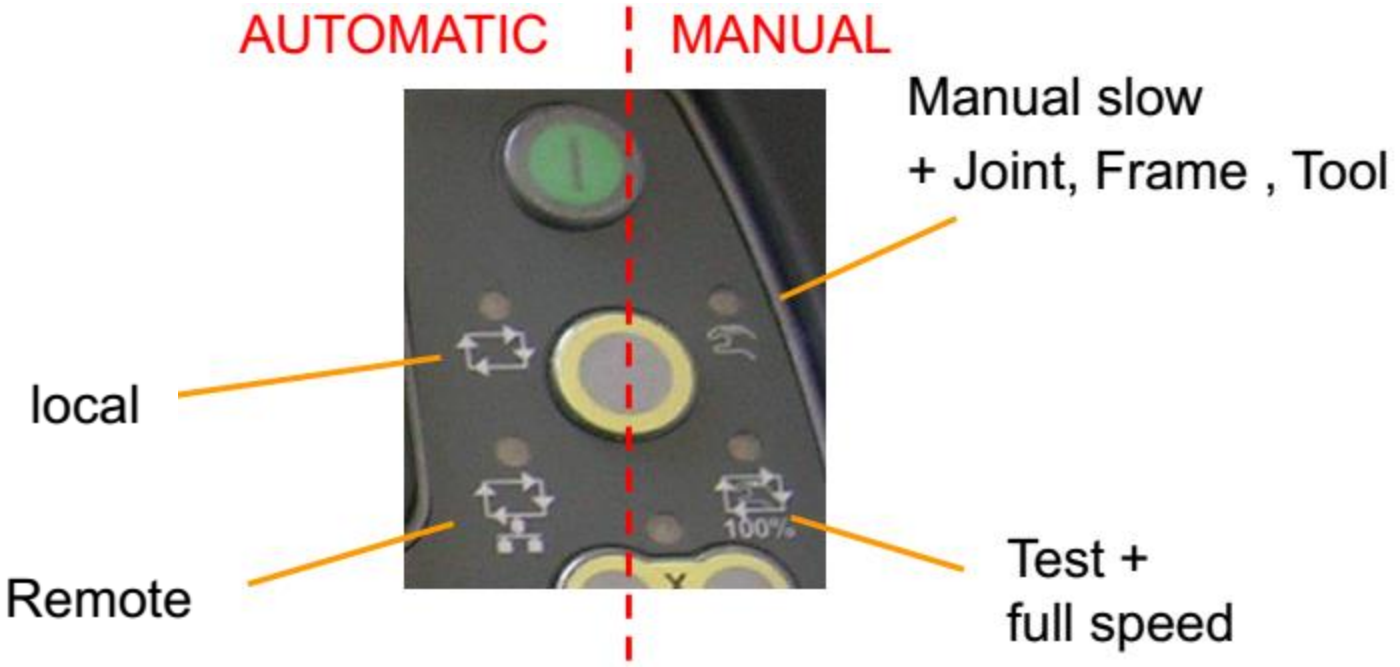
Safety

- Always operate the robot arm at low speed (%10 speed) at first
- Be ready to stop arm motion:
  - Emergency stop (the big red button),
  - Arm power off,
  - Move/hold key,
  - Dead man's switch.

Safety

- Do not use the robot pendant without supervision.
- Do not run your program without supervision.

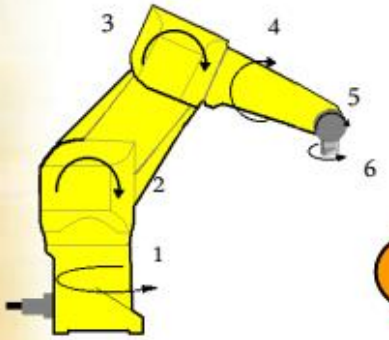
Pendant operations



Manual movement

- Joint move: Moves the robot by rotating the individual joints
  - Cannot do linear movement
  - Is not subject to singularity
- Frame move: Moves the robot in a specified coordinate frame
  - Can do linear movement
  - Subject to singularity
- Tool move: Moves the robot with respect to the tool coordinate frame
  - Can do linear movement
  - Subject to singularity

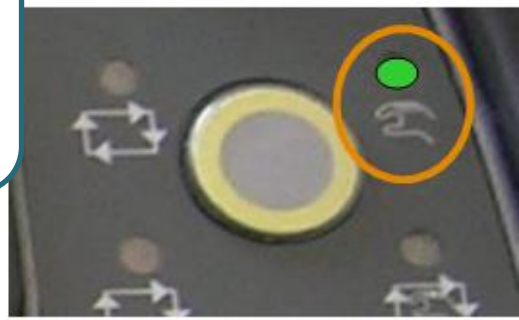
Manual movement with the joint move



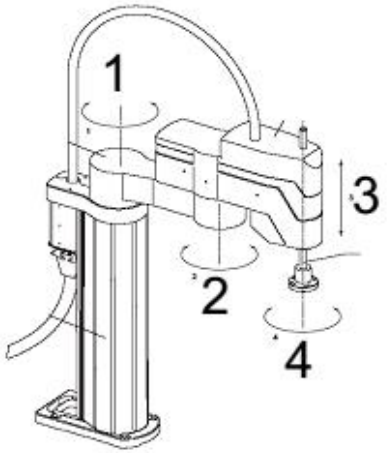
1 Joint

2 Speed

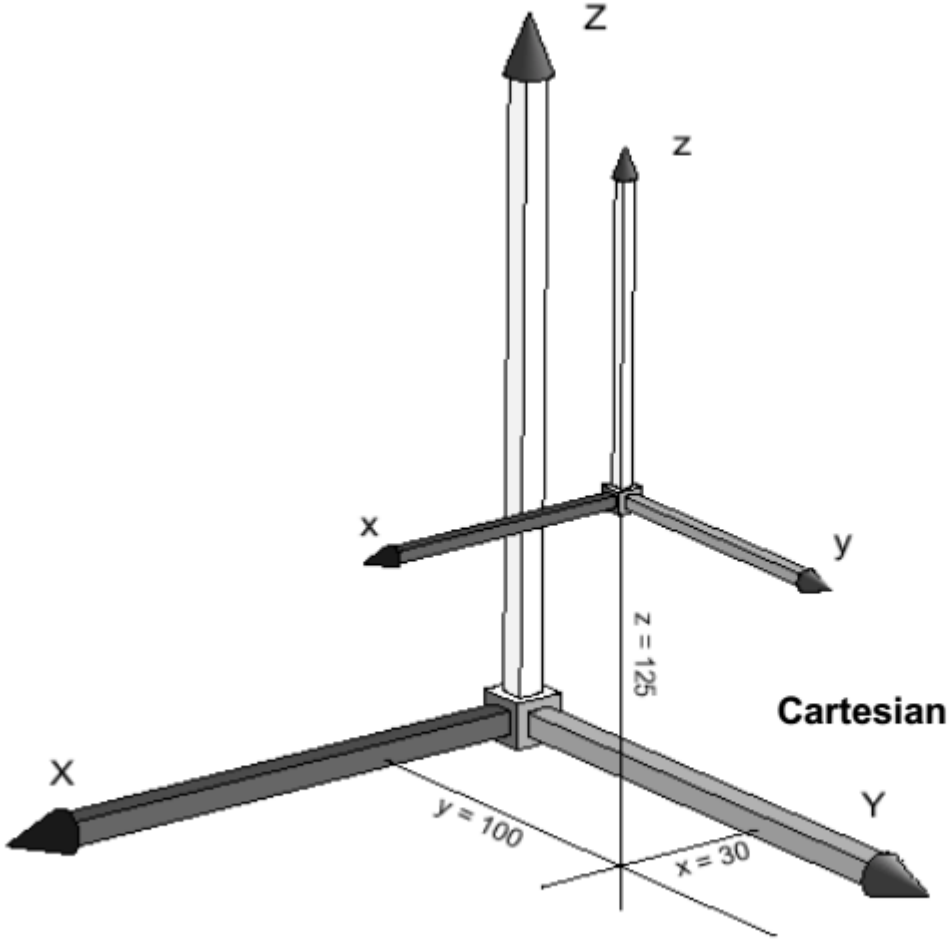
3



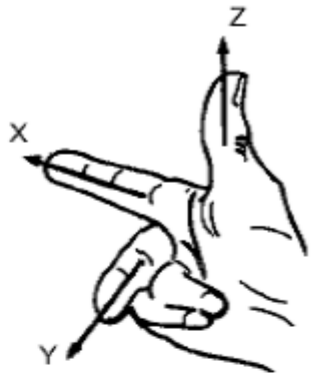
! Requested !



Manual movement with the frame move: Coordinate system



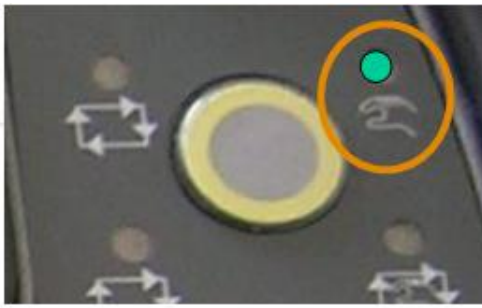
Cartesian Coordinate



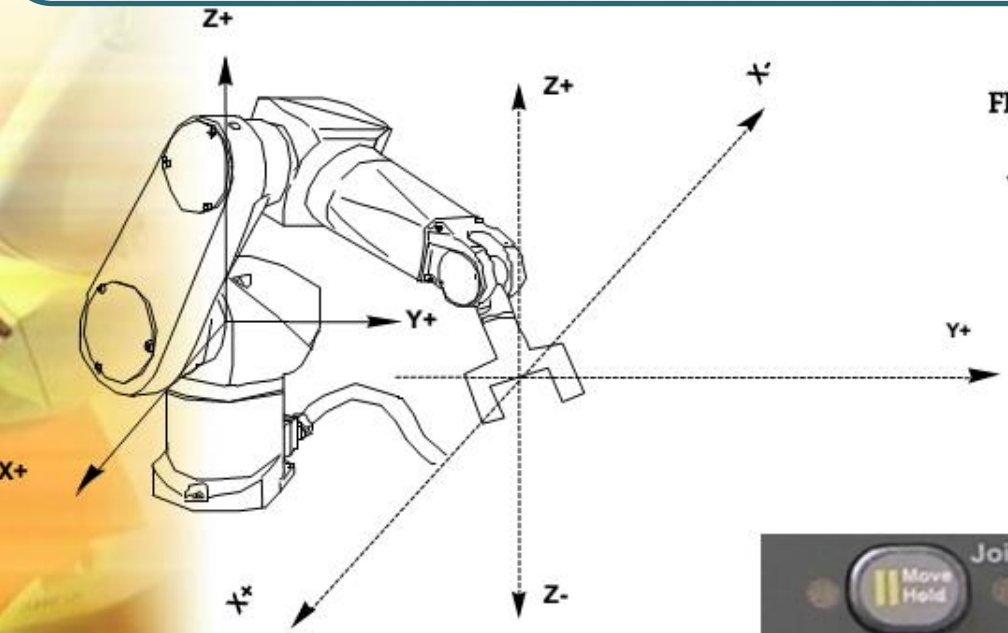
!! Right Hand !!

