# Advanced Controller for Robot Manipulator

Jwu-Sheng Hu

Professor, Dept. of Electrical and Computer Engineering, NCTU, Taiwan Technical Director, MSL, ITRI, Taiwan

Director, Intelligent Robotics Division, MSL, ITRI, Taiwan

## Outline

- Advanced Robot Arm
- Real-time OS
- EtherCat Protocol
- Robotic Arm Dynamic Control Technology
- 3D Objects Pose Estimation
- Intelligent Graphic User Interface System
- Advanced Hand-Eye-Workspace Calibration

# Advanced Robot Arm

- Designed and assembled by Industrial Technology Research Institute.
- Small and compact size
- Seven degree of freedom





## Advanced Robot Arm

- EtherCAT communication
- No control box, and the motor driver is embedded in each link.
- Hollow shaft design for routing





## Outline

- Advanced Robot Arm
- <u>Real-time OS</u>
- EtherCat Protocol
- Robotic Arm Dynamic Control Technology
- 3D Objects Pose Estimation
- Intelligent Graphic User Interface System
- Advanced Hand-Eye-Workspace Calibration

### Real-time OS Introduction

- Growing penetration of Linux
- OSS EtherCAT has been developed
- Control applications in <u>user space</u> is possible
   IgH version 1.5.0 or above
- However,
  - IgH can't guarantee RT capability
  - Even RTAI is adopted

### Real-time OS IgH EtherCAT

- IgH EtherCAT is composed of
  - master module
    - Provide functions and data structures to allow applications accessing the master functionalities via API
  - control application
    - Interface with the master modules by means of API
    - For the cyclic exchange of process data with EC slaves
    - Can be implemented with IgH EC libraries

#### Real-time OS Cereia's Work

- "A user space EtherCAT master architecture for hard real-time control systems "
  - 2012 IEEE ETFA Conference
- Implementation
  - Linux kernel 2.6.32.11 + RTAI 3.8.1
  - IgH 1.5.0
- Improve the user space control application
  - Modify the control application to be RTAI tasks
  - Make it to reach real-time capability

#### Real-time OS Drive via Character Device



#### Real-time OS Jitter: Kernel vs. User Space



#### Kernel space timing



#### User space timing

Kernel	User
<pre>while(1) {    wait(K-sem);    write_outputs();    read_inputs();</pre>	<pre>while(1) {     rt_task_wait_period();     signal(K-sem);     wait(U-sem);</pre>
<pre>signal(U-sem); }</pre>	<pre>/*data processing*/ }</pre>

#### Real-time OS Character Device vs. Shared Memory



#### Real-time OS Jitter: User Space/Cereia's Design



#### Real-time OS EC-based Robot OS



### Real-time OS Conclusion

- Control application can be implemented at user space based on RTAI
  - Real time guaranteed
- Cereia's Design can be used to realize a Robot OS
  - For real-time control

# Outline

- Advanced Robot Arm
- Real-time OS
- EtherCat Protocol
- Robotic Arm Dynamic Control Technology
- 3D Objects Pose Estimation
- Intelligent Graphic User Interface System
- Advanced Hand-Eye-Workspace Calibration

## EtherCat Protocol Outline

- Industrial Communication Technique Comparison
- COE Communication Structure
  - Cyclic Data Transmission
  - Acyclic Data Transmission
- EtherCAT Master Processing Procedure
- Device Description File
  - Device Description File Analysis
  - Package Analysis
  - Establish Motion Control Parameters
- EtherCAT Master User Interface Demo

#### EtherCat Protocol Industrial Communication Technique Comparison

Real-time Level						
Protocols	Real-time	Throughput	Max	Synchronisation		
	class		devices			
CAN 2		10 kb/s-1 Mb/s,	32	Y		
		31.25 kb/s, 1 Mb/s,				
		2.5 Mb/s (5 Mb/s				
		optical fiber)				
PROFIBUS 2				N		
PROFIBUS-DP		9.6 kb/s-12 Mb/s	126			
PROFIBUS-PA		31.25 kb/s	32/seg.			
MODBUS 1		10,100 Mb/s,	247	N		
		1 Gb/s				
Ethernet / IP 1		10,100 Mb/s,	Almost	Y		
		1 Gb/s	Unlimited			
PROFINET 三種皆有		100 Mb/s	Almost	Y		
			Unlimited			
PowerLink	3	100 Mb/s	240	Y		
SERCOS	3	100 Mb/s	254	Y		
EtherCAT	3	100 Mb/s	65535	Y		

#### EtherCat Protocol COE Communication Structure

COE(CANopen Over EtherCAT) includes the following communication mode:

(1) Mailbox mode – Use SDO (Service Data Object) for acyclic transmission.

