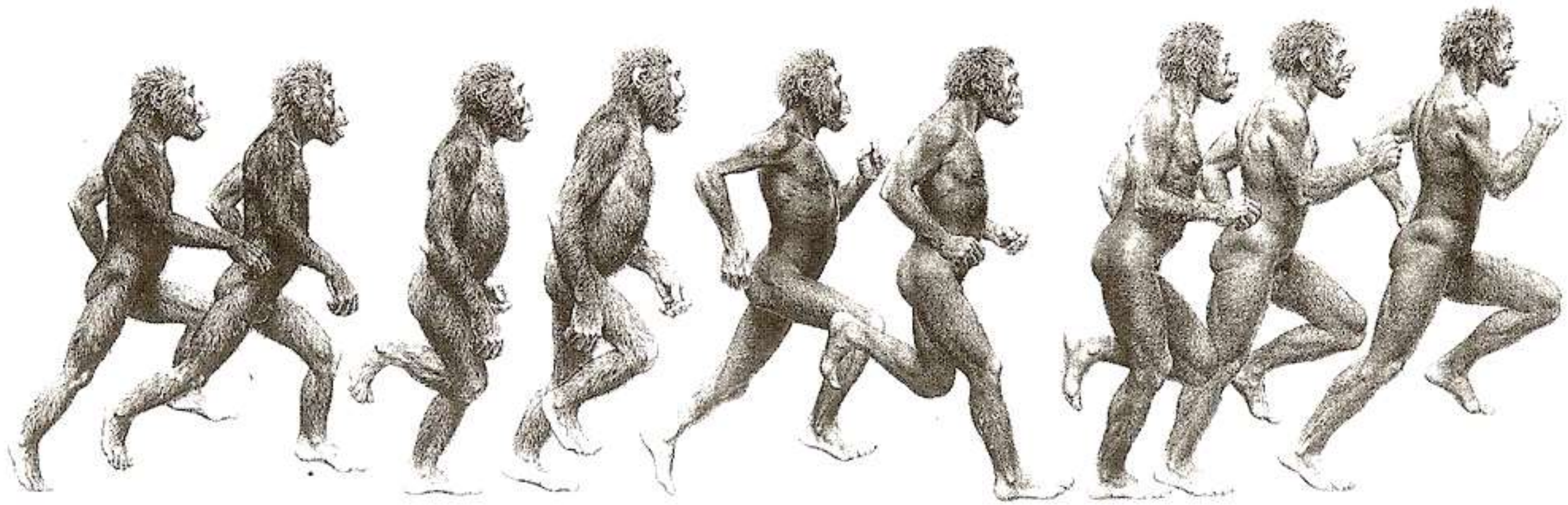
Vision-based Robot Drawing

-- A case study of Human-Machine Interaction by Melfa RV2-SD

Haoran Xie
Obuchi lab
&. Igarashi lab



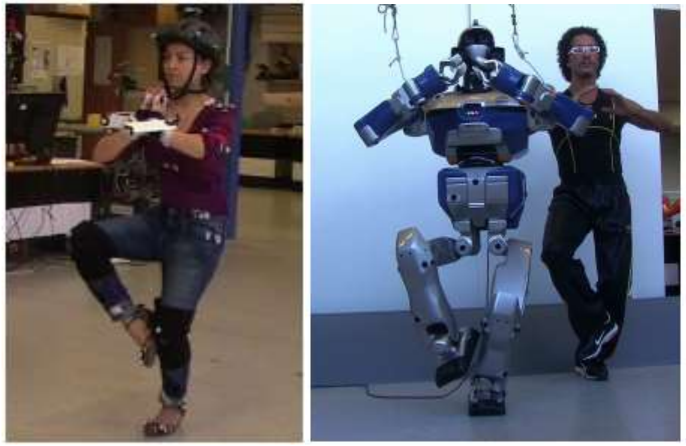
Human-Machine Interaction

excavator.

control unit.

Traditional for experts, skill training, programming

[1]



[2]