3 axis perpendicular each other

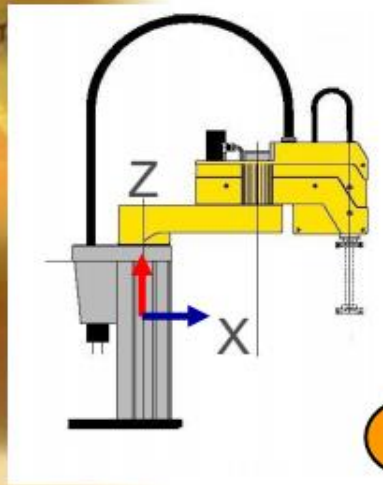
Manual movement with the frame move: Translation



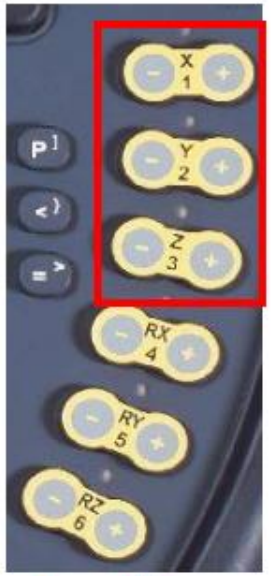
! Requested !



1 Frame



2 Speed

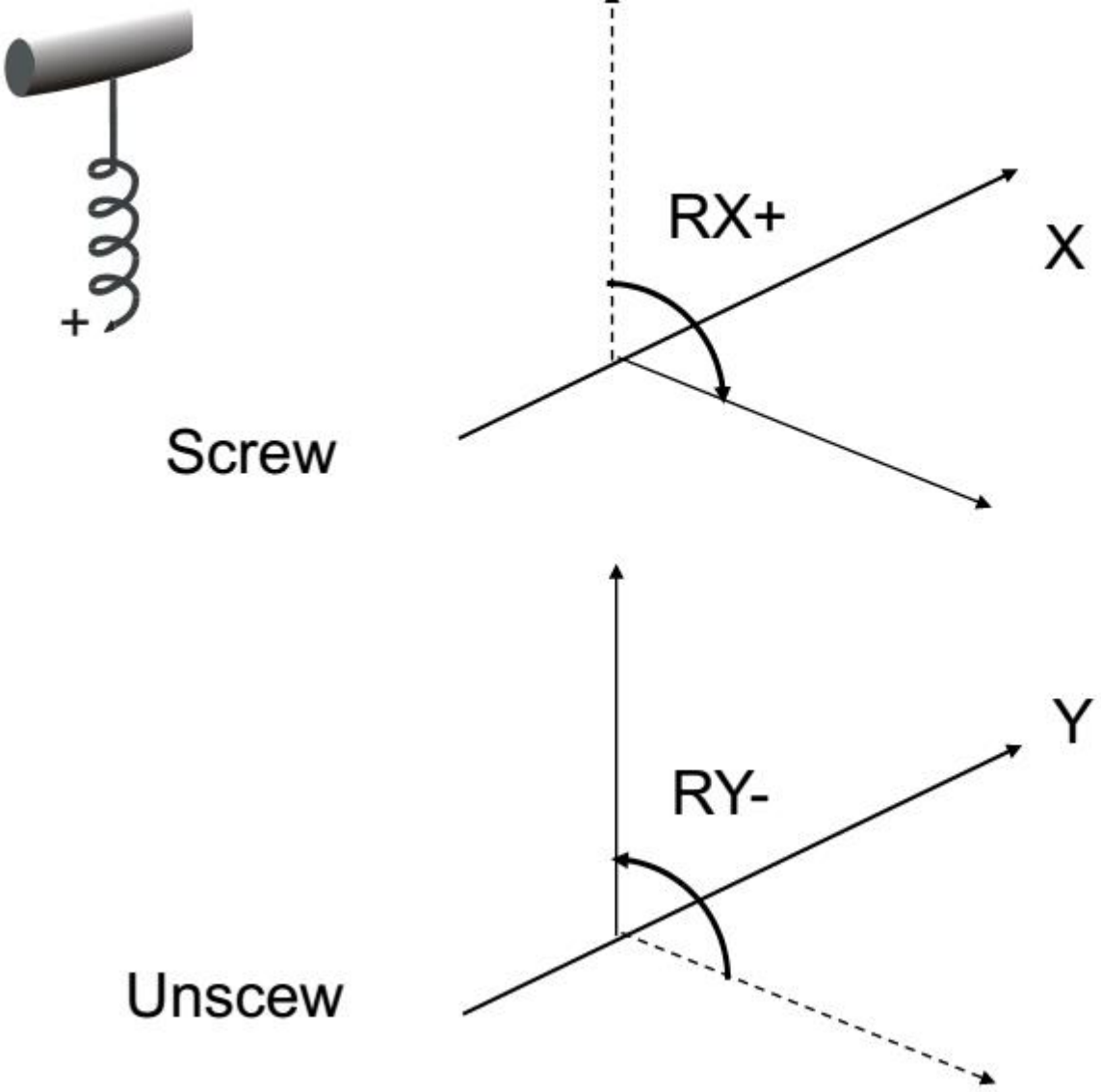


Translation

- +

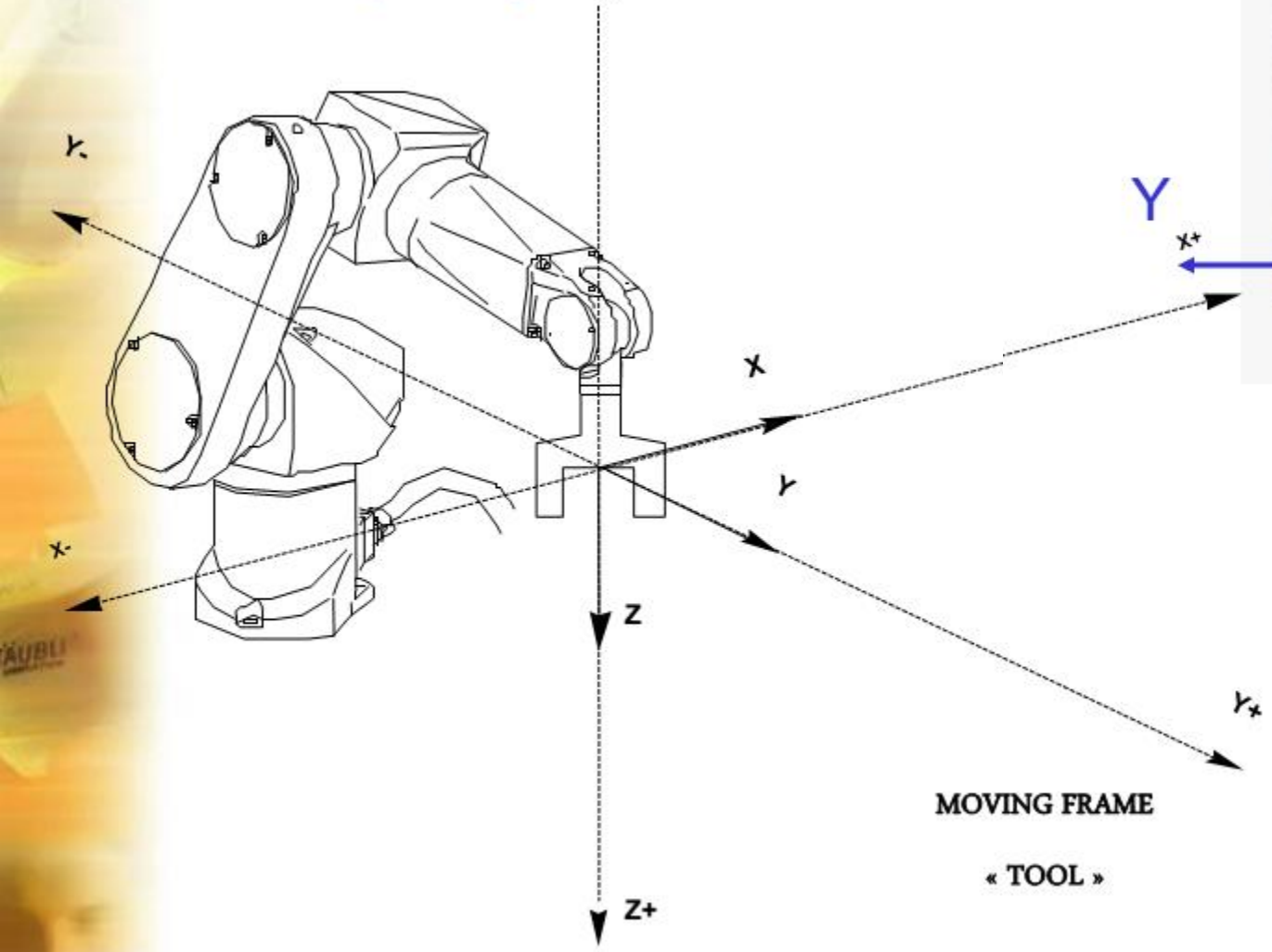
3

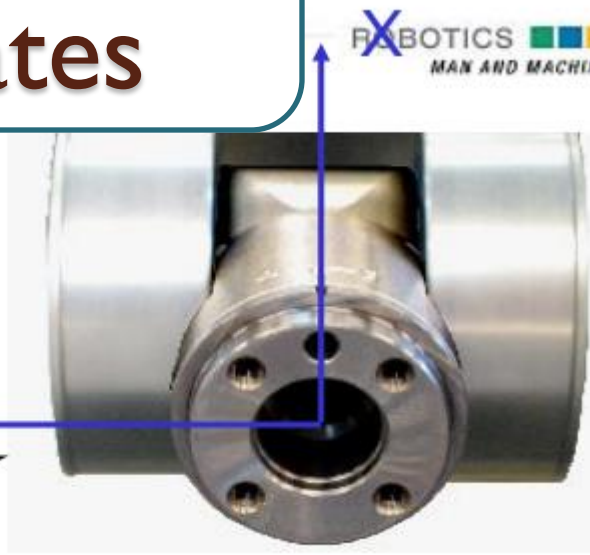
Manual movement with the frame move: Rotation