(2) Buffer mode – Use PDO (Process Data Object) for cyclic transmission.



#### EtherCat Protocol COE Communication Structure

• Acyclic Data Transmission(Mailbox mode) When the sender writes the buffer, the buffer is locked for writing until the receiver has read it out.



#### EtherCat Protocol COE Communication Structure

• Cyclic Data Transmission(Buffer mode)

The sender can always update the content of the buffer. If the buffer is written faster than it is read out by the receiver, old data is dropped. Thus, the receiver always gets the latest consistent buffer content which was written by the sender.



#### EtherCat Protocol EtherCAT Master Processing Procedure



• EtherCAT slave device describe



• Device description file analysis





Beckhoff 01:00:00

Beckhoff 10:1c:9c

MS-NLB-PhysServer-01\_05:10:1c:9c

Beckhoff 01:00:00

ECAT

ECAT

21326 21.200085000

21327 21.202077000

- Establish motion control parameters
  - When controller parameter is used, it musts set up.
  - Using handshake mechanism makes sure whether

	= DefaultSize	= StartAddress	= ControlByte	= Enable	= MinSize	= MaxSize	Bbc Text
1	128	#x1000	#x26	1	34	192	MBoxOut
2	128	#x1400	#x22	1	34	192	MBoxIn
3	2	#x1800	#x64	1			Outputs
4	6	#x1C00	#x20	1			Inputs

```
EtherCAT datagram: Cmd: 'FPRD' (4), Len: 128, Adp 0x3e9, Ado 0x1400, Cnt 1

    Header

              : 4 (Configured address Physical Read)
    Cmd
    Index: Oxh7
                        Slave addr : Slave node,
    Slave Addr: 0x03e9
    Offset Addr: 0x1400
   Interrupt: 0x0000
 EtherCAT Mailbox Protocol:CoE

    Header

     Length: 10
     Address: 0x0000
     Priority: 0
           : CoE (CANopen over EtherCAT) (0x3)
     Type
     Counter : 5
    COE
     Number : 0
     Type
           : SDO Res(3)
                                     CANopen over EtherCAT
    ■ SDO Res : Scs 2

    ⊞ Initiate Upload Response: 0x4f

     Index: 0x6000
     SubIndex: 0x00
     Data: 0x08
   Data: 454c393830305f344120285049433234295f56356930315f...
  Working Cnt: 1
```

#### EtherCat Protocol EtherCAT Master User Interface Demo

- EtherCAT master interface program develops in Window XP and uses Visual Studio 2010 C++ to develop.
- The master UI includes the following feature :
  - 1) Reading ENI file acquires network topology and slave station information
  - 2) Sending the initializing packet of ENI file initializes the slave station in different state machine transformation.
  - 3) Acyclic Data transmission. Ex : Set up parameter Write into fixed command...
  - 4) Write into the acyclic data for two different slaves at the same time

