



Implementation of Resolved Motion Rate Controller with 5-Axis Robot Manipulator Arm

Pradya Prempraneerach^{1*}, Pasan Kulvanit²

¹Department of Mechanical Engineering, Rajamangala University of Technology Thanyaburi, Pathumthani, Thailand 12110

²Department of Science Service (DSS), Ministry of Science and Technology, Ratchatewi, Bangkok, Thailand 10400

*Corresponding Author: Tel: (662) 5493430, Fax: (662) 5493432,

E-mail: ppradya@gmail.com

Abstract

In this research work, an industrial-process prototype, particularly pick-and-place of machined parts, is developed by using an OWI-535 5-axis robot manipulator arm and controlling with Faulhaber motion control systems. To make an end-effector of the OWI-535 robot arm following desired positions with specified joint velocities, the inverse kinematics technique, known as the *resolved motion rate controller*, can help generating motion trajectories automatically. This inverse kinematics technique can be implemented with the Jacobian pseudo-inverse or Jacobian singularity-robust inverse. This technique does not require to inversely solve algebraic or geometric kinematic equations. Computation of the robot inverse kinematics is simulated in Matlab and then a motion control of the OWI-535 robot arm is performed by the "Faulhaber Motion Manager". The pick-and-place motion of the OWI-535 robot arm agrees with its kinematics simulation very well.

Keywords: inverse kinematics, robot manipulator arm, Jacobian pseudo inverse, Jacobian singularity-robust inverse

1. Introduction

Nowadays, robot manipulator arms with a computer control are extensively used in most industrial factories to increase efficiency and to deal with repeated and/or hazardous tasks, such as in semiconductor wafer production processes [3]. Performing repetitive tasks continuously and repeatedly in the production line, human fatigue can lead to accident/injury or might result in a long-term health problem. On the other hand, robot manipulator arm can repetitively carry

huge and heavy machine parts from one production line to another.

Redundancy of the robot manipulator can provide many advantages in the case when the degree of freedom of a given task is less than that of the manipulator. When a joint failure occurs, the robot manipulator is still able to perform its task reliably [7]. Moreover, singular configuration of the robot manipulator can be avoided by control of redundancy using the Jacobian singularity-robust inverse [4].

In this work, we apply the redundancy control to inverse kinematics problem of the OWI-535 5-axis robot arm so that it can perform a pick-and-place operation. In Section 2, the structure and forward kinematics formulation of this robot arm are introduced along with its hardware description. The resolved motion rate control, which is a technique to solve the inverse kinematics problem by incorporating either Jacobian pseudo-inverse or Jacobian singularity-robust inverse, describes in Section 3. Then, comparisons of kinematics simulations and experimental results of the pick-and-place operation of the OWI-535 robot arm is shown in Section 4.

2. Structure and Forward Kinematics of OWI-535 Robot Arm

The structure of the OWI-535 robot arm consists of 4 revolute joints connecting by three linkages and an end-effector, which is composed of two 4-bar linkage mechanisms. The length between each joint is as show in Fig. 1. Fig. 2 shows a limitation of angular rotation of 4 revolute joints, which can specify a reachable workspace of the OWI-535 robot arm.

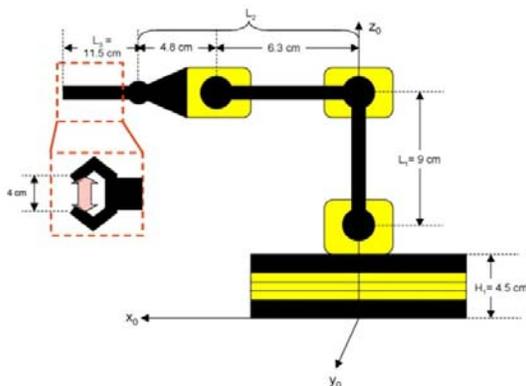


Fig. 1 Side view of the OWI-535 robot arm structure.