Intelligent for common user, no special skill, from physical states or emotions

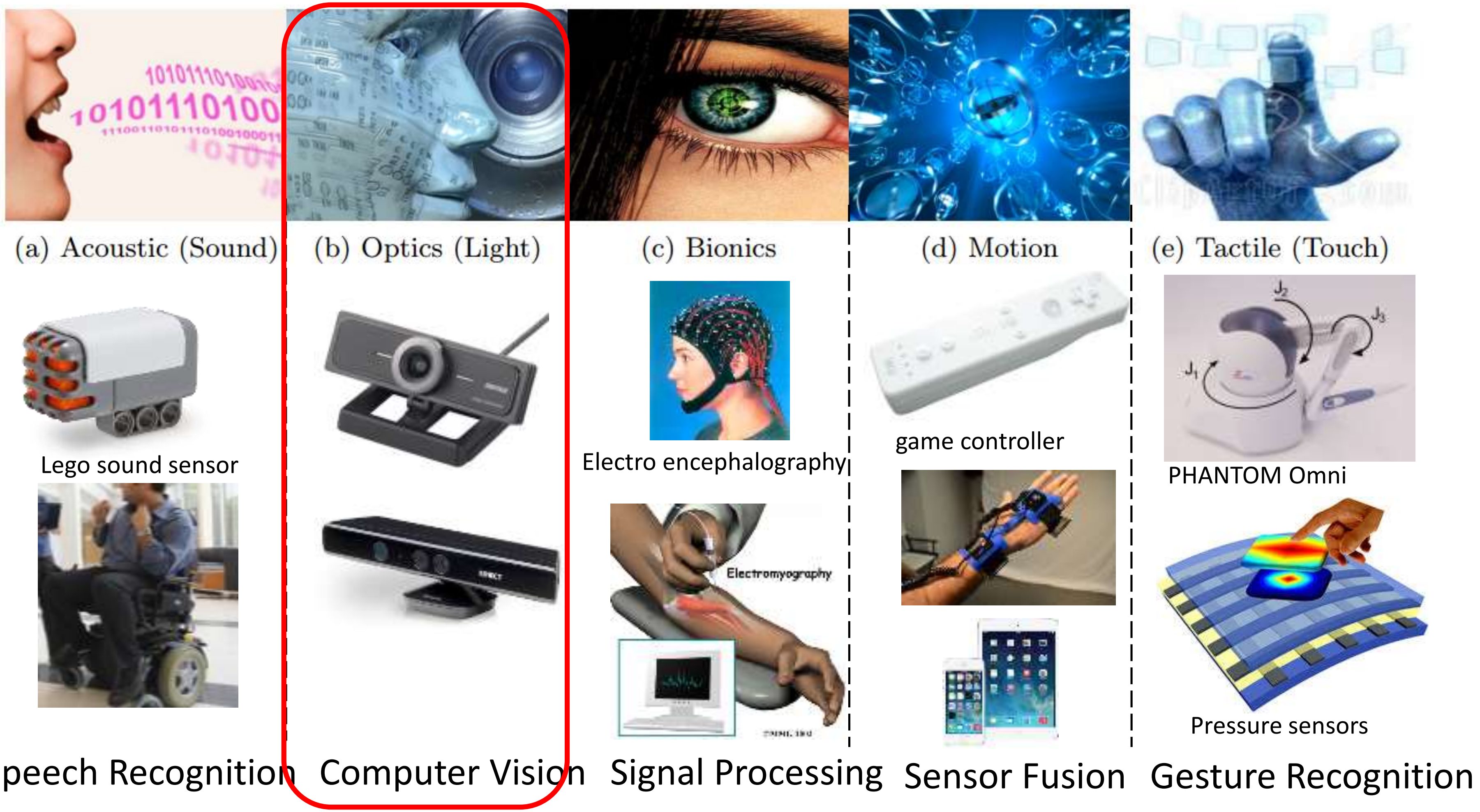
[1]. Stanton, et al. Teleoperation of a humanoid robot using full-body motion capture, example movements, and machine learning, Robotics and Automation, 2012
[2]. Ponce, et al. Dancing Humanoid Rob ots: Systematic use of OSID to Compute Dynamically Consistent Movements Following a Motion Capture Pattern, IEEE Robotics, 2015

HMI Types

Categories

Devices & Sensors

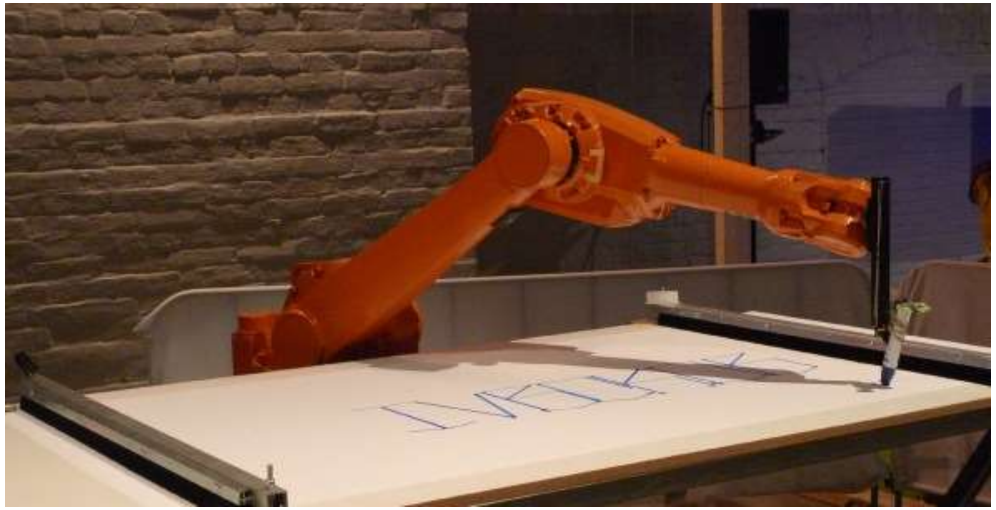
Computer Science



[3] Cannan, et al. Human-Machine Interaction (HMI): A Survey, University of Essex, 2010

