We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



21

Error Modeling and Accuracy of Parallel Industrial Robots

Hongliang Cui and Zhenqi Zhu

1. Introduction

Most industrial robots are open-chain mechanisms constructed of consecutive links connected by rotational or prismatic joints of one degree of freedom. These serial manipulators have large workspace, high dexterity and good maneuverability. However, due to their serial structure they exhibit low stiffness and poor positioning accuracy. As a result, their use in applications that require large loads (e.g. machining) and high accuracy, is limited. In the case of a parallel manipulator, the end-effector is attached to a moveable plate which is supported in-parallel by a number of actuated links. As a result, these parallel manipulators are anticipated to possess the following advantages, compared with serial manipulators: 1) high force/torque capacity since the load is distributed to several in-parallel actuators; 2) high structural rigidity; and 3) better accuracy due to less cumulative joint errors.

A large number of publications dealing with the accuracy of the serial manipulators appeared in the past. These include topics on error modeling effects of manufacturing tolerance on pose accuracy and numerous calibration strategies. However, very few publications dealing with the same issue as related to parallel manipulators can be found. Since high accuracy is generally believed to be one of their advantages compared to that of serial manipulators, it is important to address this issue. The purpose of this research is to establish the kinematic and error models for evaluating the effects of manufacturing tolerances, installation errors and stiffness effect on the accuracy of a parallel robotic system.

In order to evaluate the accuracy of parallel robotic system, it is necessary to develop a kinematic model which will accommodate the above errors. Based on this model, algorithms for forward, inverse kinematics and error modeling of the parallel robot are presented. These algorithms with a set of typical tolerances were used to compute the pose errors which include three translational and three angular errors.

Open Access Database www.i-techonline.com

Source: Industrial-Robotics-Theory-Modelling-Control, ISBN 3-86611-285-8, pp. 964, ARS/plV, Germany, December 2006, Edited by: Sam Cubero

Manufacturing tolerances, **installation errors** and **link offsets** cause deviations with respect to the nominal kinematic parameters of the robot system. As a result, if the nominal values of these parameters are used within the robot system control software, the resulting pose of the system will be inaccurate. Accuracy of a robot is the closeness with which its actual pose matches the pose predicted by its controller. A robot normally designed for repeated work such as spray painting, pick and place, etc., has high repeatability but low accuracy. An accurate robot is required in applications where off-line programming is involved. To a large extent, robot inaccuracy is induced by the propagation of *geometric errors*, *compliance errors* and *time-variant thermal errors.* The geometric errors of a robot come from manufacturing imperfections, misalignments or joint wear. Compliance errors are due to the flexibility of robot joints and link deflection under self-gravity and external load. The compliance errors also depend on the robot's changing position. Thermal errors result from thermal distortion and expansions of robot components due to internal and external heat sources such as motors, bearings and ambient temperature change.

Link and joint flexibility has a significant impact on robot performance and stability. Link gravity and external payload cause the deflection of links and flexible joints, and therefore degrade the robot performance. Link compliance effects are represented by six differential component changes: three translational and three rotational changes. This paper presents a systematic methodology for estimating the compliance characteristics of serial and parallel manipulators due to external concentrated load/deflection. In related experiments, special measurement tools and sensors are necessary to identify the stiffness of driving joints.

Also in this paper a general methodology is presented to calibrate and compensate for robot compliance errors and thermal errors in addition to geometric errors, An error synthesis model based on the Denavit-Hartenberg (D-H) convention is derived for simultaneously considering geometric errors, compliance errors and thermal errors. Based on this model a general methodology is developed to calibrate geometric errors, compliance errors and thermal errors. Experimental results show that the accuracy of the robot is improved by an order of magnitude after the compensation.

1.1 Serial and Parallel Robots

Robots are representative of mechanics devices which integrate aspects of manipulation, sensing, control, and communication. Rarely have so many technologies and scientific disciplines focused on the functionality and performance of a system as they have done in the fields of robot development and application. Robotics integrates the state of the art of many front-running technologies. Large efforts have been made to define an industrial robot and to

classify its application by industrial branches so that remarkably precise data and monitoring are available today.