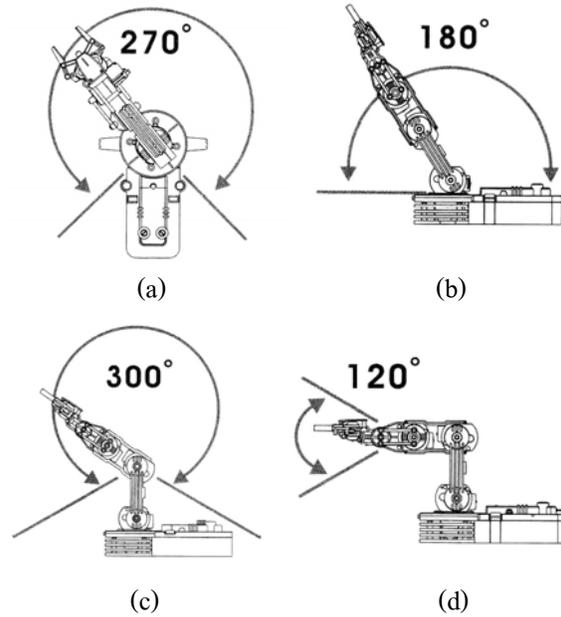


Fig 2 Limitation of angular rotation of revolute joint 1(a), joint 2 (b), joint 3 (c), joint 4 (d) [5].

The joints' rotation angle of each joint and frame assignment of the OWI-535 robot arm, used in forward and inverse kinematic calculations are illustrated in Fig. 3. Advantage of this frame assignment is that the z axes of frame 2 to 4 are parallel; as a result, a Jacobian calculation is straightforward.

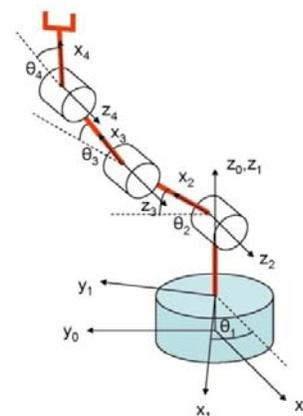


Fig. 3 A frame assignment for the OWI-535 robot arm.

2.1 Forward kinematics calculation of OWI-535 robot arm

The Denavit-Hartenberg parameters [1] for each joint and link are shown in Table 1, where L_1 , L_2 , L_3 and H_1 values are given in Fig. 1.

Table. 1 Denavit-Hartenberg parameters of the OWI-535 robot arm

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	H_1	θ_1
2	90°	0	0	θ_2
3	0	L_1	0	θ_3
4	0	L_2	0	θ_4
5	0	L_3	0	0

The forward kinematics from frame 0 to frame 4 can be computed by a transformation matrix $T_{04} = T_{01}T_{12}T_{23}T_{34}$. These transformation matrices $(T_{01}, T_{12}, T_{23}, T_{34})$ are given in Appendix. Given joint rotation angle, we can calculate the distance from the base or frame 0 to the end-effector or frame 4 using T_{04} .

2.2. Hardware and motion control description

The main improvement of the OWI-535 robot arm is to modify gear boxes, which drive all links. All original dc motors are replaced by 2224SR dc motors and IE2-512 magnetic encoders from Faulhaber [2]. Five MCDC 3006/06S Faulhaber motion control systems [2] can control five motors' incremental position and speed, as shown in Fig. 4. All Faulhaber motion controllers can be connected with a computer through RS232 interface.



Fig. 4 Hardware setup

3. Inverse Kinematics using Resolved Motion Rate Control

The Resolved Motion Rate Control (RMRC) using either Jacobian pseudo-inverse and Jacobian singularity-robust inverse is a technique for solving the inverse kinematics of robot arm. In the first step of this algorithm, initial joint angles of the OWI-535 robot arm must be specified, then we can calculate the Jacobian matrix (J), which relates the angular speed of i joint (ω_i or $\dot{\theta}_i$) to the end-effector or 4th link velocity (V_4), as shown in Eq. (1) below

$$\begin{bmatrix} \dot{x}_4 \\ \dot{y}_4 \\ \dot{z}_4 \end{bmatrix} = \begin{bmatrix} \partial x / \partial \theta_1 & \partial x / \partial \theta_2 & \partial x / \partial \theta_3 & \partial x / \partial \theta_4 \\ \partial y / \partial \theta_1 & \partial y / \partial \theta_2 & \partial y / \partial \theta_3 & \partial y / \partial \theta_4 \\ \partial z / \partial \theta_1 & \partial z / \partial \theta_2 & \partial z / \partial \theta_3 & \partial z / \partial \theta_4 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \end{bmatrix} \quad (1)$$

