Engineering Application Note 105cFD100 6 DoF Robot

Robotics Coordinate System

Harry Li [‡], Ph.D.
Computer Engineering Department, San Jose State University
San Jose, CA 95192, USA

Email[†]: harry.li@ctione.com

***Abstract*—This note describes the coordinate systems for FD100 robotics system.**

I. Introduction

This note describes the coordinate systems for FD100 robotics system. The reference material for this discussion is from the following document:

1. CTI One's Technical Document (PPT)

`https://github.com/hualili/robotics-open_abb/blob/master/fd100/2021S-104-%232019%236-20-Turin-Robotics-RemoteControl-v1.0.pdf`

2. Turin Smart Robot General Operation Manual V2.0.pdf
3. Harry Li's deep reinforcement learning white paper (Part I on Deep Reinforcement Learning);
4. Robot Arm training github repository, https://github.com/rkandas/RobotArmMLAgentUnity;
5. Published paper by Google Deepmind team.
6. Tutorial on 6 DoF with ML model.

`https://medium.com/xrpractices/how-to-train-your-robot-arm-fbf5dcd807e1`

By manufacturer's product default definition, robot supports 4 coordinate systems [2], listed in the following table. In order to integrate Computer Vision capability, we have also added the additional coordinate systems, e.g., the viewer coordinate system to form the new capability.

TABLE I
Coordinate Systems

Category	Description
Joint Coordinate JCS	$j_1, ..., j_6$
Rectangular Coordinate WCS	$x_w - y_w - z_w$
User Coordinate UCS	$x_u - y_u - z_u$
Tool Coordinate TCS	$x_t - y_t - z_t$
Viewer Coordinate VCS	$x_v - y_v - z_v$

1. The Joint Coordinate System (JCS) $j_1, ..., j_6$ is defined to describe each axis angular position.

2. The Rectangular Coordinate System is the coordinate system whose origin is defined at the centre of the robot base and it servers as the base line reference system to establish the relationship with the rest of the coordinate system. Therefore it is also called world coordinate system (WCS) $x_w - y_w - z_w$.

3. The Tool Coordinate System (TCS) $x_t - y_t - z_t$ is defined on the fixture of wrist flange plate of the robot, and it can be modifed and defined by users. The effective direction d_t of the fixture is defined as the z axis z_t of the tool coordinate system, $d_t = z_t$.

4. The User Coordinate System (UCS) is defined on the work piece reacheable by the robot end effector, and is defined by users.

5. The viewer coordinate system (VCS) is defined to describe the images and videos captured on a given camera which (1) it looks at work space defined UCS, and (2) it looks at the origin of WCS, ideally, as more operational tasks added to the robot system, the 2nd, 3rd VCS may be added, in this case, we would have VCS_i for i more than 1.

All the experiments will be carried out in FD100 robotics system shown in this figure.

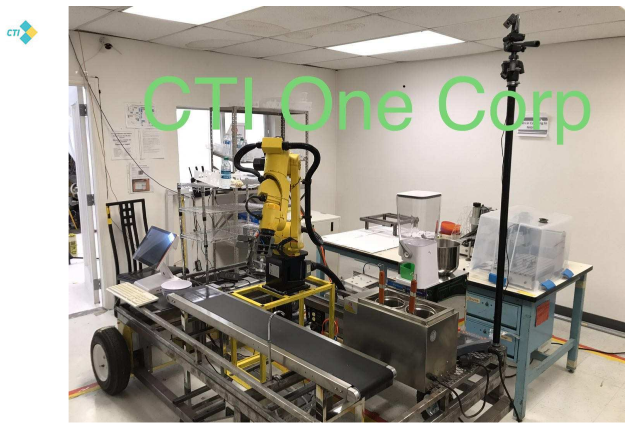


Fig. 1. All the experiments will be carried out in FD100 robotics system.

II. World (Rectangular) Coordinate System

The world (also known as rectangular) coordinate system is the right hand (RH) coordinate system, denoted as $x_w - y_w - z_w$, with its origin at the center of the FD100 robot as shown in the figure below.

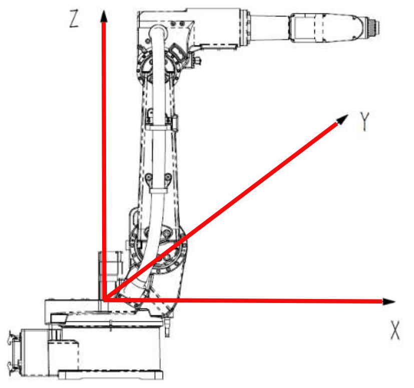


Fig. 2. The world coordinate system $x_w - y_w - z_w$ with its origin at the center of the FD100.