Motivation

Human Movement



Motion Projection

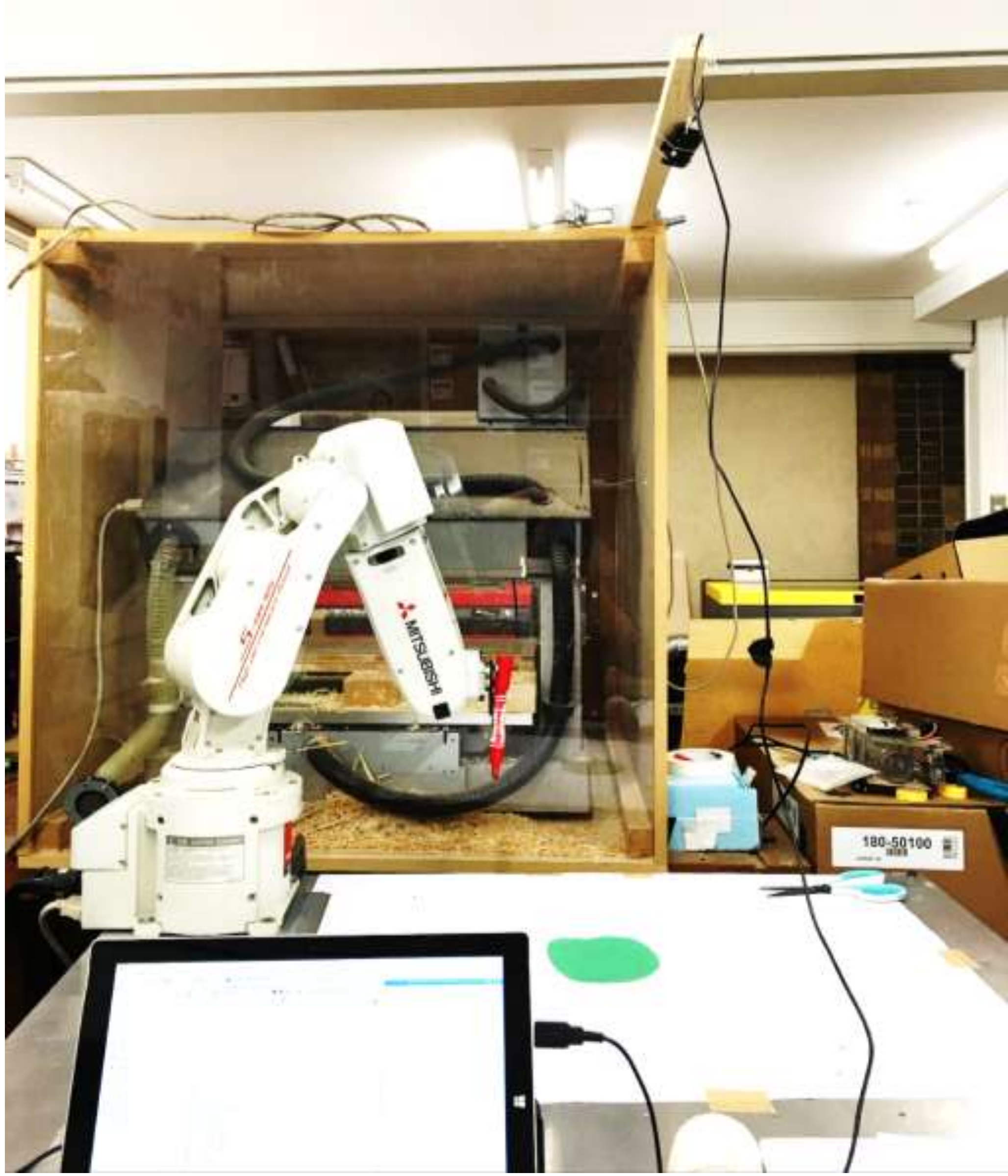
Vision-based Robot Drawing

Webcam	iBuffalo BSW20KM11
Robot	Mitsubishi Melfa RV2-SD
Program	MATLAB, Melfa Basic V
Toolbox	image acquisition toolbox
	image processing toolbox
	robotic toolbox
	Mitsubishi Melfa control toolbox
	Mitsubishi RT toolbox

BY Haoran Xie

Environments

- Webcam
  - Camera iBuffalo BSW20KM11
  - Identify object from 2D scenes
- Robot arm
  - RV2-SD
  - Joint coordinates (inverse kinematics)
  - Pick up object and put to desired location
  - Melfa Basic V programming
- Toolbox
  - (matlab) image acquisition toolbox
  - (matlab) image processing toolbox
  - (matlab) robotic toolbox – inverse kinematics
  - (matlab) Mitsubishi Melfa robot control toolbox – robot communication
  - (Mitsubishi) RT toolbox – offline evaluation



Algorithm

- 1. Capture video input
  - Motion capture of human movement in 2D
- 2. Object identification
  - Recognize the designated object by webcam
- 3. Trajectory generation
  - Generate the trajectory of movement
- 4. Code generation in Melfa Basic V
  - Make the program in Basic V programming
- 5. Robot control unit transform / communication
  - Communication via USB/TCP connection

Guiding System:

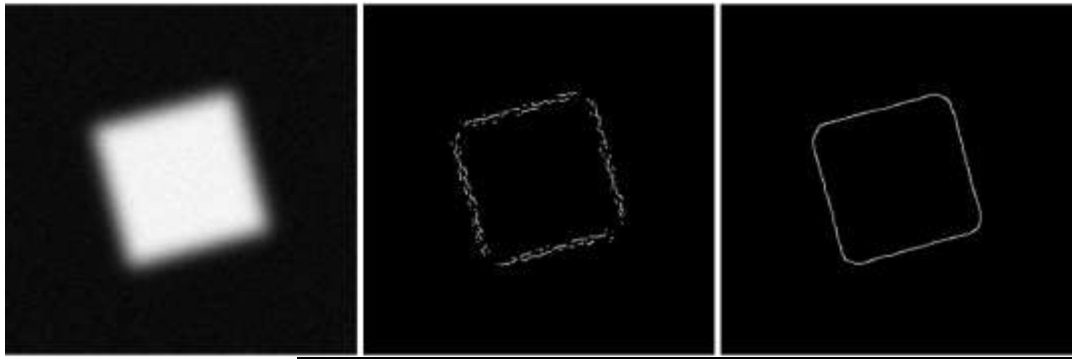
Motion Capture

Web camera

Color Recognition

- Image crop
  - Avoid trivial info
- Sharpen image
- Convert to grayscale
- Color subtraction
  - green