#### EtherCat Protocol EtherCAT Master User Interface Demo

EtherCAT Master Demo	
Matter Device MAC Address: 00 0c 29 97 6e 3a Load Configuration O Udp Device NDIS Device Read ENI file Pre-Op Operational	State Machine
ID Communication         Stat         Current State:         Operational           Cycle Time:         1         Stop         Requested State:         Operational	Logical Mapping
Maibox       slave1:       Member GREEGER WWW       slave2:       Tem 2 (EL4132)       White         Index:       QueG041       Status woord       Uk4061       White         Subindex:       0       Subindexi       2       Oc6         Number of Byte1       Number of Byte2       Oc6       So6         Data1 to Write:       Data2 to Write:       Oata2 to Write:       Servo on         Data1 to Rest_5002       0x2250       Data2 to Rest_(n)       Servo on         Result:       No Enoc (Invokeld = 0x22)       Result2:       No Enoc (Invokeld = 0x1a)       Servo off         Flee Access over EtherCAT       Upload.       Upload.       Servo and off       Servo and off	O Image         Irput         0000 00 00 00 00 00 00 00 00 00 00         0010 00 01 00 00 00 00 00 00 00 00         0018 04 00 00 00 00 00 00 00 00 00         0020 04 01 01 00 00 00 00 00 00 00         0022 04 01 01 00 00 00 00 00 00 00         0028 04 00 00 00 00 00 00 00 00         0028 04 00 00 00 00 00 00 00 00         0008 00 00 00 00 00 00 00 00         0008 00 00 00 00 00 00 00 00         0010 00 00 00 00 00 00 00 00         0010 00 01 00 00 00 00 00 00 00         0011 00 01 00 00 00 00 00 00 00         0022 04 01 01 00 00 00 00 00 00 00         0018 00 00 00 00 00 00 00 00 00 00         0018 00 00 00 00 00 00 00 00 00 00         0018 00 00 00 00 00 00 00 00 00 00         0018 00 00 00 00 00 00 00 00 00 00         0018 00 00 00 00 00 00 00 00 00 00         0018 00 00 00 00 00 00 00 00 00 00         0018 00 00 00 00 00 00 00 00 00 00         0018 00 00 00 00 00 00 00 00 00 00         0023 00 00 00 00 00 00 00 00 00 00 00

## Outline

- Advanced Robot Arm
- Real-time OS
- EtherCat Protocol
- Robotic Arm Dynamic Control Technology
- 3D Objects Pose Estimation
- Intelligent Graphic User Interface System
- Advanced Hand-Eye-Workspace Calibration

#### Robotic Arm Dynamic Control Technology Introduction

• We develop high-precision, high-response-speed multi-axis control laws for 7-dof robotic arm. The control laws are robust, not affected by the change of the load arm. We regulate the control parameters with systematic rule.

#### Robotic Arm Dynamic Control Technology Kinematics

• Forward kinematics:

We use D-H model to get the end-effector pose corresponding to angle of each axis. We transfer joint-space coordinate into Cartesian coordinate.



## Robotic Arm Dynamic Control Technology Kinematics



#### Robotic Arm Dynamic Control Technology Kinematics

• Inverse kinematics:

We calculate angle of each axis by the position of end-effector. We transfer Cartesian coordinate into joint-space coordinate. The verification through 3D simulation are shown as follow:





### Robotic Arm Dynamic Control Technology Dynamic Equations

Robotic arm robot arm dynamic equations describe the relationship between position, velocity, acceleration and torque, which can be expressed as the following formula:  $\tau = M(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + G(\theta)$ 

where  $\theta$  is the rotation angle of each axis, M( $\theta$ ) is the mass inertia matrix, C( $\theta$ , $\dot{\theta}$ ) is the centripetal force and the Coriolis force matrix, the above two matrices are 7 x 7 matrix, G( $\theta$ ) is the 7×1 gravity vector,  $\tau$  is the torque of each axis from the actuator.

#### Robotic Arm Dynamic Control Technology Dynamic Equations

• Newton-Euler equation and the Euler-Lagrange equation are common methods to derive dynamic equations. We represent length, mass, inertia and centroid location of robotic arm as symbols through MATLAB symbolic toolbox, and then derive the dynamic equations by the Newton-Euler equation and Euler-Lagrange equation, respectively, to get  $M(\theta)$ ,  $C(\theta, \dot{\theta}), G(\theta)$ . After verification, the  $\tau$  derived by the Newton-Euler equation and by the Euler-Lagrange equation are matched.

### Robotic Arm Dynamic Control Technology Current Loop System Identification

• Accelnet Plus driver provides position, velocity and current mode. Due to that current is proportional to torque, the dynamic control is implemented using the current mode. In order to determine whether we need to compensate or not, we have to get the frequency response of the current loop of the servo motor.