The relationship between the world coordinate system and the robot joints is illustrated below.

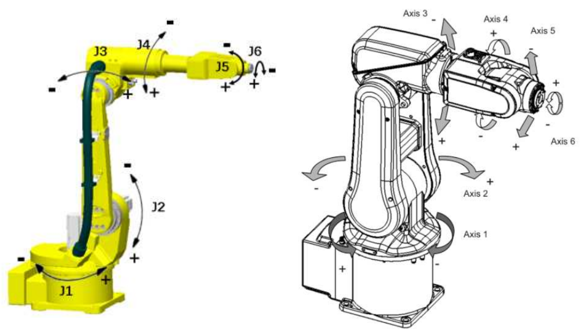


Fig. 3. The robot joints diagram.

The figure is from the Turin Robot User Guide, pp. 23.

III. User Coordinate System

An user coordinate system UCS is RH system denoted as $x_u - y_u - z_u$ which is defined by user to describe the work space which the robot end effector can reach.

According to the technical report [1] and [2], to set an user coordinate system, we use 3-point method which define 1 point at the origin, and 1 point on the positive x_u axis, and 1 point on the $x_u - y_u$ plane. The operation steps for this is described below according to reference [1] and [2]:

Step 1. Move the robot with the operating tool to the origin of the (user) coordinate system and record the point.

Step 2. Move the robot with the operating tool to any point on the x-axis and record the point.

Step 3. Move the robot with the operating tool to any point in the xy plane that belongs to the first quadrant and record the point.

Step 4. Save the calculation results. Hence the user coordinate system is established.

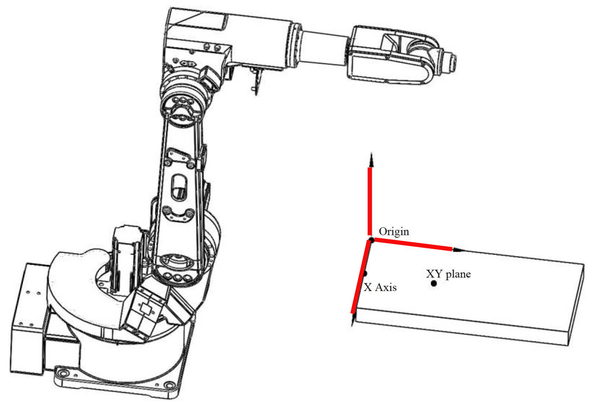


Fig. 4. An user coordinate system $x_u - y_u - z_u$ to describe the work space.

IV. Tool Coordinate System

A tool coordinate system TCS is denoted as $x_t - y_t - z_t$ which is defined by user to operate the robot end effector for its operation. This coordinate system requires a process to set it up by the user. This Coordinate system is defined on the tool. The effective direction of tool is defined as the z_t axis, and the x_t axis and y_t axis are defined according to the right-hand rule.

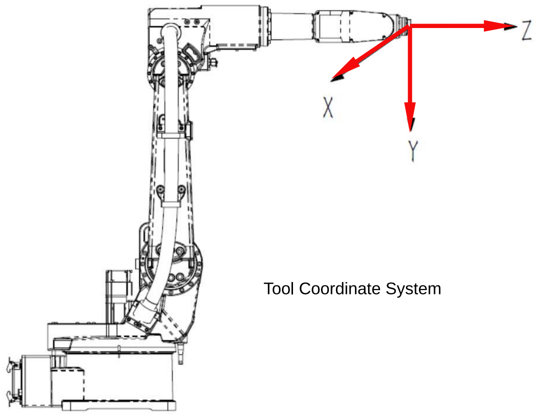


Fig. 5. An user coordinate system $x_u - y_u - z_u$ to describe the work space.

Set up TCS take 6 steps described here based on reference [6]. Illustrated in the figure here is 6 different tool points. The user will have to move the robot to each of the point and record it.