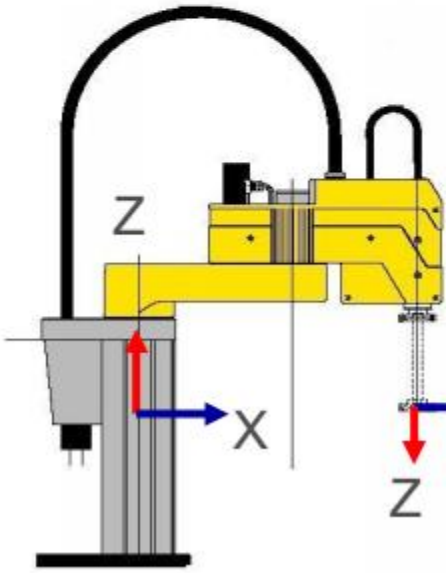


Rotation

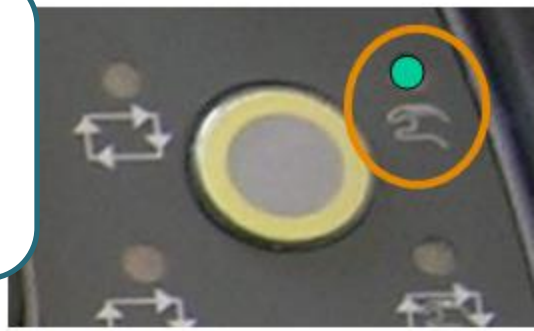
Manual movement with the tool frame: Coordinates



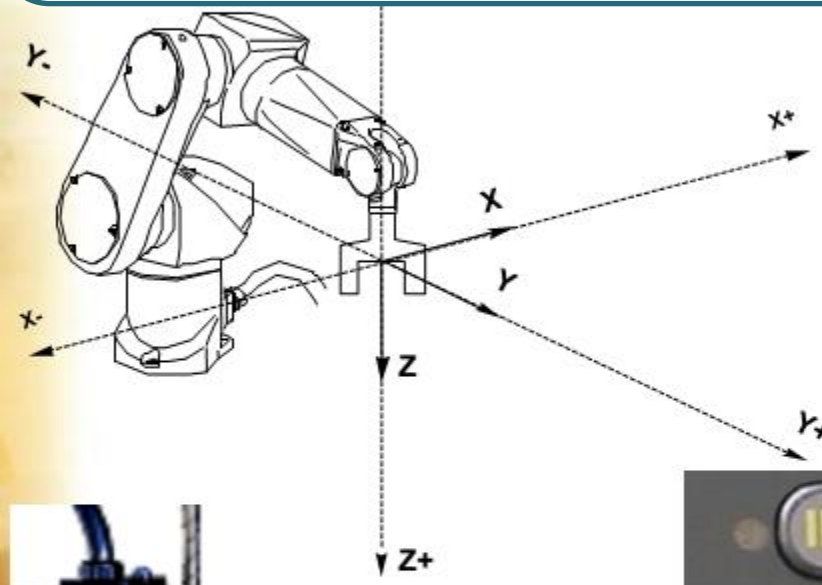




Manual movement with the tool frame:



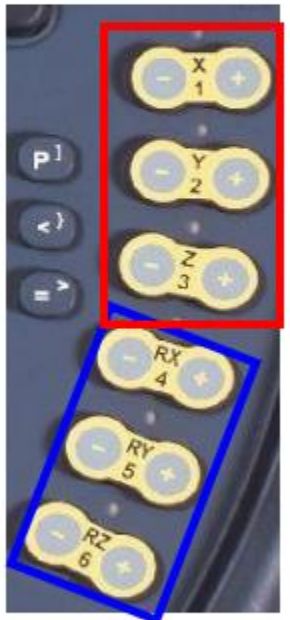
! Requested !