where

$$\begin{aligned} \partial x / \partial \theta_1 &= -(L_1 \cos(\theta_2) + L_2 \cos(\theta_2 + \theta_3) + L_3 \cos(\theta_2 + \theta_3 + \theta_4)) \sin(\theta_1) \\ \partial x / \partial \theta_2 &= -(L_1 \sin(\theta_2) + L_2 \sin(\theta_2 + \theta_3) + L_3 \sin(\theta_2 + \theta_3 + \theta_4)) \cos(\theta_1) \\ \partial x / \partial \theta_3 &= -(L_2 \sin(\theta_2 + \theta_3) + L_3 \sin(\theta_2 + \theta_3 + \theta_4)) \cos(\theta_1) \\ \partial x / \partial \theta_4 &= -(L_3 \sin(\theta_2 + \theta_3 + \theta_4)) \cos(\theta_1) \\ \partial y / \partial \theta_1 &= (L_1 \cos(\theta_2) + L_2 \cos(\theta_2 + \theta_3) + L_3 \cos(\theta_2 + \theta_3 + \theta_4)) \cos(\theta_1) \\ \partial y / \partial \theta_2 &= -(L_1 \sin(\theta_2) + L_2 \sin(\theta_2 + \theta_3) + L_3 \sin(\theta_2 + \theta_3 + \theta_4)) \sin(\theta_1) \\ \partial y / \partial \theta_3 &= -(L_2 \sin(\theta_2 + \theta_3) + L_3 \sin(\theta_2 + \theta_3 + \theta_4)) \sin(\theta_1) \\ \partial y / \partial \theta_4 &= -(L_3 \sin(\theta_2 + \theta_3 + \theta_4)) \sin(\theta_1) \\ \partial z / \partial \theta_1 &= 0 \\ \partial z / \partial \theta_2 &= (L_1 \cos(\theta_2) + L_2 \cos(\theta_2 + \theta_3) + L_3 \cos(\theta_2 + \theta_3 + \theta_4)) + H_1 \\ \partial z / \partial \theta_3 &= (L_2 \cos(\theta_2 + \theta_3) + L_3 \cos(\theta_2 + \theta_3 + \theta_4)) + H_1 \\ \partial z / \partial \theta_4 &= (L_3 \cos(\theta_2 + \theta_3 + \theta_4)) + H_1 \end{aligned}$$

In the second step, instead of solving inverse kinematic equation mathematically from the geometric of this 5-axis robot arm, the RMRC technique can compute joint rotation speed (θ_i) at each time step, given the end-effector desired position and velocity. Joint angular velocity can be written as a linear combination of desired end-effector velocity and end-effector position error multiplying with the Jacobian inverse, as shown in Eq. (2).

$$\dot{\theta} = J^{-1}(\dot{x}_d + K(x_d - x)) \quad (2)$$

where K is a gain coefficient for adjusting a convergence speed of the actual robot arm position. The larger the K coefficient is, the faster the actual end-effector position converges to the desire end-effecotr position. Using the forward kinematics, the end effectors position (X_4) at different time steps can be computed from joints' rotational angle by integrating $\dot{\theta}$ from Eq. (2) using the Euler integration.

The RMRC technique employs the Jacobian inverse, J^{-1} . In this work, two types of Jacobian inverse are used in the RMRC technique. The Jacobian pseudo-inverse ($J^\#$) can be implemented easily from Eq. (3), however a singularity of matrix inverse might occur in some configurations

$$J^\# = J^T(JJ^T)^{-1} \quad (3)$$

To avoid the singularity in Jacobian pseudo-inverse computation, the Jacobian singularity-robust inverse (J^*) [4] in Eq. (4) incorporates an additional diagonal gain matrix such that the trajectory will deviate from the singular configuration.