The task of an industrial robot in general is to move a body (workpiece) with six maximal Cartesian spatial DOF (three translations, three rotations) to another point and orientation within a workspace. The complexity of the task determines the required kinematic configuration. The number of DOFs determines how many independently driven and controlled axes are needed to move a body in a defined way. Industrial robots normally have up to four principal arm axes and three wrist axes. While many exciting robot structures use serial kinematic chains, some parallel kinematic structures have been adopted for a variety of tasks. Typical configurations of industrial robots are shown in Figure 7. Most closed-loop kinematics is based on the so-called hexapod principle (Stewart platform, 1965), which represents a mechanically simple and efficient design. The structure is relatively stiff and enables relatively high positioning accuracy and high speeds, but workspace or working volume is limited.

Parallel or closed-chain linkages and serial or open-loop kinematic chains have been substantially investigated over last several decades. A closed-chain linkage, which usually has a limited number of degrees of freedom, is not applicable as a general-purpose robot kinematic configuration. A serial kinematic chain can provide a large workspace, but with less rigidity and load-carrying capacity compared with a parallel kinematic chain. The fully parallel-driven manipulators such as Stewart-platform have been investigated by many researchers. In general, the workspace of a robot arm consisting of only parallel kinematic chains is relatively small. Currently, there has been an increasing interest in the design of hybrid or serial-parallel robot manipulators which can provide salient features of both serial and parallel kinematic chains. An appropriately designed hybrid robotic manipulator will have a large load-carrying capacity and workspace, and yet be comparatively small and lightweight.

The TAU parallel configuration (Figure 1) is rooted in a series of inventions and was masterminded by Torgny Brogardh, 2000; 2001; 2002. The configuration of the robot simulates the shape of "τ" like the name of the Delta robot named after the "∇" shape configuration of the parallel robot. As shown in Fig. 1.1, the basic TAU configuration consists of three driving axes, three arms, six linkages, 12 joints and a moving (tool) plate. There are six chains connecting the main column to the end-effector in the TAU configuration. The TAU robot is a typical 3/2/1 configuration, which configuration is shown in Figure 11 of Section 2. There are three parallel and identical links and another two parallel and identical links. Six chains will be used to derive all kinematic equations. Table 1 highlights the features of the TAU configuration.

On the subject of D-H modeling, Denavit J. & Hartenberg, H, 1955, Tasi, L. 1999, Raghavan, M. 1993, Abderrahim M. & Whittaker, A. R. 2000 have ap-

plied the method and studied the limitations of various modeling methods. On the subject of **forward kinematics**, focus has been on finding *closed form solutions* based on various robotic configurations, and *numerical solutions* for difficult configurations of robots. It can be found in the work done by Dhingra A. K. et al. 1998, 2000, Shi, X. & Fenton, R. G. 1994, Didrit, O. et al, 1998, Zhang, X. & Song, S. 1991, Nanua, P. et al, 1990, Sreenivasan, S. V. et al, 1994, Griffis, M. & Duffy, J. 1989, and Lin, W. et al, 1992. On the subject of error analysis, Wang, J. & Masory, O. 1993, Gong, C. et al, 2000, Patel, A. J. & Ehmann, F. E. 2000 used *forward kinematic solutions* to obtain errors. **Jacobian matrix** was also used in obtaining errors. On the subject of the variation of parallel configurations, based on the work done by Dhingra, A. K. et al, 1999, 2000, Geng, Z. & Haynes, L. S. 1994, the influence of the configurations on the methods of finding closed form solutions can be found.

In this paper, the D-H model (Figure 2) is used to define the TAU robot configuration, a complete set of parameters is included in the modeling process. Kinematic model and error model are established for including all types of errors using **Jacobian matrix method** for the TAU robot. Meanwhile, a very effective **Jacobian Approximation Method** is introduced to calculate the forward kinematic problem instead of the Newton-Raphson method. It denotes that a closed form solution can be obtained instead of a numerical solution. A full size Jacobian matrix is used in carrying out error analysis, error budget, and model parameter estimation and identification. Simulation results indicate that both Jacobian matrix and Jacobian Approximation Method are correct and have an accuracy of micrometers. ADAMS simulation results are used in verifying the established models.