1

Tool

2 Speed



Translation

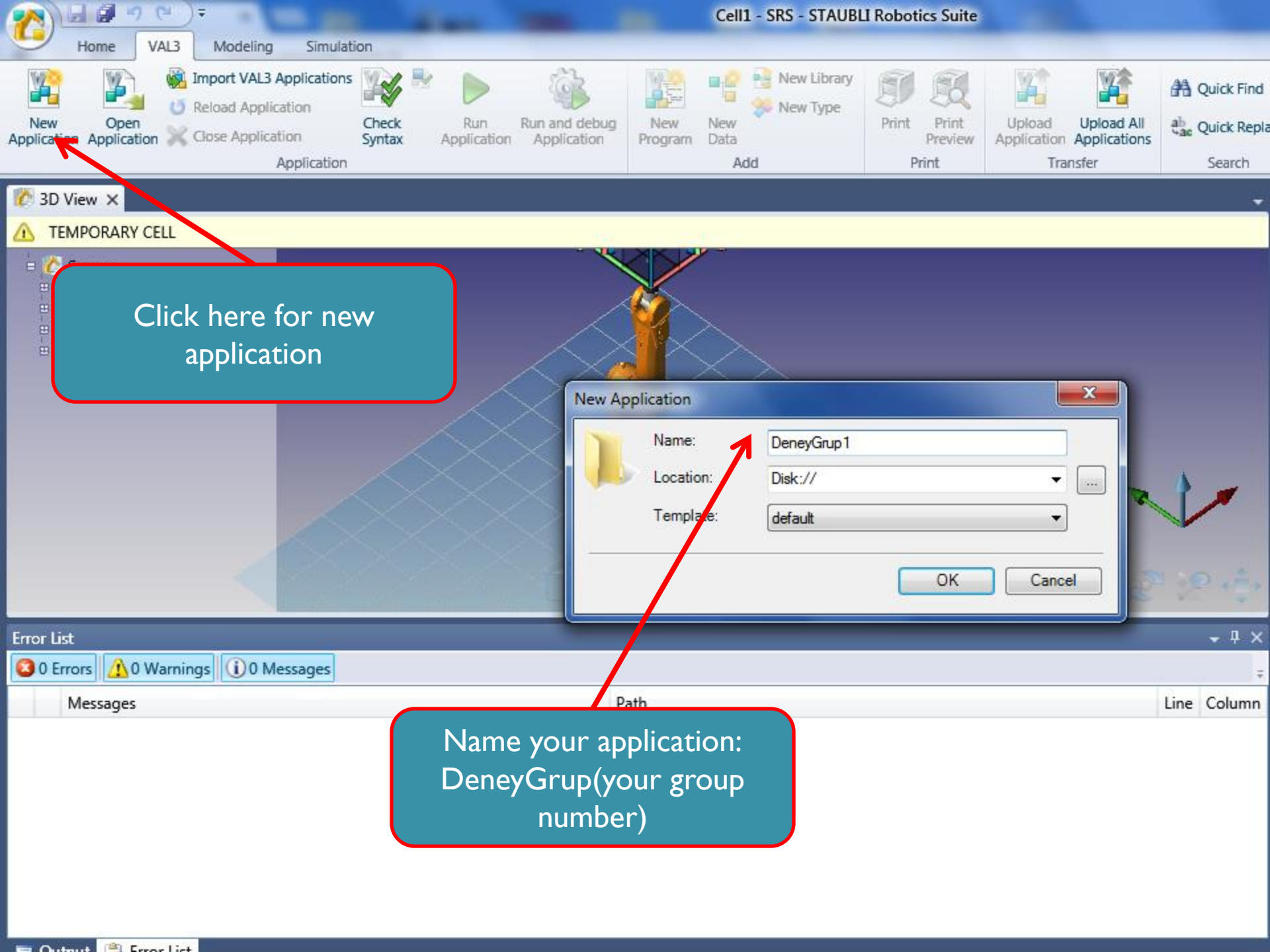
\- +

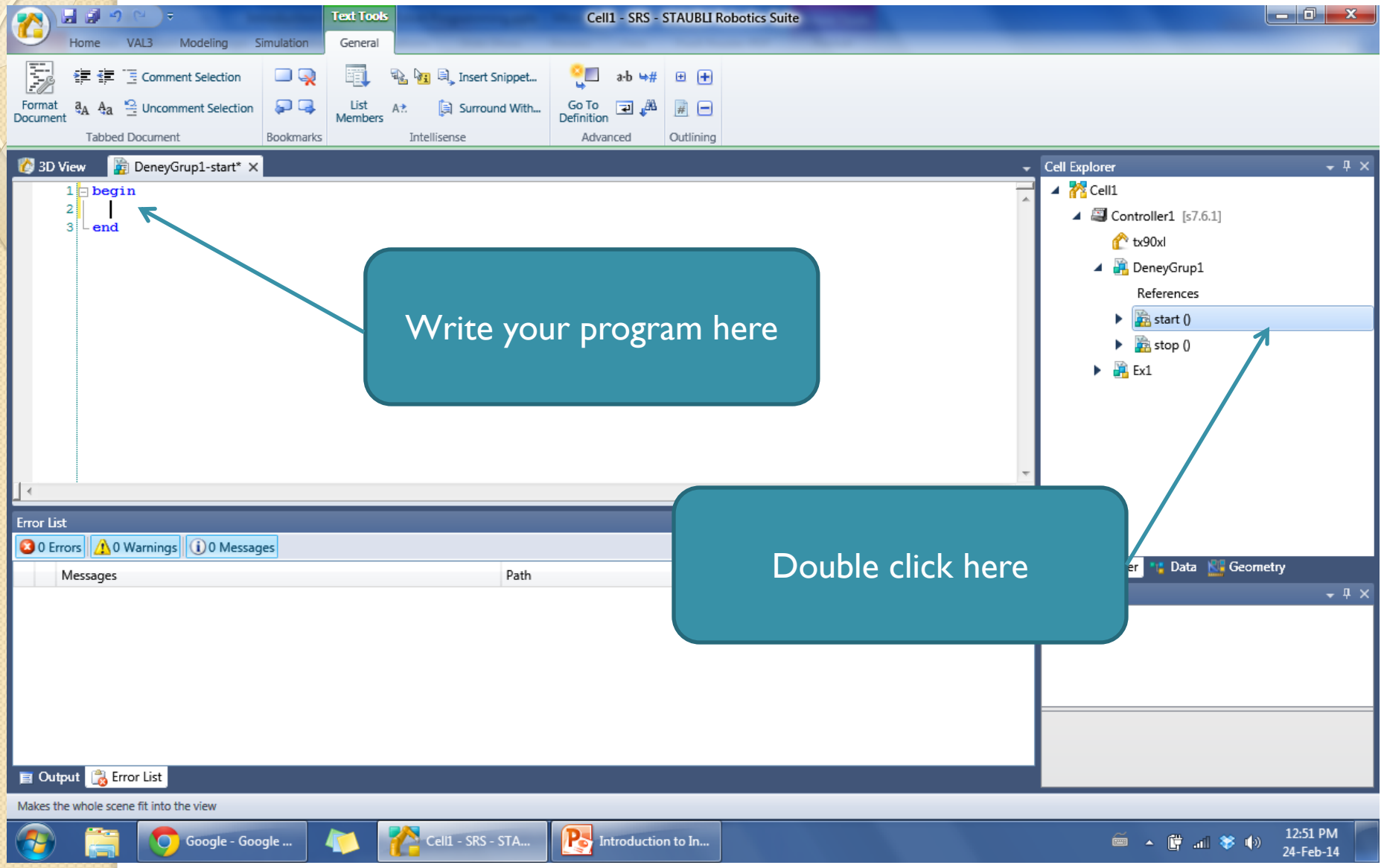
Rotation

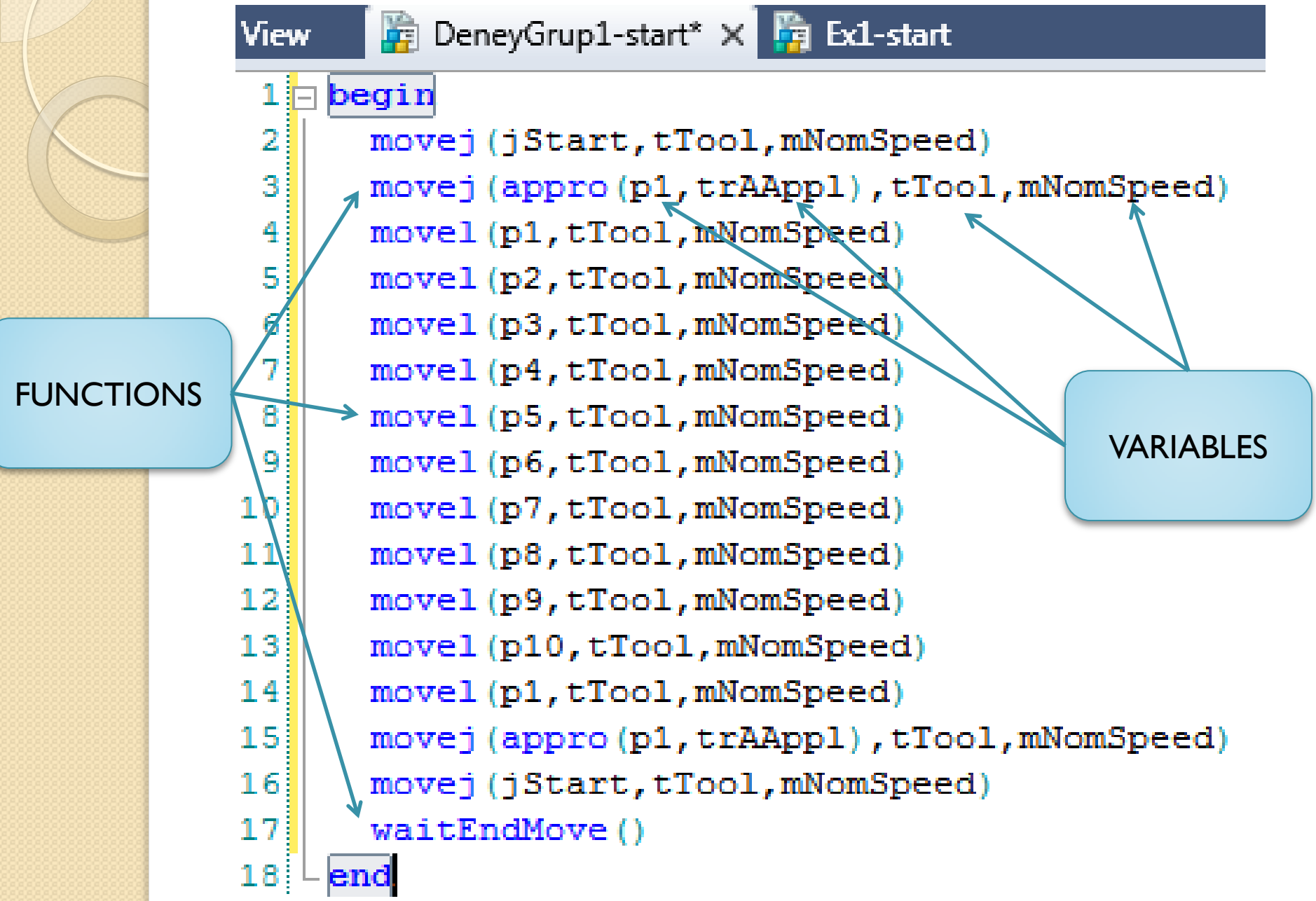
3

Creating applications with VAL3:

- Need:
  - VAL3 installed
  - A ‘cell’ defined (controller, robot, tool)
  - Licence (licence flash-disk attached to the computer)


Cell1 - SRS - STAUBLI Robotics Suite
Home
VAL3
Modeling
Simulation
New Application
Open Application
Import VAL3 Applications
Reload Application
Close Application
Check Syntax
Run Application
Run and debug Application
Application
New Program
New Data
New Library
New Type
Add
Print
Print Preview
Print
Upload Application
Upload All Applications
Transfer
Quick Find
Search
3D View
TEMPORARY CELL
Click here for new application
New Application
Name:
DeneyGrup1
Location:
Disk://
Template:
default
OK
Cancel
Error List
0 Errors
0 Warnings
0 Messages
Messages
Path
Line
Column
Name your application: DeneyGrup(your group number)


Cell1 - SRS - STAUBLI Robotics Suite
Text Tools
Home
VAL3
Modeling
Simulation
General
Format Document
Comment Selection
Uncomment Selection
Tabbed Document
Bookmarks
List Members
Insert Snippet...
Surround With...
Intellisense
Go To Definition
Advanced
Outlining
3D View
DeneyGrup1-start*
begin
end
Write your program here
Cell Explorer
Cell1
Controller1 [s7.6.1]
tx90xl
DeneyGrup1
References
start ()
stop ()
Ex1
Double click here
Data
Geometry
Error List
0 Errors
0 Warnings
0 Messages
Messages
Path
Output
Error List
Makes the whole scene fit into the view
Google - Google ...
Cell1 - SRS - STA...
Introduction to In...
12:51 PM
24-Feb-14


View
DeneyGrup1-start*
Ex1-start
1 begin
2 movej(jStart,tTool,mNomSpeed)
3 movej(appro(p1,trAAppl),tTool,mNomSpeed)
4 movel(p1,tTool,mNomSpeed)
5 movel(p2,tTool,mNomSpeed)
6 movel(p3,tTool,mNomSpeed)
7 movel(p4,tTool,mNomSpeed)
8 movel(p5,tTool,mNomSpeed)
9 movel(p6,tTool,mNomSpeed)
10 movel(p7,tTool,mNomSpeed)
11 movel(p8,tTool,mNomSpeed)
12 movel(p9,tTool,mNomSpeed)
13 movel(p10,tTool,mNomSpeed)
14 movel(p1,tTool,mNomSpeed)
15 movej(appro(p1,trAAppl),tTool,mNomSpeed)
16 movej(jStart,tTool,mNomSpeed)
17 waitEndMove()
18 end
FUNCTIONS
VARIABLES

Applications: Structure

- Variable types:
  - pointRx: a point location in cartesian coordinates (p1, p2, p3, ...)
  - jointRx: a joint location in joint coordinates (jStart)
  - tool: a tool defined by the user (tTool)
  - trsf: a transformation (trAApl)
  - mdesc: motion descriptor (mNomSpeed)

Applications: Structure

- Functions:
  - movej(joint, tool, mdesc): move to a (point or joint) coordinate with specified tool and motion descriptor
  - movel(point, tool, mdesc): move linearly a point coordinate with specified tool and motion descriptor
  - appro(point, trsf): calculate a transformed point by using a point and a transformation
  - waitEndMove: wait for the current movement to end.

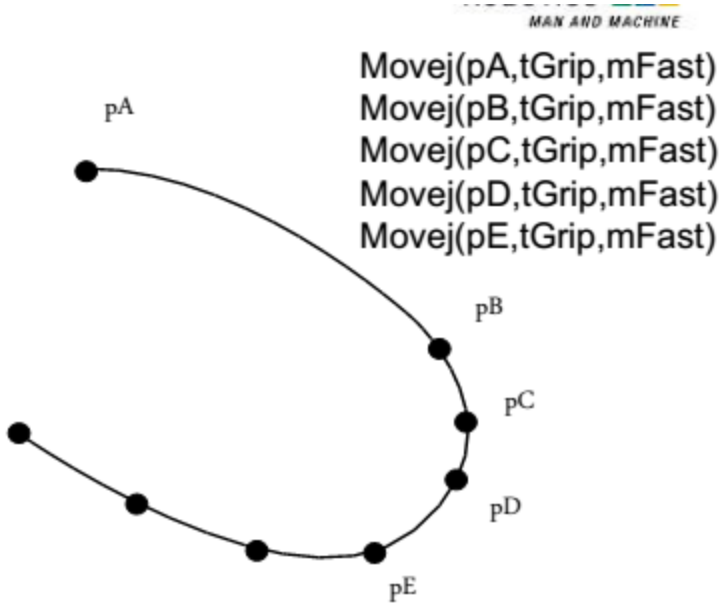
Movement with movej

MOVEMENT : MOVEJ

Movej(point,tool,mdesc)

or

Movej(joint,tool,mdesc)



Joint Interpolation : curved movement

Speed and acceleration described by motion descriptor

No problem of singularity crossing

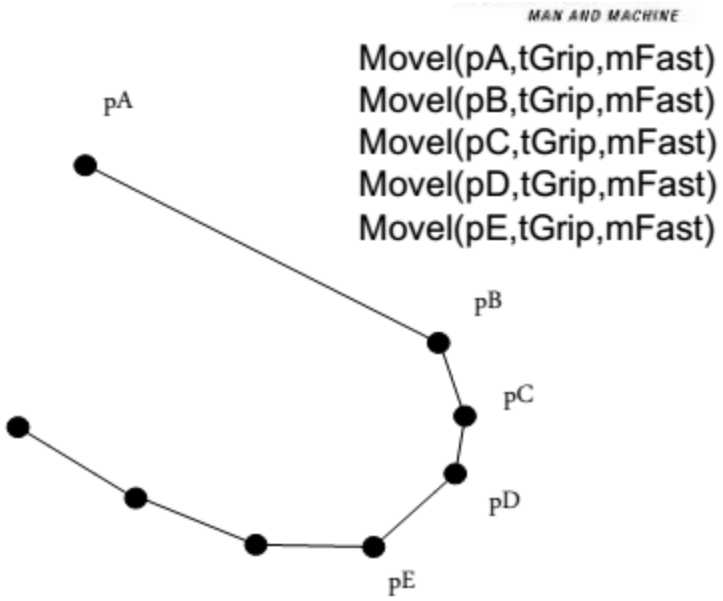
Motion to use if no constraint : Obstacle, insertion, . . .