The driver control loop block diagram
## Robotic Arm Dynamic Control Technology Current Loop System Identification

- The process of the system identification is as follows:
  - Set the parameters of current loop controller (PI controller's Kp, Ki)
  - 2) Input different frequency of sinusoidal current command to motor by CME2, respectively, and record the magnitude and phase of input and output current to plot bode plot.
  - Combine the input current command and output current in step b, and identify the system by using MATLAB identification toolbox.
  - 4) Input the test sinusoidal command to the model from step 3, and then compare the model output with the actual model output to verify the result of identification.

# Robotic Arm Dynamic Control Technology Experimental Result

- DC brushless motor: KBM-10H01-C00
- Digital servo drive: Accelnet Plus Module EtherCAT AEM-090-14
- Absolute encoder: DS-58[20] (18 bits Angular resolution)

• Obtained model :  $\frac{37760 s + 18490}{s^2 + 38470 s + 20080}$ 

# Robotic Arm Dynamic Control Technology Experimental Result

• Comparing the bode plot of models and servo motor as follows:



# Robotic Arm Dynamic Control Technology Experimental Result

• Input the sinusoidal test command and compare the outputs of the model and the motor.



#### Robotic Arm Dynamic Control Technology Future Work

- Solve the singularity problem of 7-axis Jacobian Matrix.
- Build the process of robot arm system identification to identify the parameters of dynamic model.
- Develop high efficiency motion control law of robot arm.

# Outline

- Advanced Robot Arm
- Real-time OS
- EtherCat Protocol
- Robotic Arm Dynamic Control Technology
- **3D Objects Pose Estimation**
- Intelligent Graphic User Interface System
- Advanced Hand-Eye-Workspace Calibration

## 3D Objects Pose Estimation Outline

- Introduction
- Problem Statement
- CAD Database system
- Hand-Eye Calibration
- Experimental Result
- Conclusion and Feature Work

#### 3D Objects Pose Estimation Introduction

- With the advent of new-generation depth sensors, using 3D data is becoming increasingly popular. As these sensors are commodity hardware and sold at low cost, a rapidly growing group of people can acquire 3-D data cheaply and in real time.
- For a texture-less object like a mechanical part, conventional visual feature matching usually fails due to the absence of rich texture features.
- The vision system is easily set up to recognize different objects by using CAD models database.

#### 3D Objects Pose Estimation Problem Statement

- How to carefully chosen feature descriptor encode more information compactly and thereby provide higher accuracy 6-DOF pose estimation and enable faster computation.
- How to use CAD data to establish the CADmodel database, so that the system not only work even in highly cluttered environment, but also to accurately estimate 6-DOF pose of the workpieces.

# System Architecture



# 3D Objects Pose Estimation6-DOF Pose Estimation System

• The system can convert scene depth image to point cloud format, and estimate the 6-DOF pose of object by using CAD Database which has been established in offline phase.



# 3D Objects Pose Estimation6-DOF Pose Estimation System

• The Voxel Grid Filter created 3D voxel grids from the input point cloud data, then points in each voxel (i.e. 3D box) will be describe as their centroid.



The centroid point of all the points in the voxel.

3D Objects Pose Estimation Voting-Scheme Matching



(1) Reference point Sr is paired with every other point Si, and their point pair feature F is calculated.

(2) Feature F is matched to the hash table, which returns a set of point pairs on the model that have similar distance and orientation.

(3) For each point pair on the model matched to the point pair in the scene, the local coordinate is calculated by solving  $\mathbf{s}_i = T_{s \to g}^{-1} R_{\mathbf{x}}(\alpha) T_{s \to g} \mathbf{m}_i$ .

(4) After is calculated, a vote is cast for the local coordinate  $(\mathbf{m}_i, \alpha)$ 

## 3D Objects Pose Estimation Pose Clustering

• The retrieved poses are clustered such that all poses in one cluster do not differ in translation and rotation for more than a predefined threshold.

 $if (trans_{pose_n} < threshold_{trans_n}) and (rotation_{pose_n} < threshold_{rotation_n})$ 

=> pose\_n is put in cluster\_n

• The score of a cluster is the sum of the scores of the contained poses, after finding the cluster which vote is bigger than threshold, the resulting pose is calculated by averaging the poses contained in the cluster.