Shape Recognition

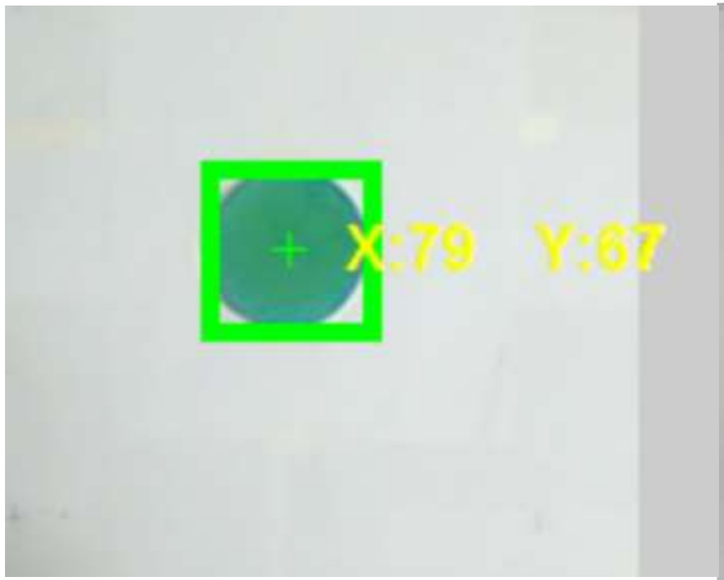


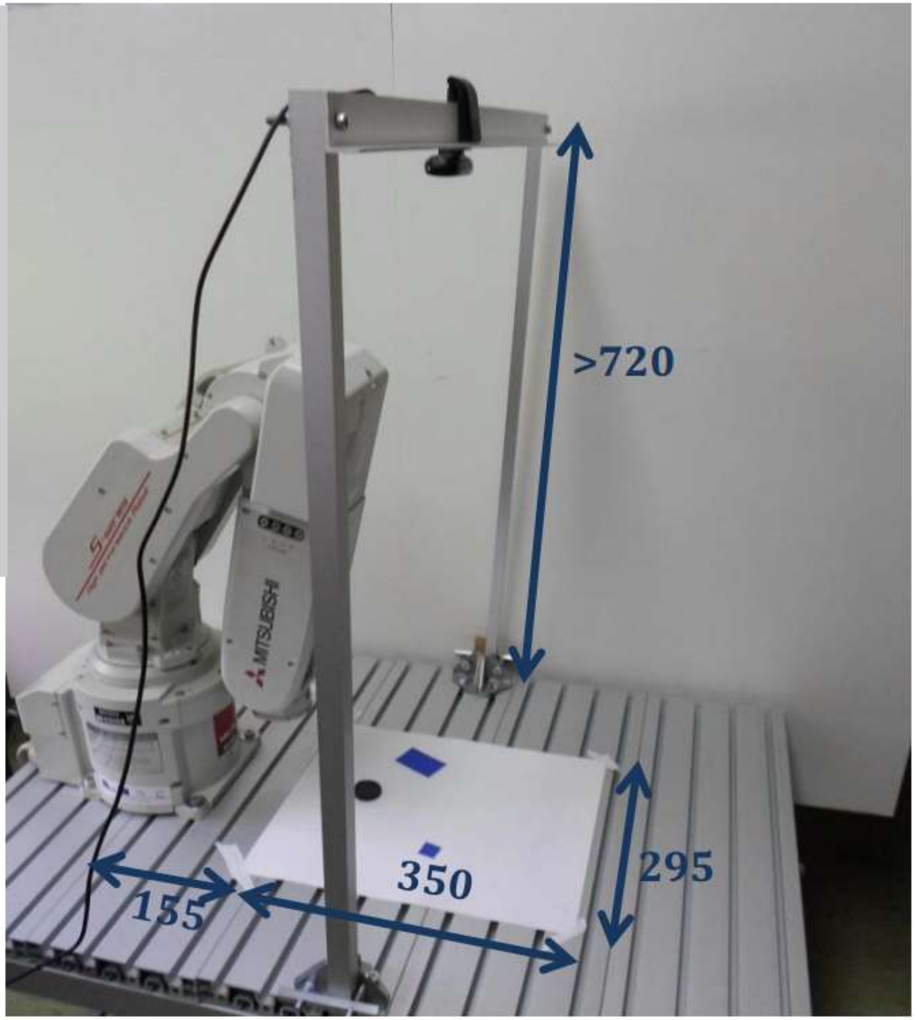
- Catch image boundary

- Get bounding box
- Calculate shape compactness
  - Compactness c = perimeter * perimeter / area
  - If shape is circle c = (2*pi*r)^2/(pi*r^2) = 4*pi
  - If shape is square c = (4*r)^2/r^2 = 16
  - If shape is rectangle c > 16