A six-degrees-of-freedom precision measuring system is introduced in this study as an application of all methods mentioned above. The methods are also applied to explore new robotic applications such as grinding and machining. These new developments also revive the interest in robotic performance evaluation. Given the mechanical configurations of industrial robots with their popular six degrees of freedom, industrial robots have to be evaluated with metrology device or system of 3 or more degrees of freedom. Evaluation methods and equipment are needed to measure the spatial pose of robot efficiently with low cost.

Several methods are available for characterizing robot performance in accordance with ISO 9283 “Manipulating Industrial Robots Performance Criteria and Related Test Methods”. Eight major performance measuring methods and techniques are introduced in the technical report ISO TR 13309, including the accurate, easy-to-use but costly laser tracking technique. The pros and cons of existing multi-degrees of freedom measuring systems, including laser tracker, straight edges, multi-probes at certain check points, image and scanning techniques etc, are well documented [Lau and Hocken, 1984; Van Brussel, 1990; Ji-

ang et al, 1988]. Pose measurement of robotic end-effecter has been the focus [Ziegert and Datseries, 1990; Zhu and Cui, 2001, 2003].

Precision booster (Figure 3) a 6-DOF piezoelectric ultraprecision positioning drive is developed to provide industrial robots with 6-DOF fine positioning capability. It is designed to mount at the end of the forearm of a robot before its end-effector. With the added fine positioning capability, the accuracy of industrial robots can be greatly enhanced. Working with more accurate feedback sensors or calibration processes, the booster enables industrial robot to reach micrometer accuracy – one or two orders of magnitude higher than those of conventional serial robots. The accuracy of the precision booster can be designed in the range of sub-micrometer or micrometer over a range of millimeters enough to cover the sub-millimeter positioning resolution offered by existing industrial robots. The booster features monolithic flexure construction and the flexure structure functions as a spatial motion mechanism. This monolithic motion mechanism is backlash free and stick-slip free. High strength and high stiffness piezoelectric actuators are used to power the booster to perform fine positioning.

	TAU robot
Work area	1.5 m * 3 m
Repeatability	15 μm
Path accuracy	30 μm
Acceleration	5 g
Maximum positioning speed	180m / min
Excitation frequency	> 40 Hz
Cost	< 250 KUSD