$$J^* = J^T(JJ^T + k_1 I)^{-1} \quad (4)$$

$$\text{where } k_1 = \begin{cases} k_0 \left(1 - \frac{\omega}{\omega_0}\right)^2 & \text{if } \omega < \omega_0 \\ 0 & \text{if } \omega \geq \omega_0 \end{cases} \quad \text{and}$$

$$\omega = \sqrt{\det(JJ^T)} \quad \text{and } (\omega_0, k_0) \text{ are constants.}$$

When $\omega_0 = 0$ or $k_0 = 0$, the Jacobian singularity-robust inverse becomes the Jacobian pseudo inverse. Using Matlab, the motion trajectories of the robot arm can be generated after solving the inverse kinematics problem. Then, the RMRC technique is implemented on the OWI-535 robot arm and a parameters' effect is examined.

4. Simulation Results of RMRC Technique

Combining the RMRC technique with Euler integration, the inverse kinematics in Eq. (2) can be solved for joint rotation angles (θ_i). In the following examples, integration time step (dt) of the Euler integration is 0.01 second. In the pick-and-place simulation, a object is assumed to originally locate at position of (15,15,3) cm, then the robot arm moves this object to a new position of (15,-15,3) cm. The initial joint angle ($\theta_1, \theta_2, \theta_3, \theta_4$) = (0.01, $\pi/2, 0, 0$).

Trajectory for each link with Gain = 2 and time interval of 0.01sec

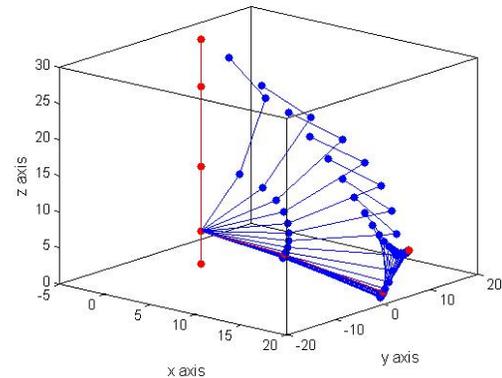


Fig. 5 The trajectory of robot arm using the RMRC technique using $J^\#$ with $K = 2$ at every 0.1 second from position of (0,0,31.1) cm to position of (15,15,3) cm.

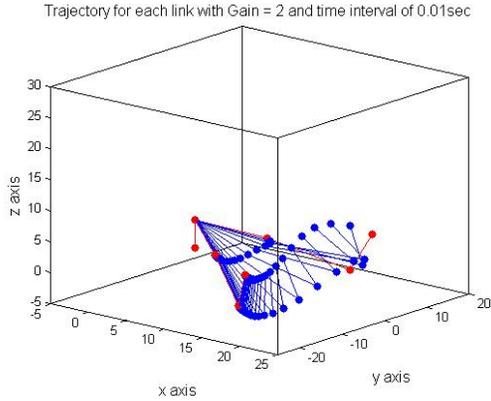


Fig. 6 The trajectory of robot arm using the RMRC technique using $J^\#$ with $K = 2$ at every 0.1 second from position of (15,15,3) to position of (15,-15,3) cm.

Fig. 5 and 6 show Matlab simulation results of the RMRC technique using $J^\#$ to calculate joint rotation angles at every 0.1 second. Then, we substitute joint angles into Jacobian in Eq. (1) to compute joints' position in forward kinematics equations. In the first section ($0 < t \leq 5$ second), the robot arm moves from the initial position $(x_{01}, y_{01}, z_{01}) = (0, 0, 31.1)$ cm to the first desired position $(x_{d1}, y_{d1}, z_{d1}) = (15, 15, 3)$ cm, as shown in Fig. 5 and 7. In the second section ($5 < t \leq 10$ second), the robot arm moves from the first desired position $(x_{d1}, y_{d1}, z_{d1}) = (15, 15, 3)$ to the second desired position $(x_{d2}, y_{d2}, z_{d2}) = (15, -15, 3)$ cm as shown in Fig. 6 and 8. Both trajectories in the first and second sections in Fig. 5 and 6 use the gain, K , of 2. To speed up motion trajectories from initial to desired positions, K is set to 10, as shown in Fig. 7 and 8, thus, the end effector reaches the desired position in fewer time steps. In addition, the initial position (x_{01}, y_{01}, z_{01}) is a singular configuration because it locates at a boundary of the reachable workspace. As a result, the RMRC