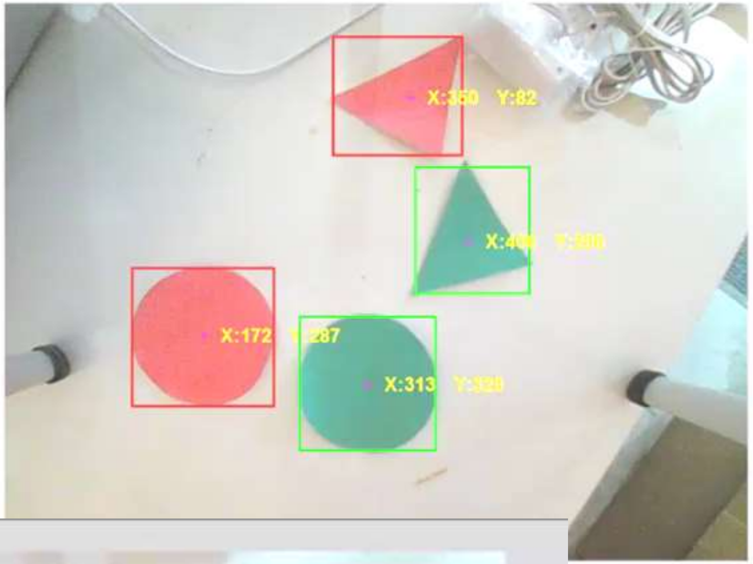
Trajectory generation

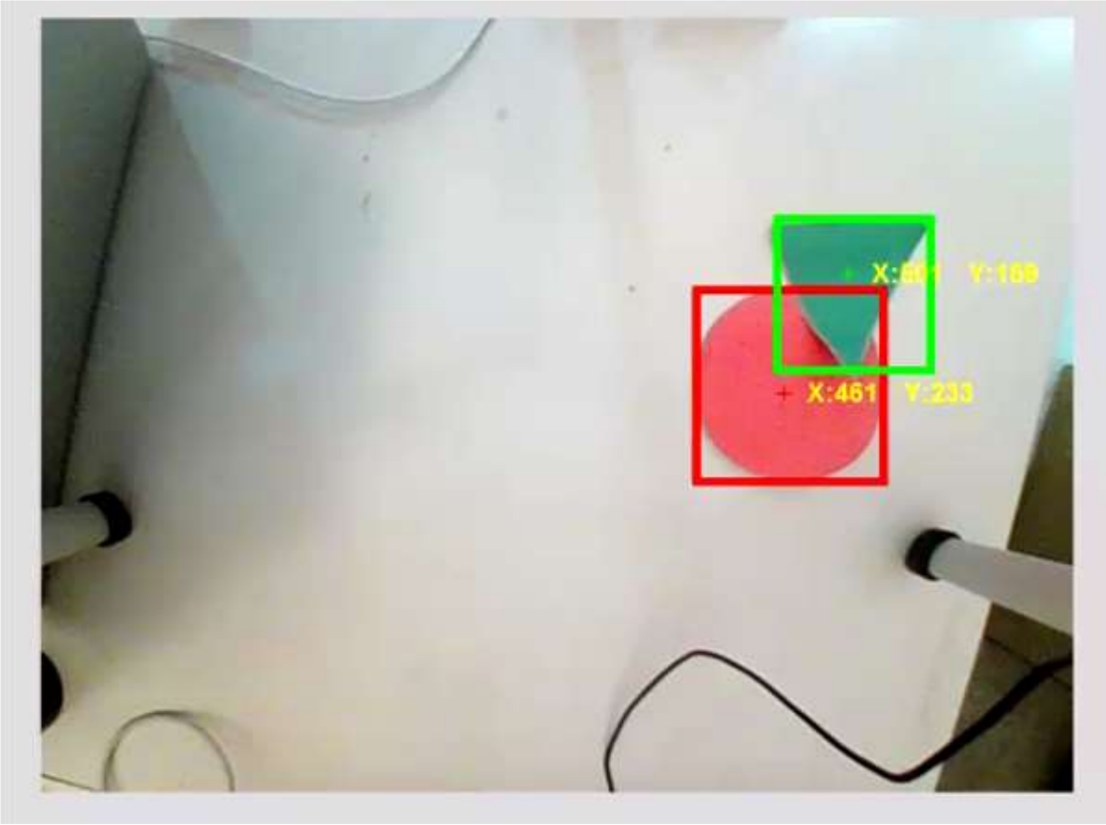
- Desktop
  - Size range
- Web camera
  - Location
- Robot arm
  - Location of robot base
- Object position
  - From pixel to location (millimeter)
  - Mapping