Table 1. Specifications of the TAU Robot Based on Certain Applications

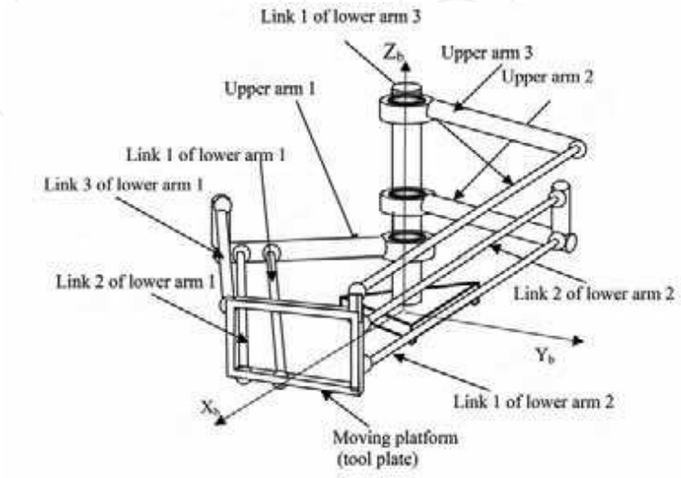


Figure 1. One of the TAU Robot Configurations

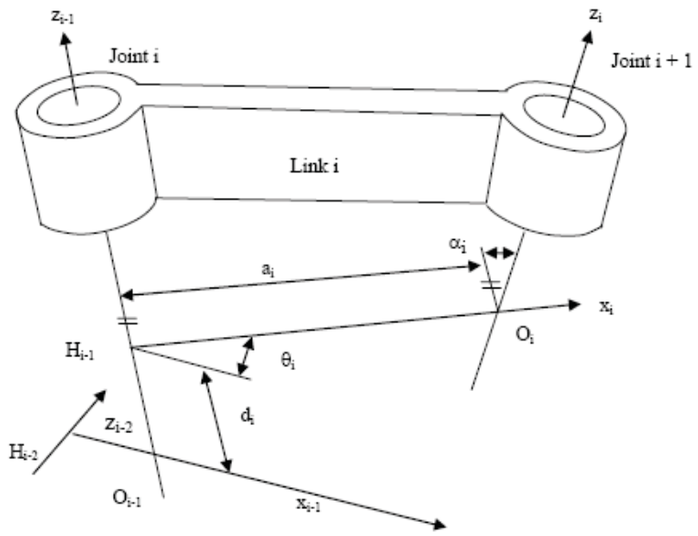


Figure 2. Definition of the Parameters in D-H Model

1.2 Kinematic Configurations of Parallel Robots

Gough-Stewart parallel robot, or so-called 'hexapod' shown in Figure 3 (Gough 1957 and Stewart 1965), is an assembly consisting of a fixed base with universal joints connecting the base to six linear-actuated limbs that support a moving platform through six ball-and-socket joints. This configuration allows the platform to move with six degrees-of-freedom employing the fewest number of actuators while maintaining stiffness by using only two-force-members. It is a closed-loop kinematic system with parallel links and is considered to be far more rigid than that of its serial counterparts of the same size and weight. Its force-output-to-manipulator-weight-ratio is generally an order of magnitude bigger than that of most industrial robots (Liu, 1993). *The same closed-loop kinematic configuration that gives its rigidity also complicates the solution of the forward kinematics in such a way that no closed-loop solution for this problem has been found* (Lacaze, Tasoluk and Meystel, 1997).

Tricept robot, shown in Figure 4, logically derived from the Tetrabot (Thornton, 1988), has a 3-DOF (degree of freedom) configuration of the parallel type to execute translational motions and a 3-DOF spherical wrist to execute rotational motions (Neumann and Neos Robotics, 1998). Its workspace is to be considered relatively large compared to the size of the robot. In order to further enlarge the size of the workspace, the addition of a revolute joint at the fixed base has been envisaged, introducing kinematic redundancy into the robotic manipulator. Its translational part can be thought as a reduced Stewart

platform with only three limbs. Like the Stewart platform, its kinematics has not been completely obtained: the inverse kinematics problem admits an analytical solution whereas the direct kinematics problem may require the use of iterative algorithms (Siciliano, 1999).

Delta robot, patented in U.S. in 1990 (Clavel, 1990), is shown in Figure 5. The basic idea behind the Delta parallel robotic design is the use of parallelograms. A parallelogram allows an output link to remain at a fixed orientation with respect to an input link. The use of three such parallelograms restrains completely the orientation of the mobile platform, which remains only three purely translational degrees of freedom. The input links of the three parallelograms are mounted on rotating levers via revolute joints. The revolute joints of the rotating levers are actuated in two different ways: with rotational (DC or AC servo) motors or with linear actuators. Finally, a fourth leg is used to transmit rotary motion from the base to an end-effector mounted on the mobile platform.
The use of base-mounted actuators and low-mass links allows the mobile platform to achieve accelerations of up to 50-G in experimental environment and 12 G in industrial applications. This makes the Delta robot a perfect candidate for pick and place operations of light objects. The Delta design has been applied to industry robot for several years. Its kinematics and dynamics also have been developed (Hunt 1973 and Codourey 1998).



Figure 3.Piezo Driven Flexure Based Hexapod (Zhu and Cui, 2001)