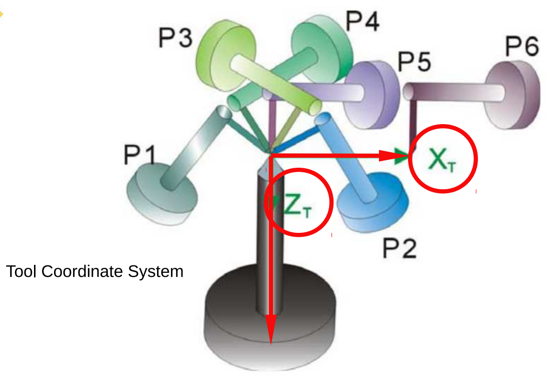
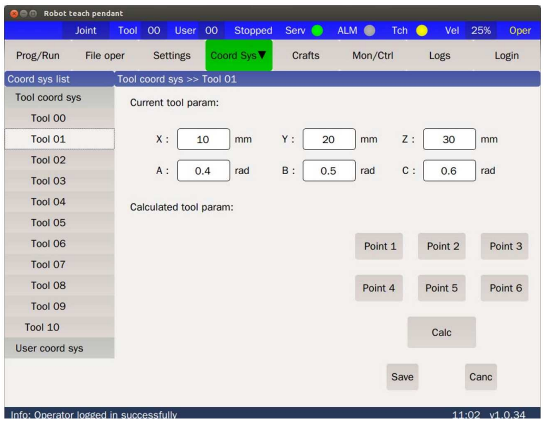


Fig. 6. Illustration of 6-point TCS setup process.

The set up process of TCS is described in the reference document [6]



Tool coord sys >> Tool 01

Current tool param:

X : 10 mm Y : 20 mm Z : 30 mm

A : 0.4 rad B : 0.5 rad C : 0.6 rad

Calculated tool param:

Info: Operator logged in successfully 11:02 v1.0.34

Fig. 7. The GUI from robot teach pandant for setting up TCS.

V. Experiment Set Up to Test User Coordinate

At CTI One's robotics lab, an experiment setup allows FD100 robot to test its user space with the computer vision system's imaging data $I_v(x, y)$ which is defined in viewer coordinate system $x_v - y_v - z_v$.

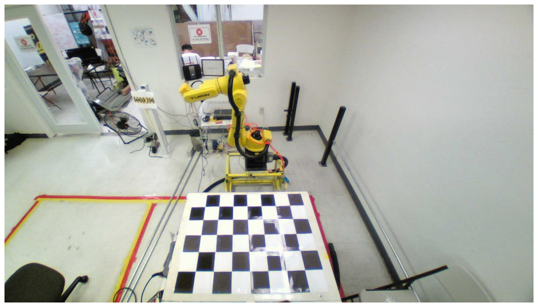


Fig. 8. Test an user space with the computer vision system's imaging data $I_v(x, y)$ in the viewer coordinate system $x_v - y_v - z_v$.

To prepare for the Computer Vision guided robot experiment, build a checker board and set the checker board in a visible location for video camera CAM_1 and to allow robot end effector reachable to any of the point on the board, as illustrated in the UCS photo in this section.

Computer Vision system plays a significant role in robotics operation:

1. Computer Vision system with deep learning capability can identify an object.

2. Then a mapping from a viewer coordinate system (VCS) $x_v - y_v - z_v$ to world coordinate system (WCS) $x_w - y_w - z_w$ allows the object location to be calculated in the WCS.

3. Compute the inverse kinematics. Note, inverse kinematics is the mathematical process of computing the joint parameters needed to place the end effector of a kinematic chain, e.g., a robot manipulator in a given position and orientation relative to the start of the chain.

4. Move the robot to the joint parameters based on certain rules or guidelines, such as, best long term reward, to reach and grip the object.

Hence, the mathematical forumulation for the mapping from a viewer coordinate system (VCS) $x_v - y_v - z_v$ to world coordinate system (WCS) $x_w - y_w - z_w$ is crucial step to complete this computer vision based automated process. We just cover the basic introduction here, we will give detailed discussion on this technique and its realization in a separate engineering note.

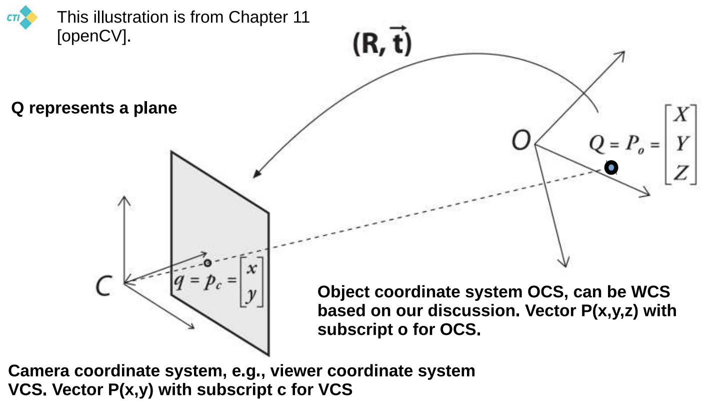

This illustration is from Chapter 11 [openCV].
(R, $\vec{t}$)
Q represents a plane
O
$Q = P_o = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$
C
$q = P_c = \begin{bmatrix} x \\ y \end{bmatrix}$
Object coordinate system OCS, can be WCS based on our discussion. Vector P(x,y,z) with subscript o for OCS.
Camera coordinate system, e.g., viewer coordinate system VCS. Vector P(x,y) with subscript c for VCS

Fig. 9. Mapping with rotation and translation.