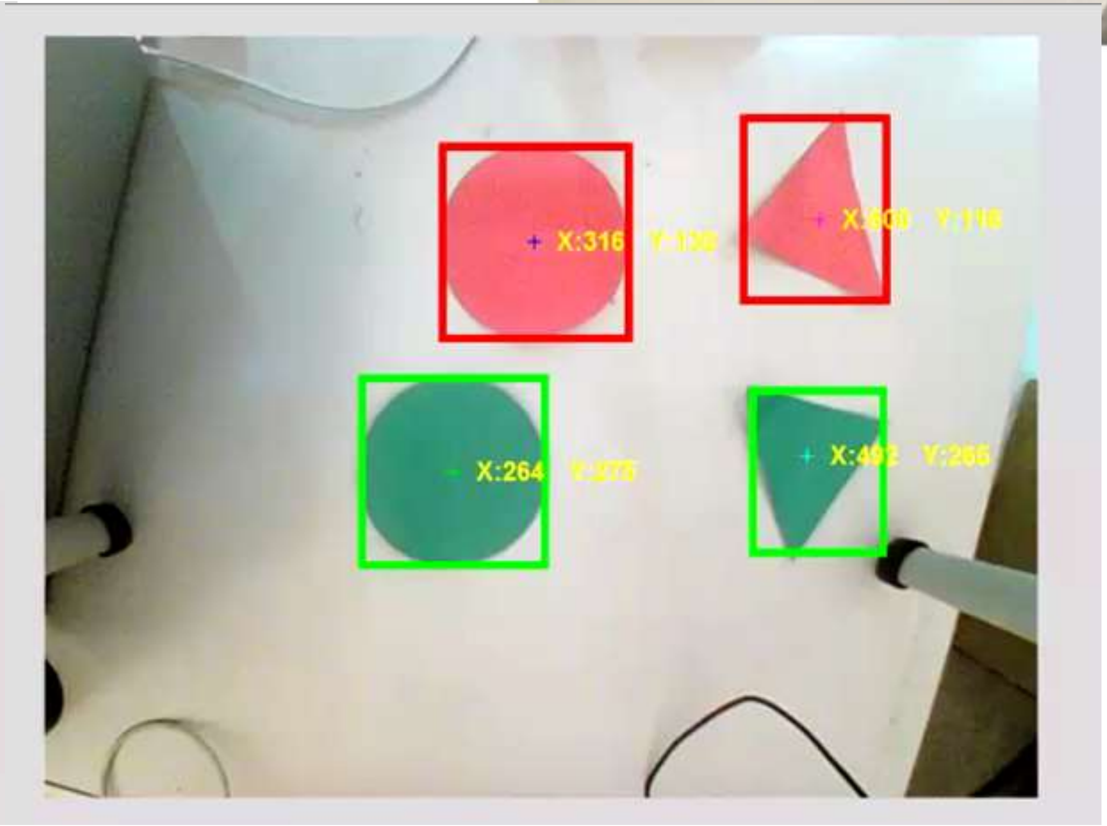




Object recognition







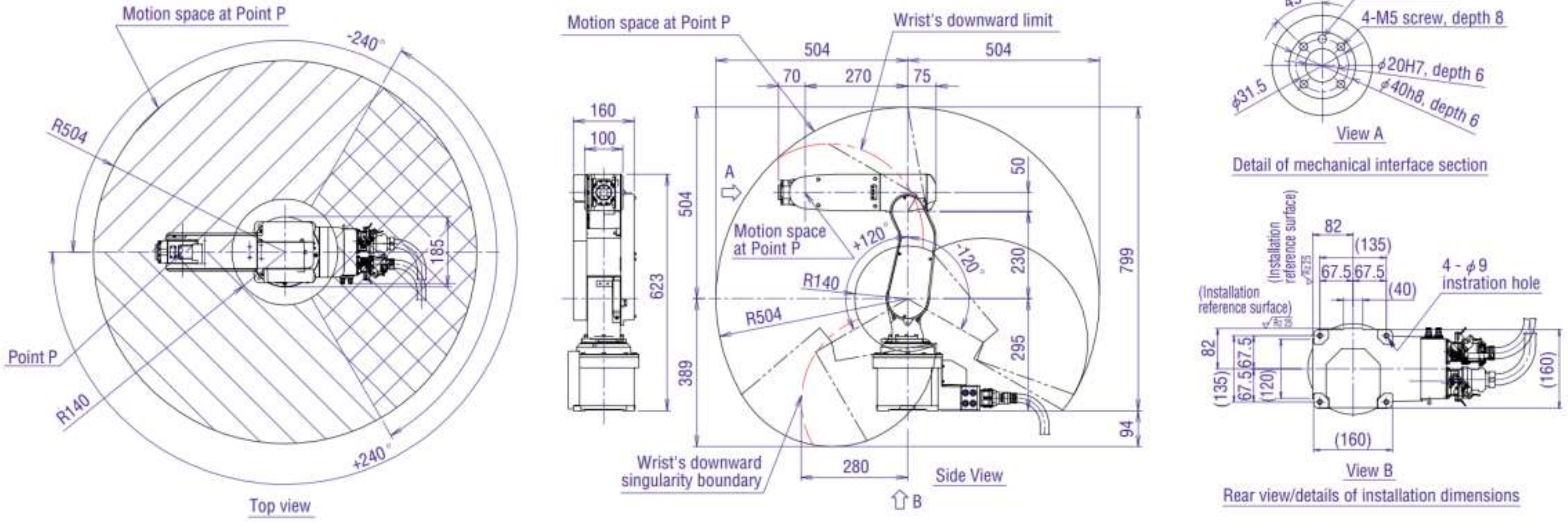
Projection System:
Robot Control

Melfa RV2-SD robot arm

Robot Configuration

Operating range	J1	deg	480 (-240 to +240)
	J2		240 (-120 to +120)
	J3		160 (0 to +160)
	J4		400 (-200 to +200)
	J5		240 (-120 to +120)
	J6		720 (-360 to +360)

[Robot's Outer Dimensions and Operating Range]