 $if vote_{cluster_n} > threshold$ 

=> cluster\_n is final pose estimation

## 3D Objects Pose Estimation Eye-in-Hand Calibration

- We set up an eye-in-hand configuration using the Kuka 6-DOF robot arm installed with a Kinect camera.
- We first calibrated the eyein-hand configuration by using the DLR Camera Calibration Toolbox.



#### 3D Objects Pose Estimation Calibration Result

• We took 15 images of the calibration pattern with varying orientations and translation. The error means are 2.54 mm, 7.99mm and 2.74mm in x-coordinate, y-coordinate and z-coordinate, respectively.



#### 3D Objects Pose Estimation Experimental Result



3D Objects Pose Estimation Conclusion and Feature Work

- The 6-DOF object pose estimation based on CAD-model database has been proposed.
- We have implemented the 6-DOF pose estimation system in kuka robot. The experimental result shown that the proposed method can pick 3-4 bins in one minute.
- In the feature, we will speed up the cycle time of the method.

# Outline

- Advanced Robot Arm
- Real-time OS
- EtherCat Protocol
- Robotic Arm Dynamic Control Technology
- 3D Objects Pose Estimation
- Intelligent Graphic User Interface System
- Advanced Hand-Eye-Workspace Calibration

# Intelligent Graphic User Interface System Goal

- The graphic user interface system is developed for operating an *industrial robotic arm*, with the implementation on the tablet of an Android based application.
- The graphic user interface system:
  - Providing a more ease-to-use virtual interface for the operator to execute the task.
  - Providing a platform to realize the remote control.

# Intelligent Graphic User Interface System Communication

- The WiFi technology makes it possible for us to control robot *remotely*, while tablet makes it possible for us to control the robot in the visual interface.
- The programming based on TCP Windows Sockets is used to realize the communication.
  - TCP Windows Sockets:
    - A reliable transport protocol
    - Ordered
    - Streaming

# Intelligent Graphic User Interface System Communication

- It's a simple way to realize the *Client / Server* model.
  - Server:

• Client:



# Intelligent Graphic User Interface System Communication

- PC : The program uses multi-threads.
  - Multi-thread is the ability of a program to manage multiple requests at one time in the computer.



# Intelligent Graphic User Interface System Specifications

- Tablet
  - ASUS Memo Pad FHD ME302KL-1B024A
  - Qualcomm 8064 Quad-core 1.5GHz
  - -OS: Android4.2
  - Size : 264 x 182 x 9.5mm
  - LCD size : 10.1-inch IPS
  - WiFi 、 Bluetooth

Mode-selected button • Tablet 👩 operate Connection button XYZ Mode Teaching Joint Mode - Joint Mode sw\_op Joint buttons J1-J1+ Joint J2information J2+ J3-J3+ J4-J4+ J5-Save points button J6gripper ave Point Gripper button up Viewing angle buttons OPEN Save right left down IJ

- Tablet
  - xyz Mode



- Tablet
  - Teach Mode



• Tablet 🧑 operate XYZ Mode Teaching Joint Mode - Knob Mode OFF sw\_op JI J4 Seekbar to adjust the J2 J5 scale of the knob Joint buttons J6 J3 360 Knob Ū 씁

## Intelligent Graphic User Interface System Demo



# Outline

- Advanced Robot Arm
- Real-time OS
- EtherCat Protocol
- Robotic Arm Dynamic Control Technology
- 3D Objects Pose Estimation
- Intelligent Graphic User Interface System
- Advanced Hand-Eye-Workspace Calibration

#### Advanced Hand-Eye-Workspace Calibration Introduction

• The proposed Hand-Eye-Workspace calibration system including a laser pointer installed on the robot arm, a stationary camera, and a planar surface. The laser pointer beams to the surface, and the camera observes the projected laser spot. It is not necessary for robot arm to enter the view scope of the camera for calibration. The configuration of camera is more flexible.