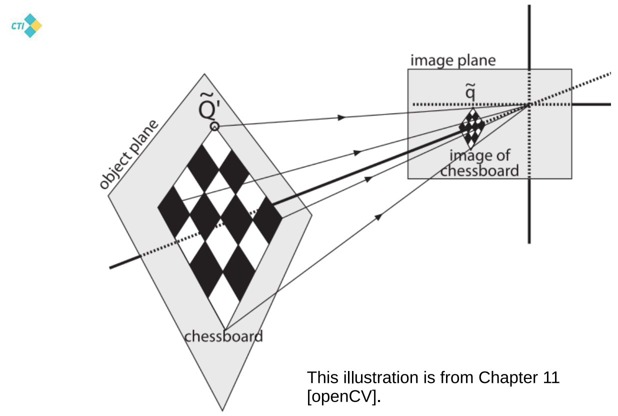


Fig. 10. Mapping example on a checker board with rotation and translation.

Mathematical analysis leads to a homography matrix H. The homography matrix H relates the positions of the point(s) P_o on a source image plane to the points on the destination imager plane P_v by the following equation

$$P_o(x_o, y_o, z_o) = H^{-1} P_v(x_v, y_v). \tag{1}$$

OpenCV provides cvFindHomography() function, which takes a list of correspondences and returns the homography matrix H that best describes those correspondences. A minimum of four points to solve for H is needed, but many more would be better as we can solve over-determined system. In the case of chessboard bigger than 3-by-3, more matching pair of points is not a problem. Using more points is better for noise reduction.

VI. Robot Movement

FD100 robot has 3 different types of movement listed in the following table.

TABLE II
THREE TYPE OF MOVEMENT

Category	Description
Joint movement	MOVJ
Linear movement	MOVL
Arc movement	MOVC

Joint movement is the movement defined by the joint angular change.

Linear movement is the movement to move the robot to the present (demonstration) point through linear path. During the movement, the robots movement control point shall follow straight line, and the gesture of the fixture shall be changed automatically as shown in the following figure.

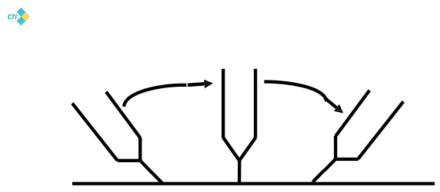


Fig. 11. Illustration of linear movement with the gestrure of fixture.

Arch movement is needed when the robot needs to move to the present demonstration point through arc path. Three points determine an unique arc, so three points are needed to define an arc movement. Arc movement is realized by starting from the present point, passing through the first point (auxiliary point) and reaching the second point (ending point).

Acknowledgment

I would like to express my thanks to CTI One engineering member and CTI One engineering intern team, as well as to SJSU graduate research assistants for letting me introducing DRL in Robotics to our next project and for the codings in Deep Learning, Robotis and embedded systems.

Quiz

1. How many coordinate systems have we defined in this note? list each of them and explain its definition?
2. Draw illustration of JCS, WCS, UCS, TCS, VCS?
3. What is homography matrix H? how many corresponding pair of points needed to define H?
4. Suppose H matrix is found, then how would you find P_o on a source image plane? (Answer: use $P_o(x_o, y_o, z_o) = H^{-1} P_v(x_v, y_v)$.)
5. How many type of movement? name each of them, draw 3D motion trajectory to illustrate each of the movement.
6. Design a peak and place operation for a robot to pick a coffee cup on a table top and place it back on the same table top but at different location by listings each of the movement type, (assuming the robot start from an initial position (x_t, y_t, z_t).

References

[1] [Franceschetti, 2020] Andrea Franceschetti, Elisa Tosello, Nicola Castaman, and Stefano Ghidoni, "Robotic Arm Control and Task Training through Deep Reinforcement Learning", https://arxiv.org/pdf/2005.02632.pdf, May 2020.
[2] [Reinforcement, 2020] Reinforcement learning, https://en.wikipedia.org/wiki/Reinforcement_learning, 2020.
[3] [OpenCV, 2017] Learning OpenCV, Book, 2017.