Movement with movel

MOVEMENT : MOVEL

Movel(point,tool,mdesc)

Not available on JOINT



Cartesian Interpolation : straight line movement

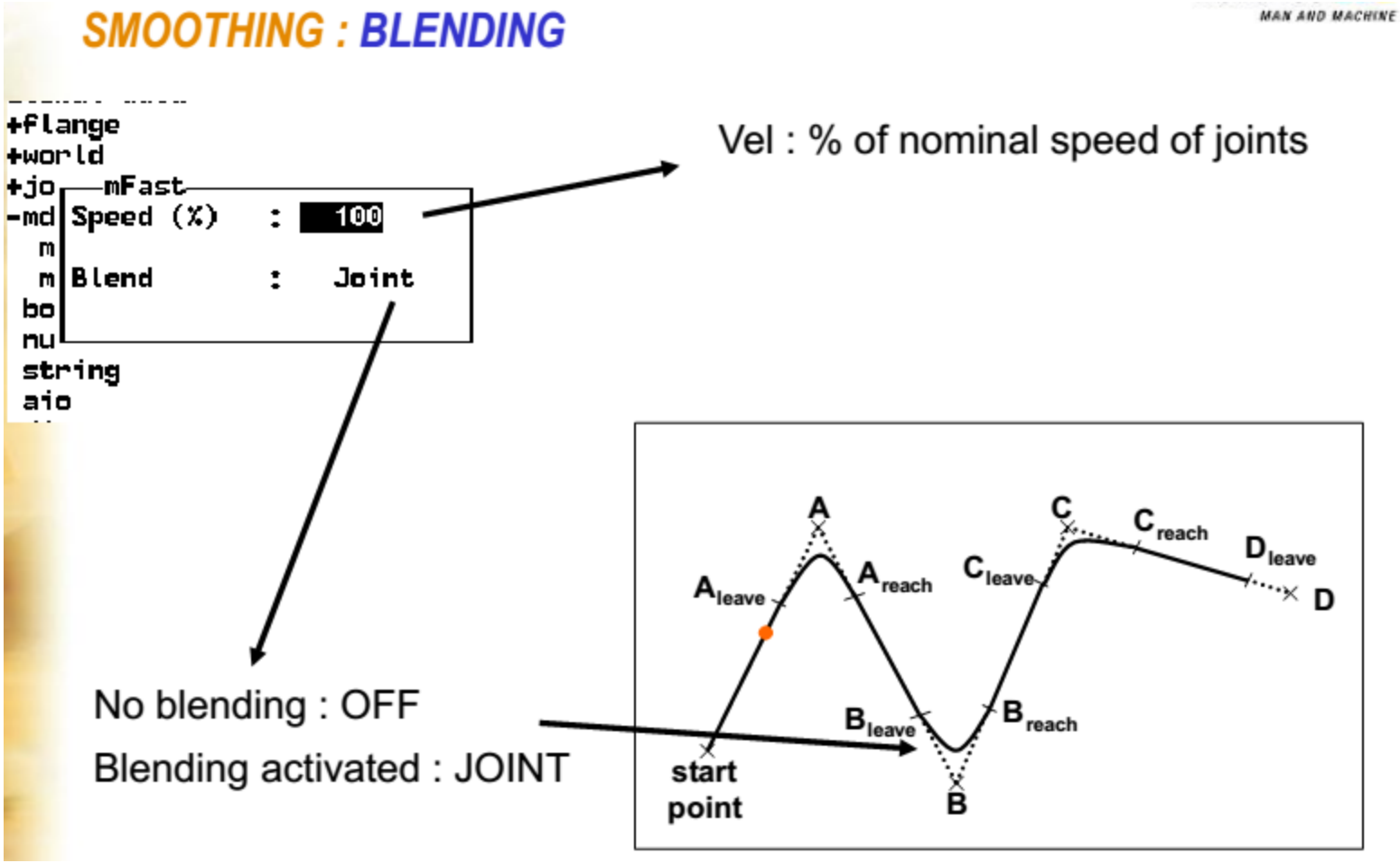
Speed and acceleration described by motion descriptor

Problem of singularity crossing => slow down

Motion to use in case of constraint : obstacle, insertion, process,

. . .

Motion descriptor parameters: blend



Example program:

View

```
1  begin
2      movej(jStart,tTool,mNomSpeed)
3      movej(appro(p1,trAAppl),tTool,mNomSpeed)
4      movel(p1,tTool,mNomSpeed)
5      movel(p2,tTool,mNomSpeed)
6      movel(p3,tTool,mNomSpeed)
7      movel(p4,tTool,mNomSpeed)
8      movel(p5,tTool,mNomSpeed)
9      movel(p6,tTool,mNomSpeed)
10     movel(p7,tTool,mNomSpeed)
11     movel(p8,tTool,mNomSpeed)
12     movel(p9,tTool,mNomSpeed)
13     movel(p10,tTool,mNomSpeed)
14     movel(p1,tTool,mNomSpeed)
15     movej(appro(p1,trAAppl),tTool,mNomSpeed)
16     movej(jStart,tTool,mNomSpeed)
17     waitEndMove()
18 end
```

Move to the starting location

Approach to point1

Move to point 1

Move to point 2

- Use as many points as necessary

Move to point 10

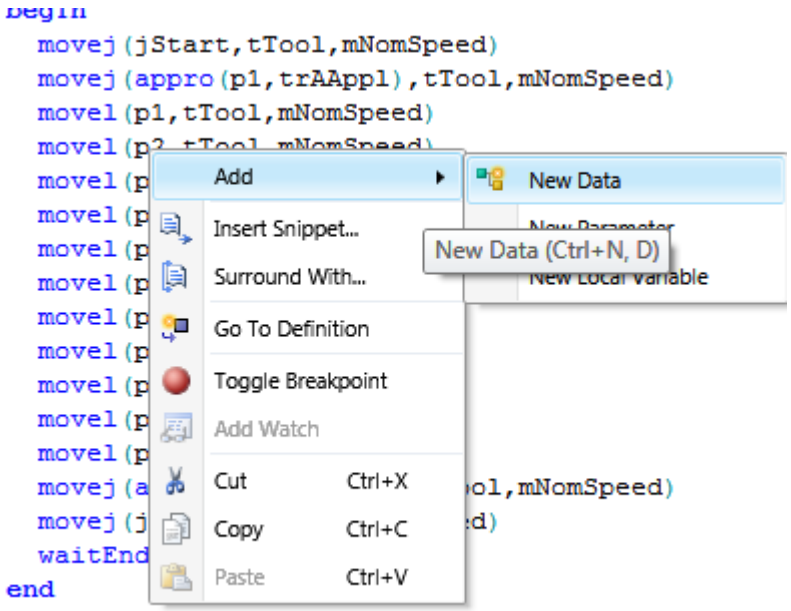
Move to point 1

Approach to point1

Move to the starting location

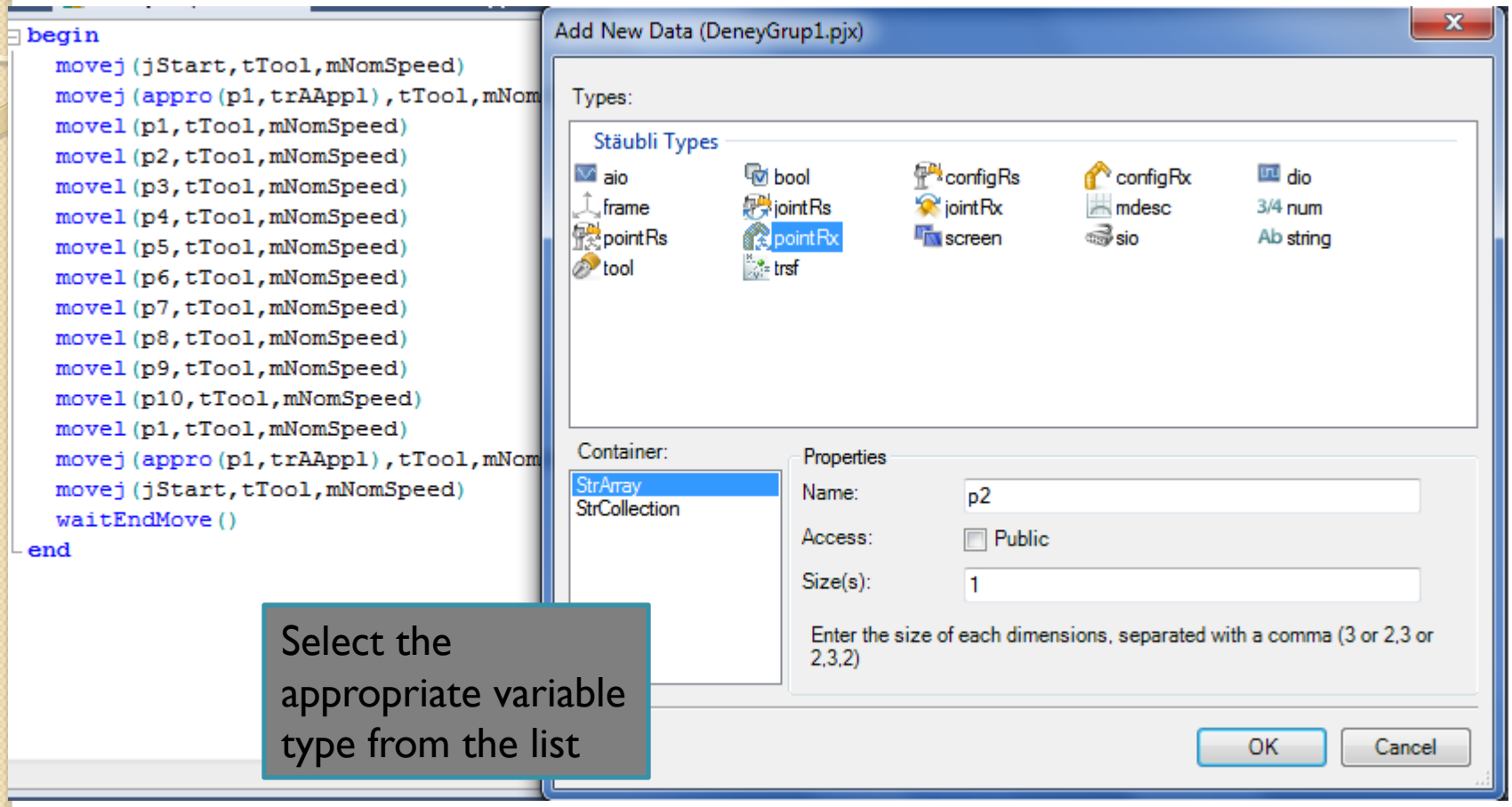
Wait for the movement to end

Adding variable definitions:



Right click on the variable to add (p2), select add, new data

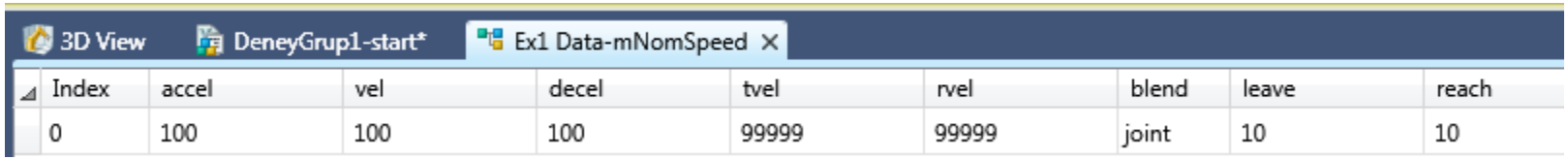
Adding variable definitions:



Editing mdesc and trsf

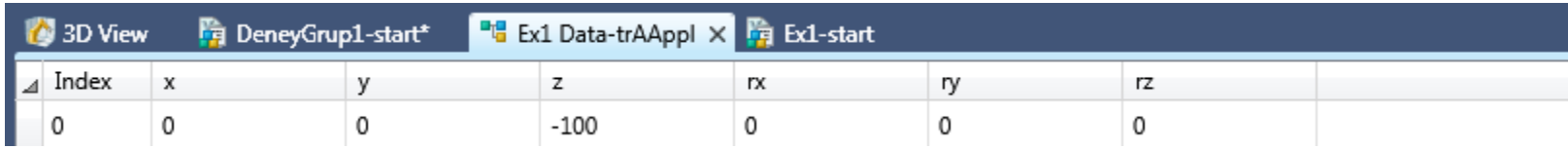
Right click on a variable and select 'Go to definition' to edit the variable. Edit only mdesc and trsf this way.

- mdesc: mNomSpeed



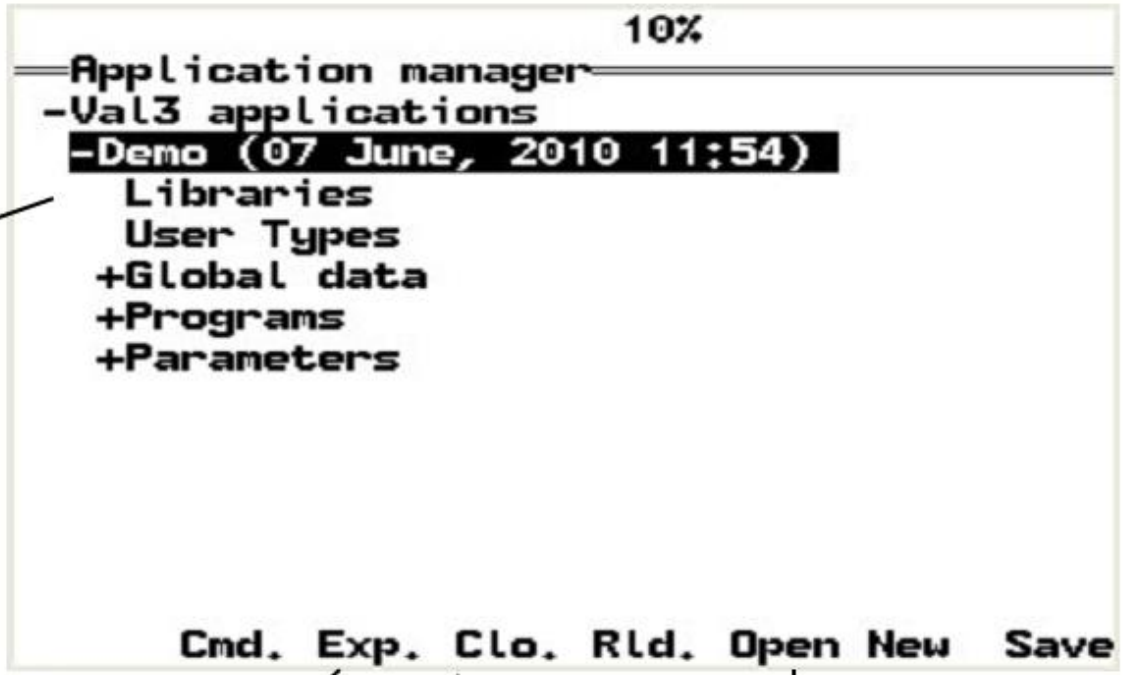
Index	accel	vel	decel	tvel	rvel	blend	leave	reach
0	100	100	100	99999	99999	joint	10	10

- trsf: trAAplI



Index	x	y	z	rx	ry	rz
0	0	0	-100	0	0	0

Applications: using pendant



Application name
(save date)

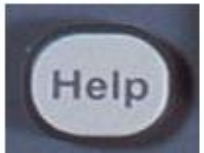
Contextual Menu

Export :
Copy

Close

Reload
(after SRS transfer)