technique using $J^\#$ causes the robot arm to approach (x_{d1}, y_{d1}, z_{d1}) with the full extension of link 2 and 3, which is close to a singularity. However, using J^* in the RMRC technique, the motion trajectory is deviated from the singularity because the gain k_1 in Eq. (4) prevents the inverse of $(JJ^T + k_1 I)$ to approach infinity near the singularity. Moreover, in the first section, all joints' velocity is much smaller when comparing Fig. 7 with Fig. 5.

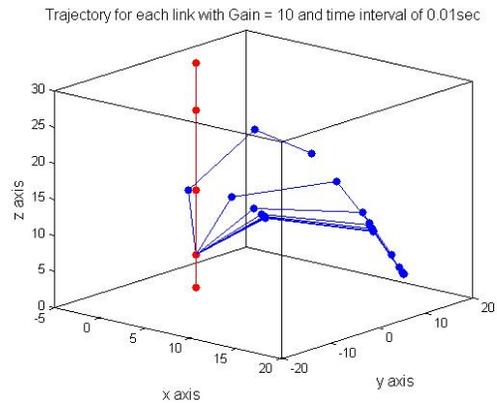


Fig. 7 The trajectory of robot arm using the RMRC technique using J^* with $K = 10$ at every 0.1 second from position of (0,0,31.1) cm to position of (15,15,3) cm.

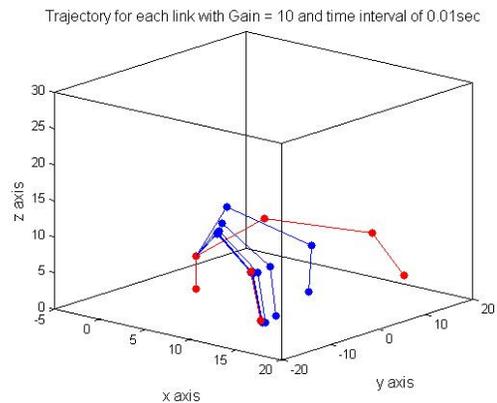


Fig. 8 The trajectory of robot arm using the RMRC technique using J^* with $K = 10$ at every 0.1 second from position of (15,15,3) to position of (15,-15,3) cm.

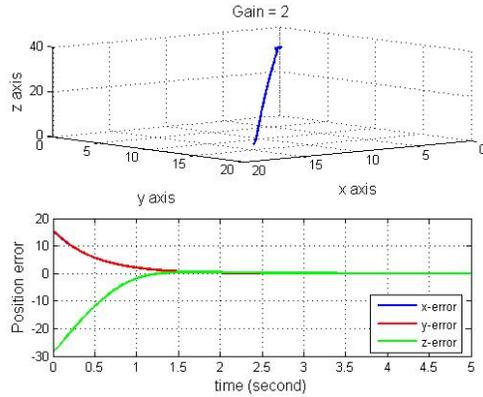


Fig. 9 Using $J^\#$, end-effector trajectory (top) and position error (bottom) of the 1st section with $K = 2$.

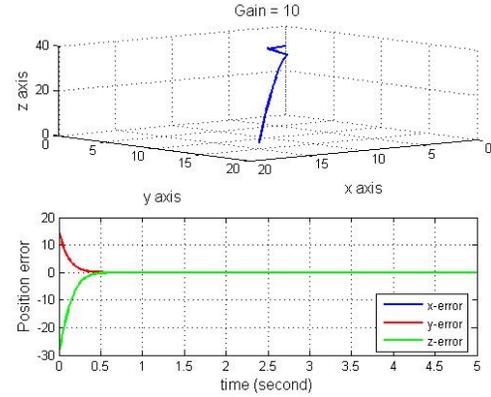


Fig. 12 Using J^* , end-effector trajectory (top) and position error (bottom) of the 1st section with $K = 10$.