#### Advanced Hand-Eye-Workspace Calibration Definition

- ${}^{C}\mathbf{H}_{B}$ : The transformation matrix between the fixed camera and the robot base. It represents the camera extrinsic matrix.
- ${}^{B}\mathbf{H}_{E}$ : The transformation matrix between the robot base and the end-effector. It represents the robot kinematics.
- The relationship between the laser pointer and the end-effector.
- Plane equation
- Camera intrinsic parameters



#### Advanced Hand-Eye-Workspace Calibration Definition

The laser line is defined in the end-effector coordinate with two position parameters (y, z) and two Euler angles  $(\theta, \phi)$ . The values of four parameters are unique. The direction of line is computed with  $(\theta, \phi)$  as  ${}^{E}_{L} \mathbf{u} \left( {}^{E}_{L} \theta, {}^{E}_{L} \phi \right)$ 



$$= \left[ \cos {}^{E}_{L} \theta \sin {}^{E}_{L} \phi \quad \sin {}^{E}_{L} \theta \sin {}^{E}_{L} \phi \quad \cos {}^{E}_{L} \phi \right]^{T}$$

### Advanced Hand-Eye-Workspace Calibration Line-Plane Intersection

• Laser line

The laser line in the camera coordinate can be obtained using  ${}^{C}\mathbf{H}_{B}$ ,  ${}^{B}\mathbf{H}_{E}$  and  $\left({}^{E}_{L}y, {}^{E}_{L}z, {}^{E}_{L}\theta, {}^{E}_{L}\phi\right)$ . The direction is denoted as  ${}^{C}\mathbf{z}_{laser}$ . The position is denoted as  ${}^{C}\mathbf{p}_{laser}$ .

• Plane definition

A plane is generally described by the normal vector  $\mathbf{n}$  and the position vector  $\mathbf{p}$  of any point on the plane, but it can be simply defined by the unique position vector whose direction is the normal vector as

$$\mathbf{a} = \begin{cases} \mathbf{n}, & \text{if } \mathbf{n} \cdot \mathbf{p} = 0\\ (\mathbf{n} \cdot \mathbf{p})\mathbf{n} & \text{otherwise} \end{cases}$$

#### Advanced Hand-Eye-Workspace Calibration Line-Plane Intersection

• Line-Plane intersection

The laser spot on the plane can be defined with the distance  $d_{laser}$  as  ${}^{C}\mathbf{p}_{spot} = d_{laser} {}^{C}\mathbf{z}_{laser} + {}^{C}\mathbf{p}_{laser}$ . It matches the geometrical constraint as

$$\left(d_{laser}^{C}\mathbf{z}_{laser}+{}^{C}\mathbf{p}_{laser}-\mathbf{a}\right)\cdot\mathbf{a}=0.$$

The distance  $d_{laser}$  can be obtained as

$$d_{laser} = \frac{\|\mathbf{a}\| - {}^{C}\mathbf{p}_{laser} \cdot \mathbf{a}}{{}^{C}\mathbf{z}_{laser} \cdot \mathbf{a}}.$$

#### Advanced Hand-Eye-Workspace Calibration Camera Model

The camera model is a pin-hole type including lens distortion. The 2D position (*u*,*v*) in an image of a 3D point <sup>C</sup>**p** =[<sup>C</sup>x, <sup>C</sup>y, <sup>C</sup>z]<sup>T</sup> can be obtained with the intrinsic matrix **K** and the lens distortion parameters (κ<sub>1</sub>, κ<sub>2</sub>, ρ<sub>1</sub>, ρ<sub>2</sub>) as

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K} \cdot \begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} \text{ with } \mathbf{K} = \begin{bmatrix} f_u & \alpha_c \cdot f_u & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$
### Advanced Hand-Eye-Workspace Calibration Camera Model

• The camera model is a pin-hole type including lens distortion. The 2D position (u, v) in an image of a 3D point  ${}^{C}\mathbf{p} = [{}^{C}x, {}^{C}y, {}^{C}z]^{T}$  can be obtained with the intrinsic matrix **K** and the lens distortion parameters  $(\kappa_1, \kappa_2, \rho_1, \rho_2)$  as  $\begin{bmatrix} x_d \\ y_d \end{bmatrix} = (1 + \kappa_1 \|\mathbf{x}_r\|^2 + \kappa_2 \|\mathbf{x}_r\|^4) \mathbf{x}_r + \begin{bmatrix} 2\rho_1 x_r y_r + \rho_2 (\|\mathbf{x}_r\|^2 + 2x_r^2) \\ \rho_1 (\|\mathbf{x}_r\|^2 + 2y_r^2) + 2\rho_2 x_r y_r \end{bmatrix}$ where  $\mathbf{x}_r = \begin{bmatrix} x_r & y_r \end{bmatrix}^T = \begin{bmatrix} C & x/C & C \\ C & Z & C \\ Y/C & Z \end{bmatrix}^T$  is the ray

direction from the camera.