Open another application

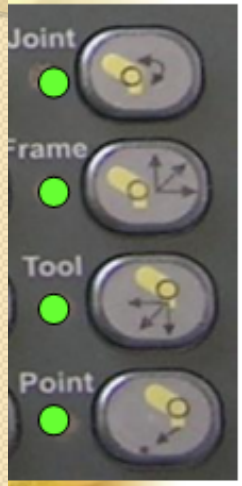


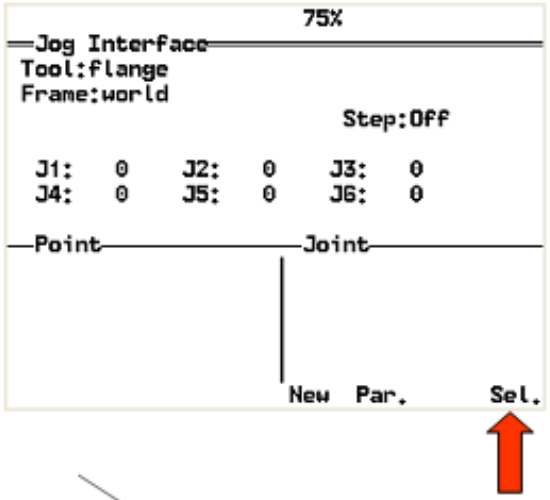
Help on menus

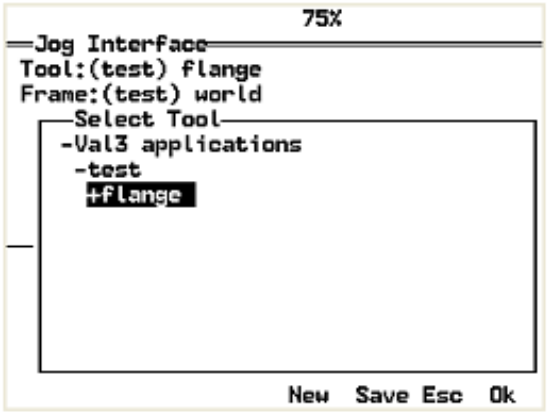
Applications: Teaching with pendant

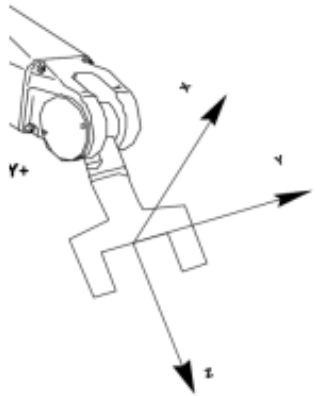
TOOL SELECTION

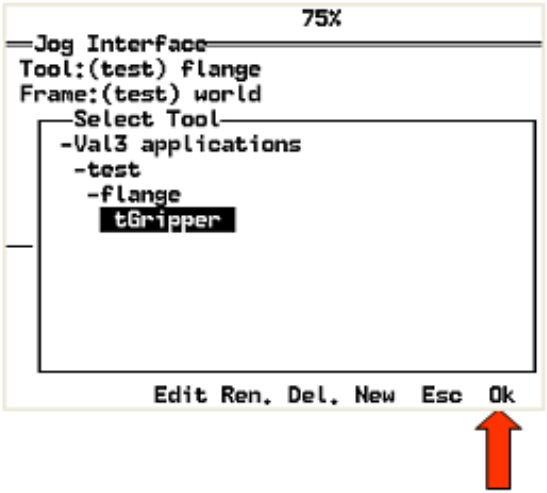
MAN AND MACHINE





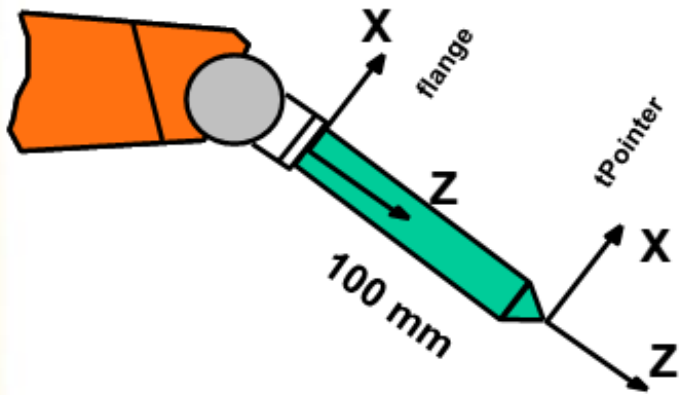






Applications: Teaching with pendant

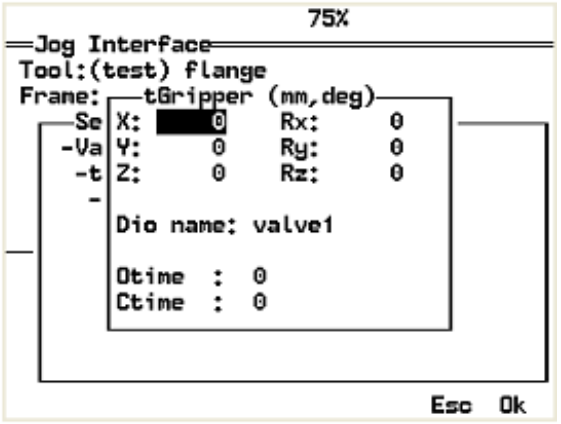
TOOL EDITION



Teach locations

Position and speed control

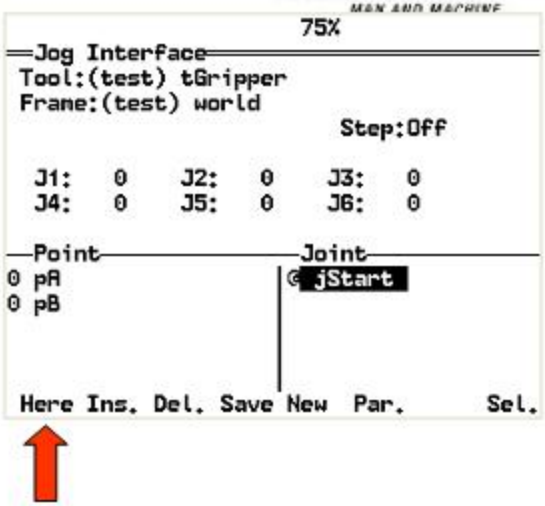
Geometrical adjustments

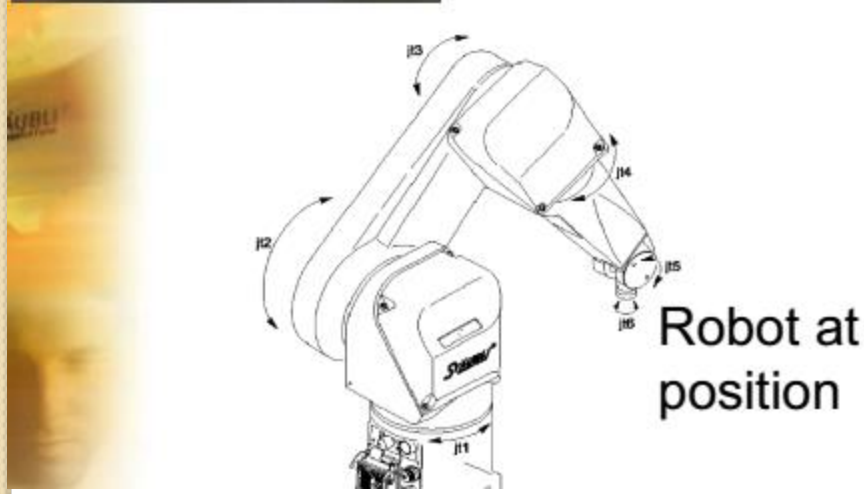


- Geometrical Definition
- Associated digital output
- Delay to open / close in secs.

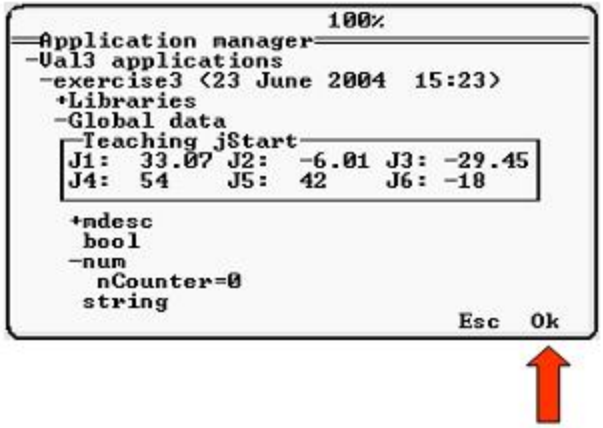
Applications: Teaching with pendant

Move to the desired position
by using the manual control



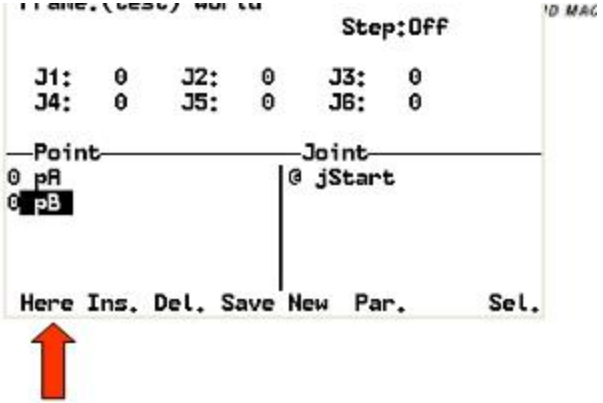


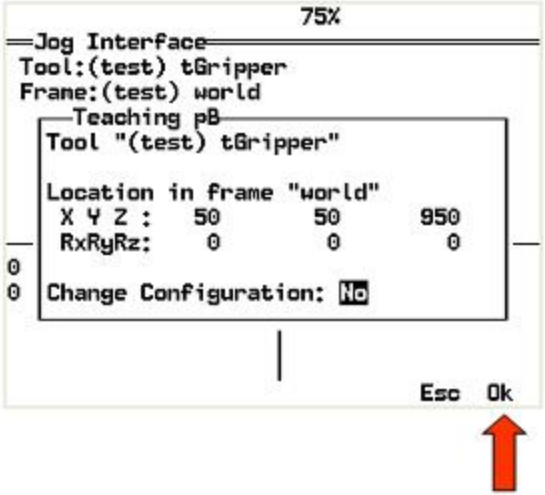
Robot at position

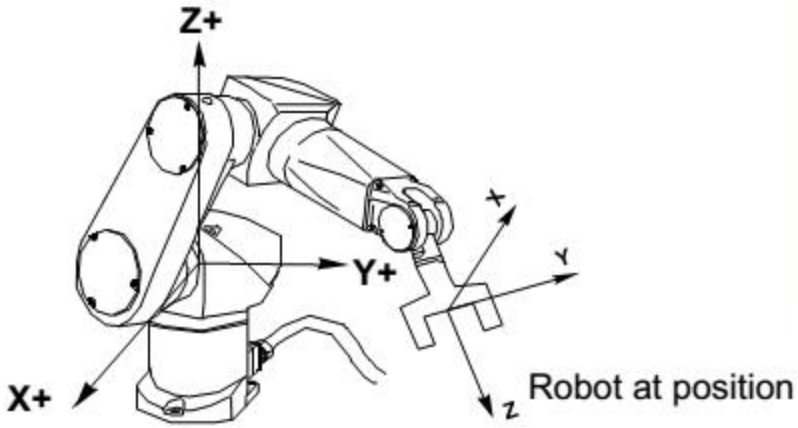


Applications: Teaching with pendant

Move to the desired position
by using the manual control







Robot at position

Elements of the jog interface

JOG INTERFACE

MAN AND MACHI

Points of an application

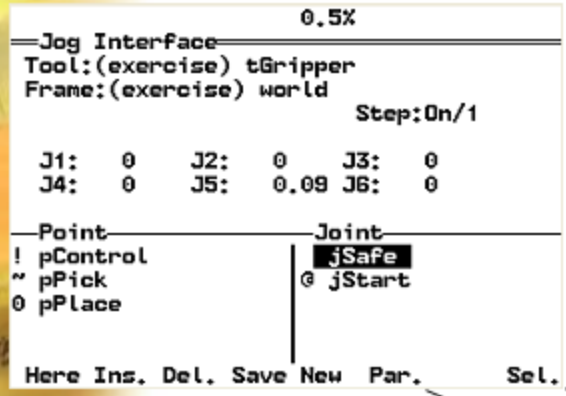
Symbols in front of points name : VAL3 6.5+

0 : Point not taught, coordinates = 0

@ : Robot at position with current tool (approx 0.01 mm)

~ : Robot close to position (approx. 1 mm)

! : Position not reachable with current tool



Select tool or frame

Parameters step mode

New point

Insert point In an array

Teach current position

Save current application (select by the tool)

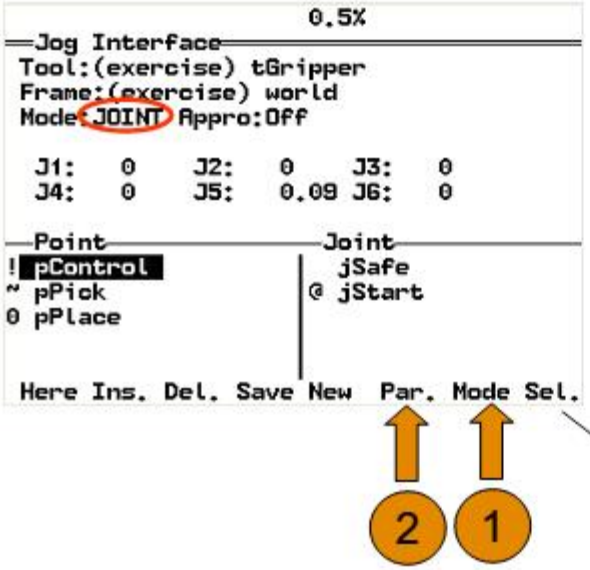
delete point (check if not used) or array element

Re-teaching an existing point:

POINTS RE-TEACHING

- Possible to verify / re-teach points : POINT mode

1. movement type = Mode : JOINT
2. with/without APPRO : direction selection + distance



motion Control

JOINT→LINE→ALIGN

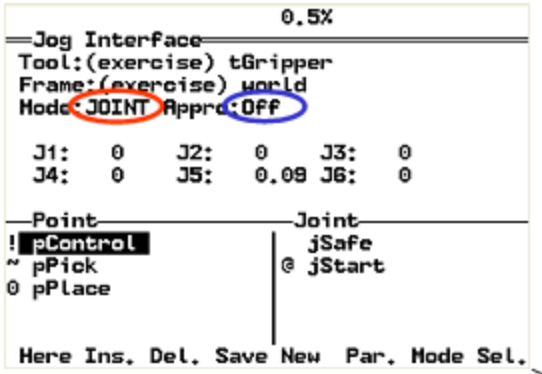
tool select

Re-teaching an existing point:

POINTS RE-TEACHING

- Possible to verify / re-teach points : POINT mode

1. movement type = Mode : LINE
2. with/without APPRO : direction selection + distance



motion Control

JOINT→LINE→ALIGN

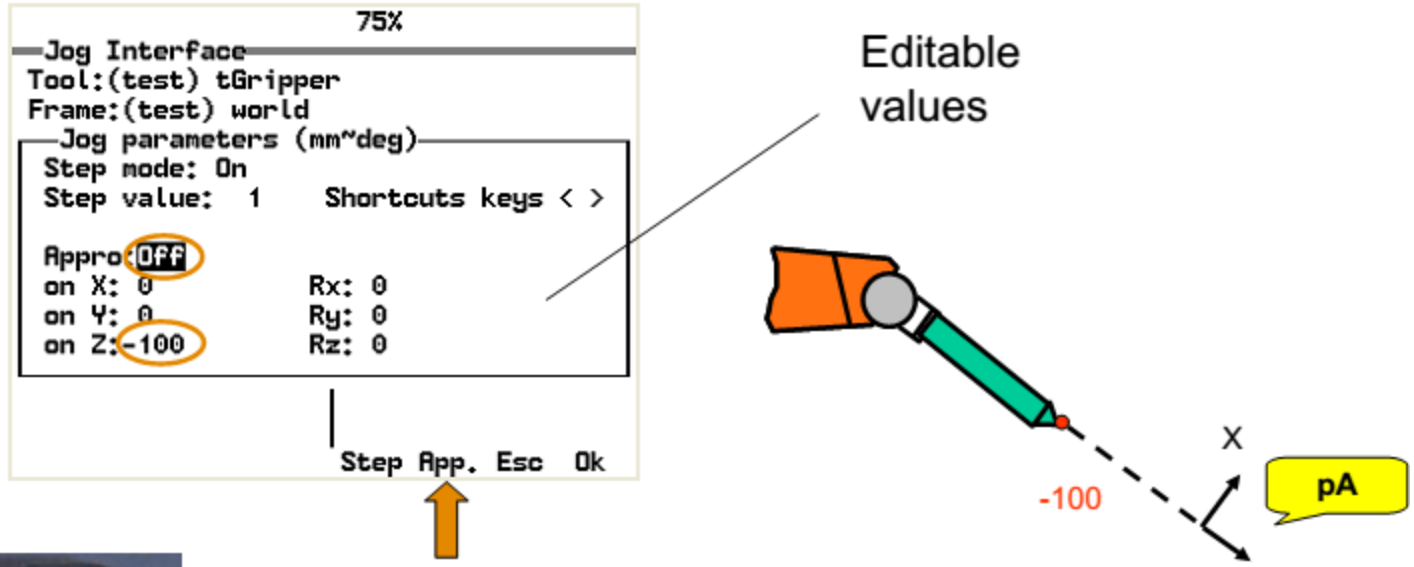
2 1

tool select

Re-teaching an existing point:

POINTS RE-TEACHING

Verify / re-teach points : APPRO mode



motion Control

- Shortcuts : APPRO ON/OFF = F6 + F6

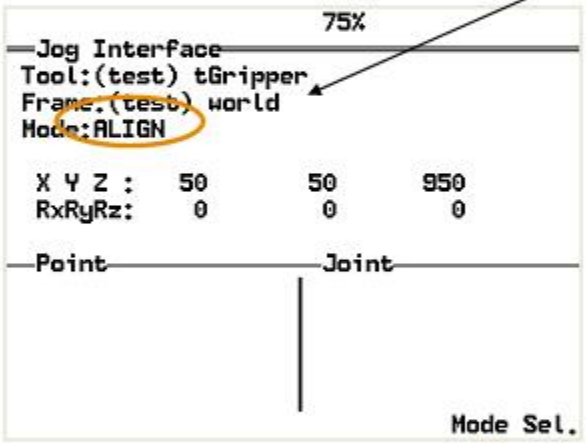
Re-teaching an existing point:

RE TEACHING : ALIGN

Possible to verify / re-teach points : POINT mode

3 move Type = Mode : ALIGN (related to current frame)

Align Z of the TOOL Z on the closest axis of current frame.

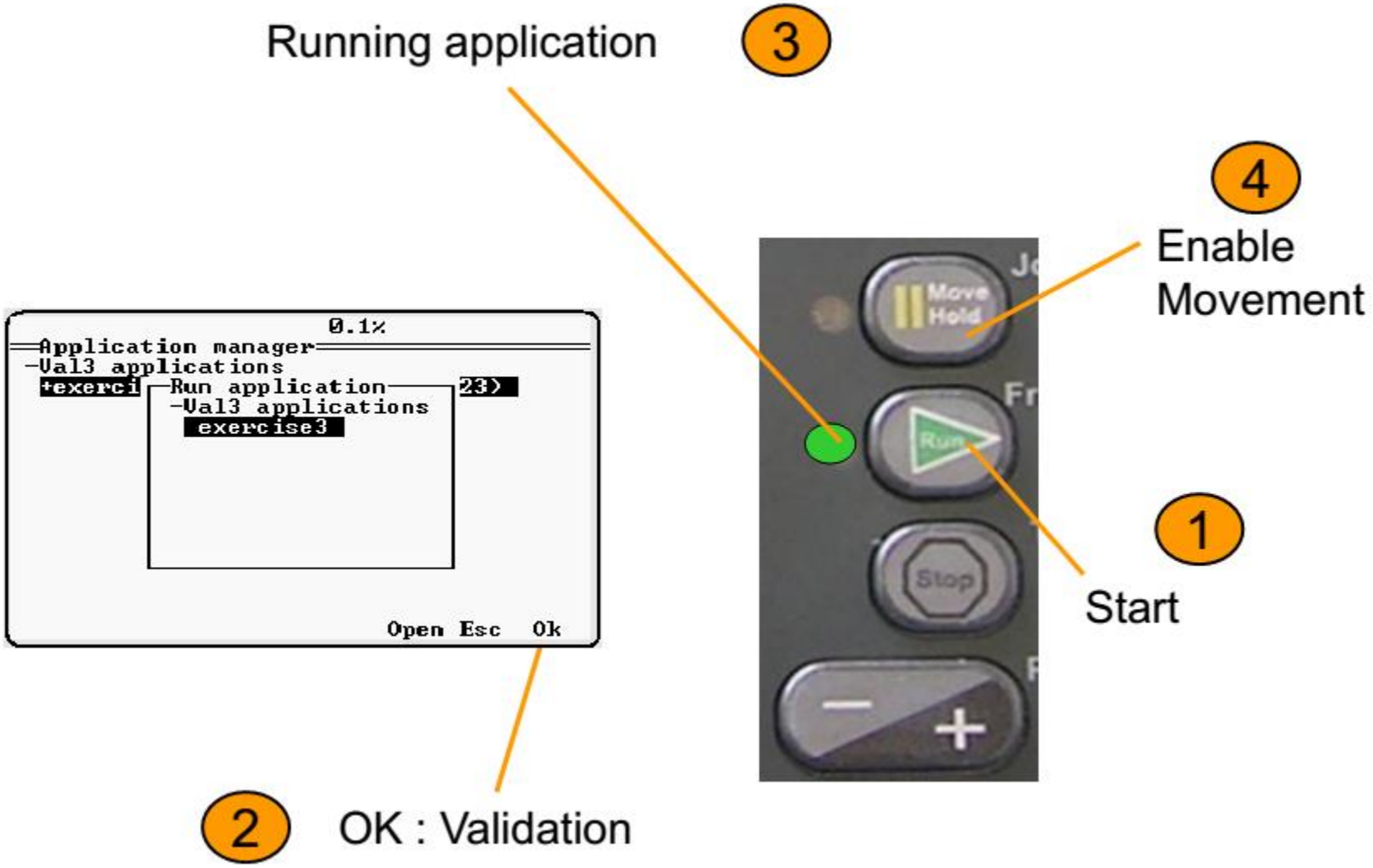


3

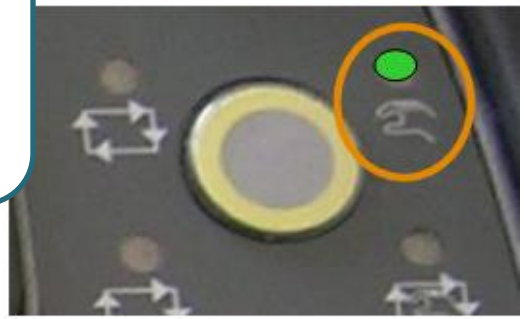
JOINT→LINE→ALIGN

motion Control

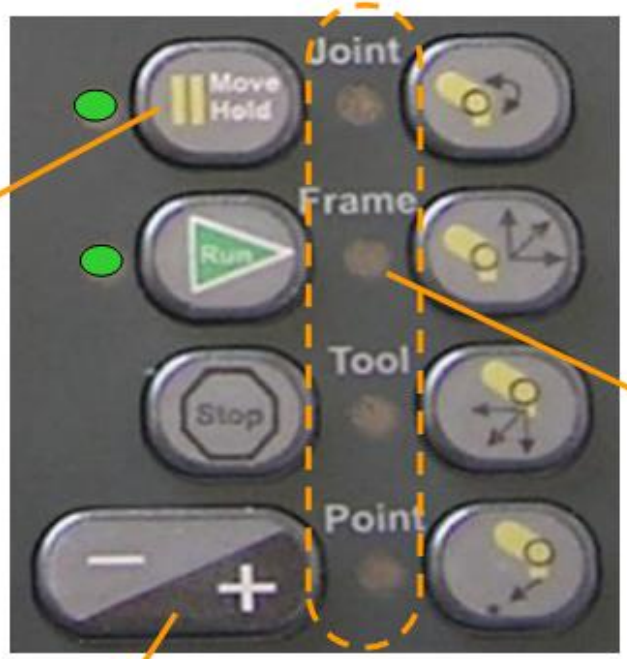
Running applications:



Running applications: Slow running



Cell opened



2

Motion when keep pressed

1

Remove current selected mode = press button

Speed adj.

Create your own application:

- Get together with a group of 3-4 people (no less than 3 no more than 4)
- Get an appointment from Mr. Volkan Erbay (robotics lab techician) !!do this as soon as possible!!
- Create an application, teach the points, and write a letter of 'MEKATRONİK' by using your application
- Almost everything that you need to create and run the application are in these slides
- You need to complete this assignment in 2 weeks from now.