Melfa Basic V programming

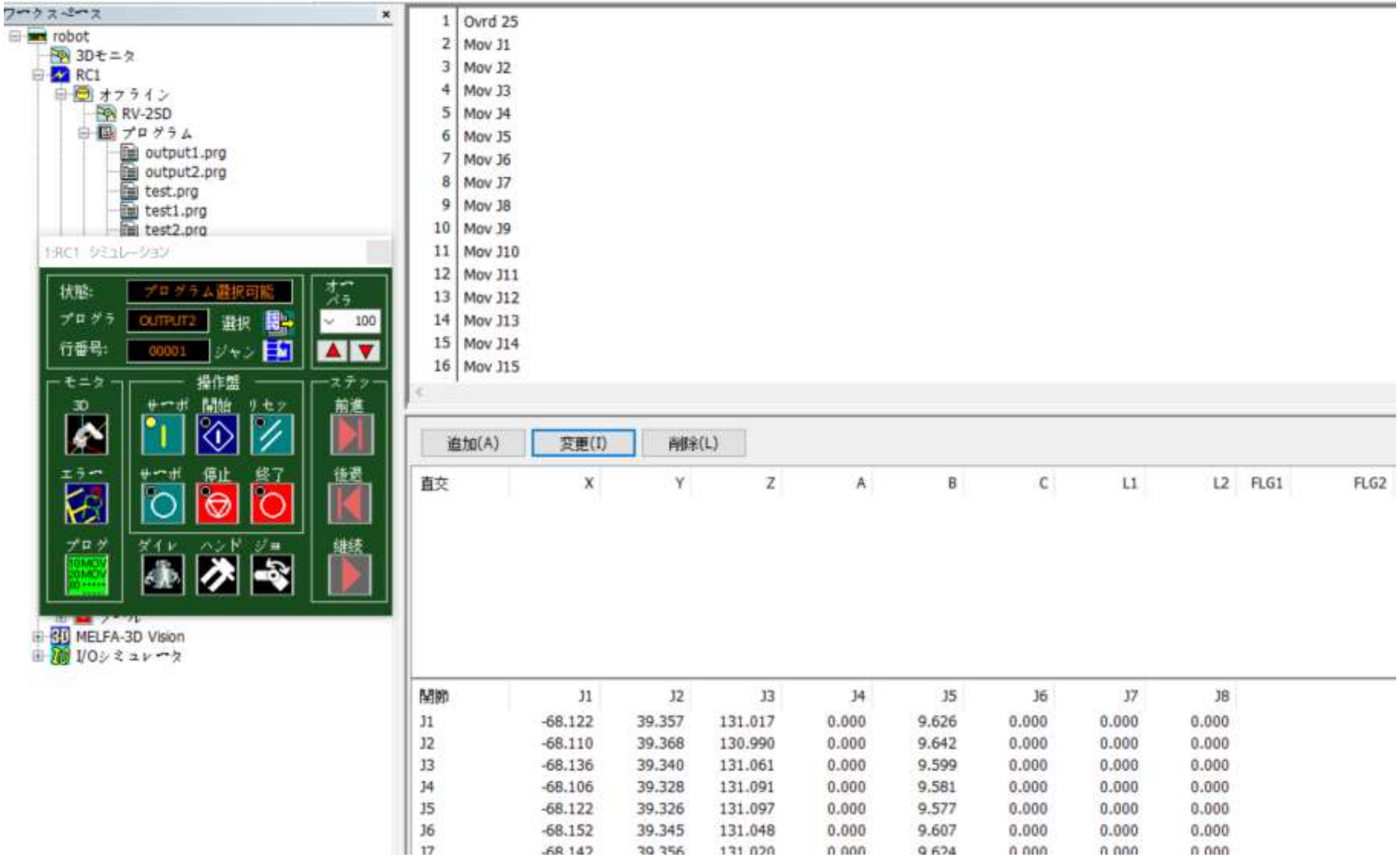
Command	Explanation	Example
MOV	Moves the robot to a teach-point position	MOV P1
MVS	Moves the robot to a teach-point in a straight line	MVS P1
	Moves the robot to a position above the teach-point in a straight line. (Z-Axis distance in Tool-frame)	MVS P1, -50
OVRD	Override speed limit (0 to 100%) (never use more than 30 for safety ! make this the first line of your program)	MVS 20
DLY	Delay in seconds – Robot waits	DLY 0.5
HOPEN	Opens the gripper	HOPEN 1
HCLOSE	Closes the gripper	HCLOSE 1
GOSUB	Calls a subroutine	GOSUB *PICK
RETURN	Returns from the subroutine	RETURN
DEF POS	Defines a position variable	DEF POS PTMP
END	End of program	END

Programming example

```
OVRD   20                         'set speed to 20%
MOV     P10, -50                  'go within 50mm of teach-point P10
OVRD   5                          'set speed to 5%
MVS     P10                       'go to P10
DLY     0.5                       'wait 0.5 seconds to make sure the robot stopped
HOPEN  1                          'open gripper
DLY       0.5                     'wait 0.5 seconds
OVRD   20                         'set speed to 20%
MVS       P10, -50                'move up, to leave position P10
END                               'end of program
```

Pr1: placing of an object

Programming Interface



Controllers

Emergency Stop

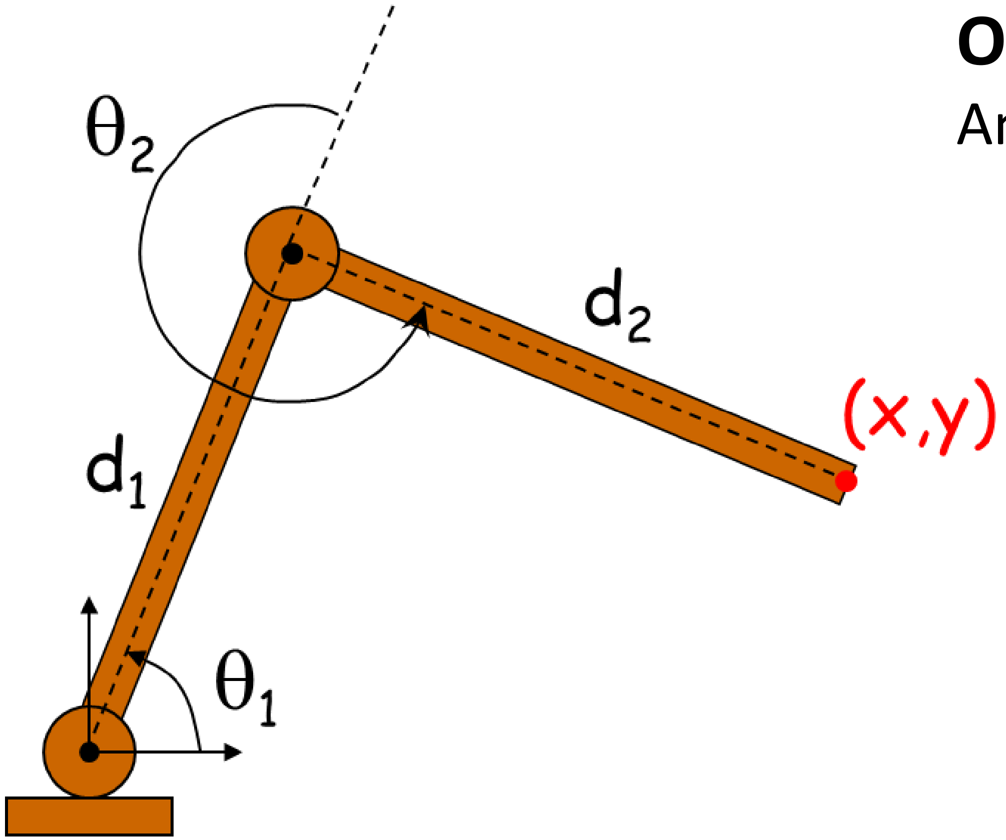
Teach Box

Emergency Stop

Program Error

Code	Meaning
L2800 – L2803	position data is inadequate. *
L2600 – L2603	position is out of range *
H0060	Emergency Stop on controller was pushed
H0070	Emergency Stop on T/B was pushed
H5000	The T/B Enable key was validated in the automatic mode.
H1010	Collision
C1350	Overload (possibly Collision)
C4340	Variable not defined (you forgot DEF POS or you forgot to download the teach-point file)