# Advanced Hand-Eye-Workspace Calibration All Parameters

- ${}^{C}\mathbf{H}_{B}$ : The transformation matrix between the fixed camera and the robot base. It is defined with the Euler angles  $\begin{pmatrix} B \\ C \\ \theta \end{pmatrix}, {}^{B}_{C} \phi, {}^{B}_{C} \phi \end{pmatrix}$  and the position  ${}^{B}_{C}\mathbf{t}$ .
- ${}^{B}\mathbf{H}_{E}$ : The transformation matrix between the robot base and the end-effector. The robot kinematic parameters.
- The relationship between the laser pointer and the endeffector is defined with Euler angles  $\begin{pmatrix} E \\ L \\ \theta \end{pmatrix}$ ,  $\stackrel{E}{L} \phi$  and the position  $\stackrel{E}{L} \mathbf{t}$ .
- Plane equation : **a**
- Camera intrinsic matrix **K** and the lens distortion parameters  $(\kappa_1, \kappa_2, \rho_1, \rho_2)$

- For Hand-Eye-Workspace calibration, the goal is to estimate the relationship between the robot arm, the camera and the workspace plane. Assuming the robot arm and the camera are pre-calibrated. The estimated parameters include
  - $-\left({}^{B}_{C}\theta, {}^{B}_{C}\phi, {}^{B}_{C}\phi\right) \text{ and } {}^{B}_{C}\mathbf{t} : \text{ the transformation matrix between the fixed camera and the robot base.}$
  - $-\left({}_{L}^{E}\theta, {}_{L}^{E}\phi\right)$  and  ${}_{L}^{E}\mathbf{t}$ : the relationship between the laser pointer and the end-effector.
  - **a** : the plane parameters.



**Experiment Configuration** 

- The robot arm, Stäubli TX60, is operated to generate multiple laser spots on the plane in the view of camera. The measurement is the (*u*,*v*) position of laser spot. The optimal solution is obtained by minimizing the total position errors using nonlinear optimization method (the Levenberg-Marquardt method).
- The laser spot position in the image is obtained by two steps. First, the grey image converts to binary image using the simple threshold method. It can reject the image noise. Second, the average position of light pixels is the laser spot position.

#### Calibration Result

Parameter		30 samples	50 samples	Unit
	Initialized	Refined	Refined	
$({}^{B}_{C}\theta, {}^{B}_{C}\phi, {}^{B}_{C}\phi)$	(-100.14, -3.29, -172.69)	(-102.55, -6.15, -174.22)	(-102.60, -5.88, -173.53)	degree
${}^B_C \mathbf{t}^T$	[1343.73, -4.34, 319.28]	[965.56, -47.41, 440.90]	[976.59, -38.35, 436.62]	mm
$({}_{L}^{E}\theta, {}_{L}^{E}\phi)$	(-89.93, 80.21)	(-91.15, 84.45)	(-91.13, 84.36)	degree
${}^{E}_{L}\mathbf{t}^{T}$	[0, -30.14, 423.31]	[0, 8.55, 135.06]	[0, 7.49, 140.77]	mm
$\mathbf{a}^{T}$	[-70.39, 55.95, 690.76]	[-50.57, 35.52, 705.07]	[-61.34, 33.18, 702.78]	mm
RMS	30.32	1.0626	1.1494	pixel

# Advanced Hand-Eye-Workspace Calibration Conclusion

• The proposed calibration system is cost-efficient and flexible for any manipulator. It can be extended to calibrate the camera intrinsic parameters and the kinematic parameters of robot arm.