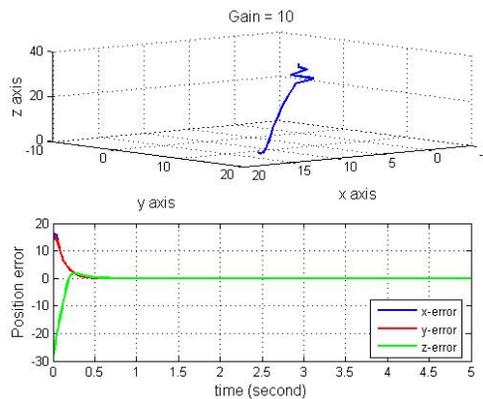


Fig. 10 Using $J^\#$, end-effector trajectory (top) and position error (bottom) of the 1st section with $K = 10$.

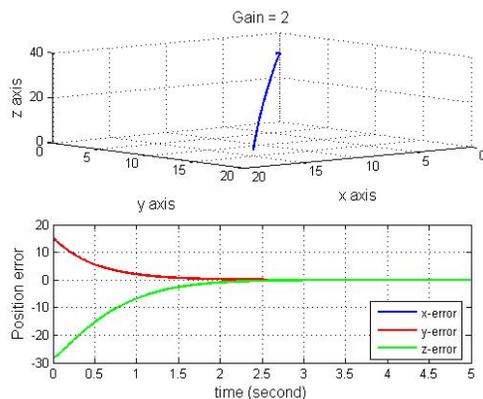


Fig. 11 Using J^* , end-effector trajectory (top) and position error (bottom) of the 1st section with $K = 2$.

We also examine the end-effector trajectory as well as x-,y-,z-axis position error in the first section using $J^\#$ with $K = 2$ and 10, as shown in Fig. 9 and 10 respectively, and using J^* with $K = 2$ and 10, as shown in Fig. 11 and 12. With a large gain, K , the end-effector position error quickly converges to zero. The RMRC technique using $J^\#$ has larger end-effector's and joints' velocities than that using J^* . Especially, using $J^\#$ with $K = 10$ in the first section, there exist a large oscillation when this robot arm is close to the singularity initially, as shown in Fig. 10. However, the motion trajectory as well as joints' velocity becomes smoother in Fig. 11 and 12. In the second section, when the robot arm is away from the singularity configuration, the motion trajectories using both $J^\#$ and J^* are continuous and very smooth, as seen in Fig. 13 and 14. Notice that the position errors in x-,y- and z-axis converge to zero eventually.

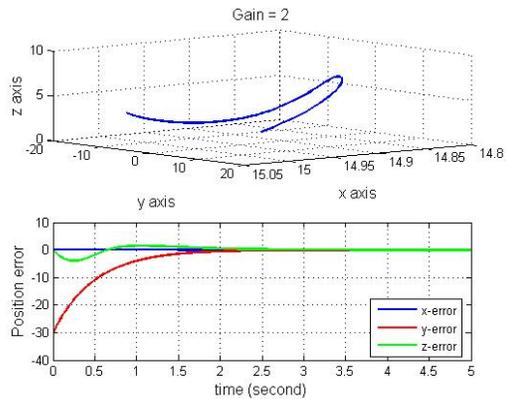


Fig. 13 Using $J^\#$, end-effector trajectory (top) and position (bottom) of of the 2nd section with $K = 2$.

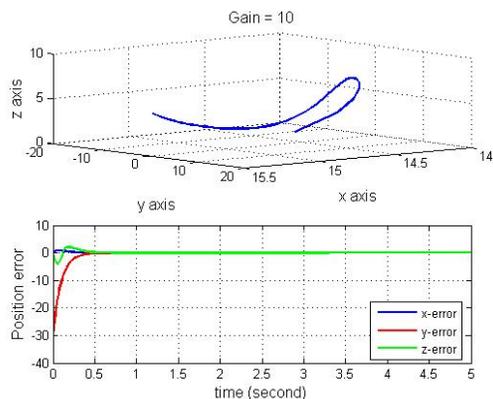


Fig. 14 Using $J^\#$, end-effector trajectory (top) and position error (bottom) of of the 2nd section with $K = 10$.