Inverse Kinematics



INPUT

The length of each link

The position of some point on the robot

OUTPUT

Angles of each joint needed to obtain that position

$$\theta_2 = \cos^{-1}\left(\frac{x^2 + y^2 - d_1^2 - d_2^2}{2d_1d_2}\right)$$

$$\theta_1 = \frac{-x(d_2\sin\theta_2) + y(d_1 + d_2\cos\theta_2)}{y(d_2\sin\theta_2) + x(d_1 + d_2\cos\theta_2)}$$

→Using robot toolbox

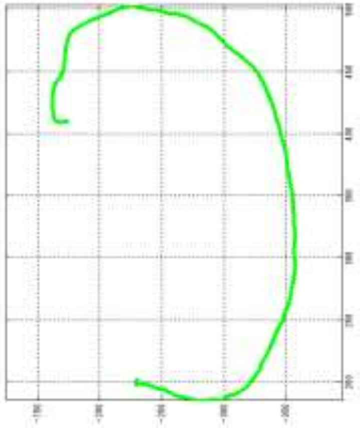
Examples

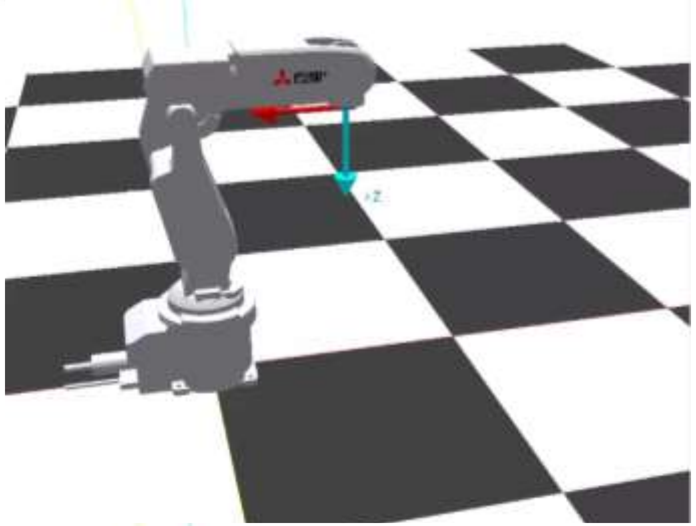
INPUT

Trajectory

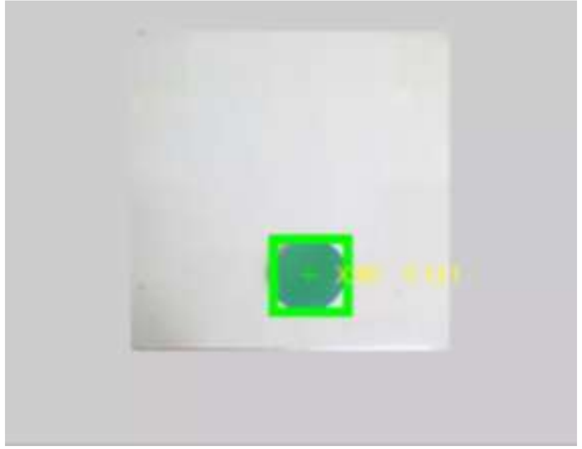
Inverse Kinematics

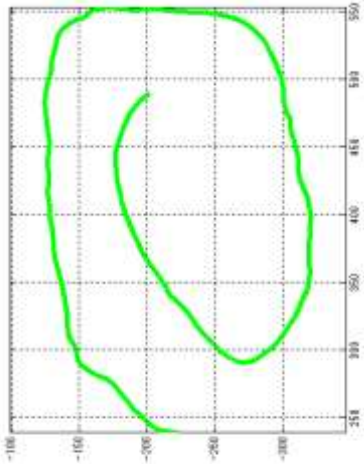
Case 1

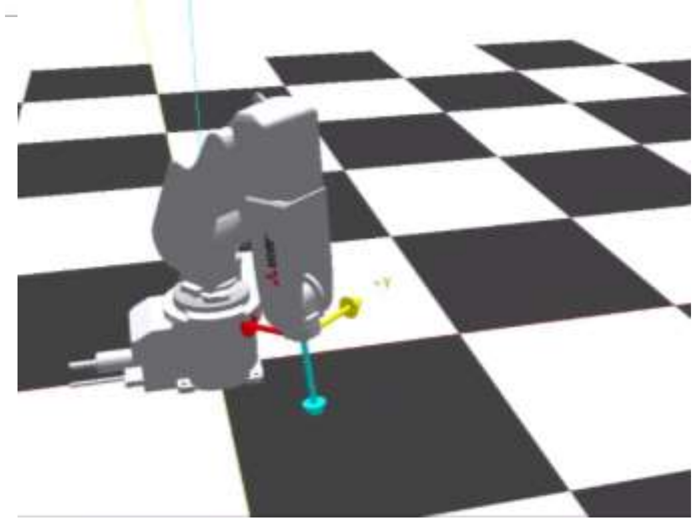




Case 2







Conclusion

- A guiding system using image processing
- A projection system using inverse kinematics