5. Experimental Results of RMRC Technique

The pick-and-place operation of the OWI-535 robot arm, similar to the simulation scenario, is implemented with "Faulhaber Motion Manager" software in VBscript language. The RMRC technique using the Jacobian singularity-robust inverse combining with forward kinematic are developed in Motion Manager to control the end-effector position. Because of a limitation of Motion Manager software that operates each individual motor sequentially, the end effector can not follow the desired speed. Nonetheless, the robot arm approaches the desired end-

effector position accurately. Motion trajectories starts from the same initial position (x_{01}, y_{01}, z_{01}) , as shown in Fig. 15, and moves to the first desired position (x_{d1}, y_{d1}, z_{d1}) , as shown in Fig. 16, and then moves to the second desired position (x_{d2}, y_{d2}, z_{d2}) , as shown in Fig. 17.



Fig. 15 End effector of OWI robot arm located at the initial position (x_{01}, y_{01}, z_{01}) .



Fig. 16 End effector of OWI robot arm located at the first desired position (x_{d1}, y_{d1}, z_{d1}) .



Fig. 17 End effector of OWI robot arm located at the second desired position (x_{d2}, y_{d2}, z_{d2}) .

6. Summary

The inverse kinematics problem of the OWI-535 robot arm has been implemented with the resolved motion rate control (RMRC) technique both in Matlab for pick-and-place simulations and in Faulhaber Motion Manager software for motion trajectory of the actual OWI-535 robot arm. The joints' position of simulation results agrees well with that of the actual OWI robot arm. In the RMRC method, the Jacobian pseudo inverse might lead to the singular configuration, but the Jacobian singularity-robust inverse prevents the robot arm to approach the singularity.

7. Acknowledgement

This work is supported by A New Researcher Scholarship of CSTS, MOST 2551, Coordinating Center for Thai Government Science and Technology Scholarship Students, National Science and Technology Development Agency, Thailand.

8. References

- [1] Craig, J.J. (1989). Introduction to robotics: mechanics and control, 2nd edition, ISBN: 0-201-09528-9, Addison-Wesley Publishing Company, Inc.
- [2] Faulhaber, Instruction Manual for Motion Controller Series MCDC 3003/06S, Version 4th edition (2009), URL: <http://www.faulhaber.com>, access on 15/07/2010
- [3] Kim D.I., Kim J.K., Sung H.K., and Kim S. (1995), Sensor-Based Navigation Control and Calibration of a Wafer-Handling Mobile Robot, paper presented in Thirtieth IEEE Industry Applications Conference Annual Meeting 1995, Vol. 3, 1995, pp 1922-1929.

[4] Nakamura N., Hanafuse H. (1985), Inverse Kinematic Solutions with Singularity Robustness for Robot Manipulator Control, paper presented in 85 ASME Winter Annual Meeting, Miami Beach, 1985, Vol. 15, pp. 193-204.

[5] OWI, Robotic Arm Edge, Instruction Manual

[6] Sánchez A.J. and Martínez J.M. (2000), Robot-arm Pick and Place Behavior Programming System Using Visual Perception, paper presented in Proceedings. 15th International Conference on Pattern Recognition, Vol. 4, 2000, pp. 507-510.

[7] Yoshikawa, T. (1990). *Foundations of robotics: analysis and control*, ISBN: 0-262-24028-9, The MIT Press, Massachusetts.

9. Appendix

The transformation matrices from joint 0 to 1 (T_{01}), from joint 1 to 2 (T_{12}), from joint 2 to 3 (T_{23}), from joint 3 to 4 (T_{34}) can be written below.

$$T_0^1 = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & H_1 \\ 0 & 0 & 0 & 1 \end{bmatrix};$$

$$T_1^2 = \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \sin(\theta_2) & \cos(\theta_2) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix};$$

$$T_2^3 = \begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & L_1 \\ \sin(\theta_3) & \cos(\theta_3) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix};$$

$$T_3^4 = \begin{bmatrix} \cos(\theta_4) & -\sin(\theta_4) & 0 & L_2 \\ \sin(\theta_4) & \cos(\theta_4) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$