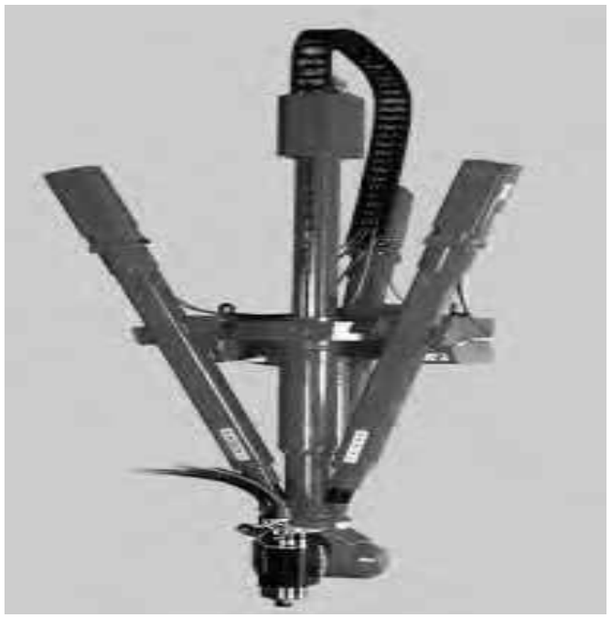


Figure 4. Tricept Robot (Neumann and Neos Robotics, 1998)

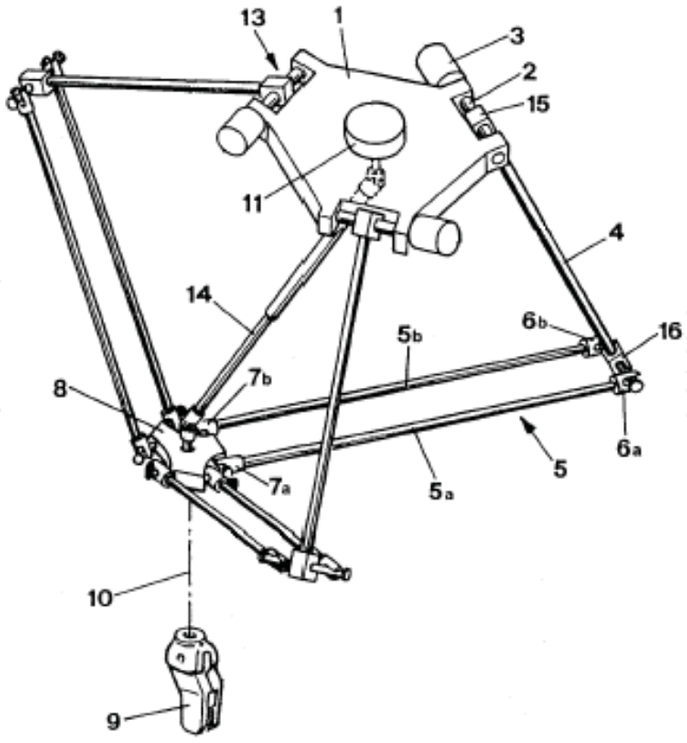


Figure 5. Delta Robot from US patent No. 4,976,582

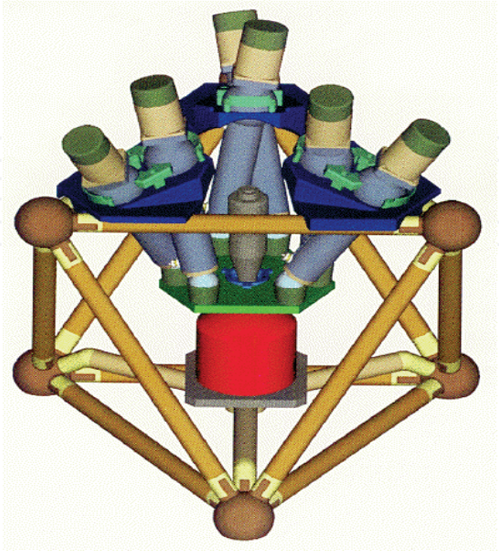


Figure 6. Octahedral Hexapod (NIST)

Octahedral Hexapod as shown in Fig. 1.6 is a demonstration machining center with six DOFs. It is a small, portable machine based on an octahedral framework. Machine motion is achieved by a Stewart Platform style actuation system. The framework and machining system can achieve high overall stiffness due to the fact that the structural members are generally in tension or compression with a minimum amount of bending stress. This structure allows the machine's capabilities to be independent of its foundation. Six identical struts with spherical pivots are mounted to the framework to drive the machining spindle, providing six-axis machining capability. The machine has a work volume of approximately 5" diameter X 3.5" high. The assembled machine will fit in a 24" X 24" X 25" volume. The machine completely disassembles and stores in a case approximately 24" X 16" X 10".

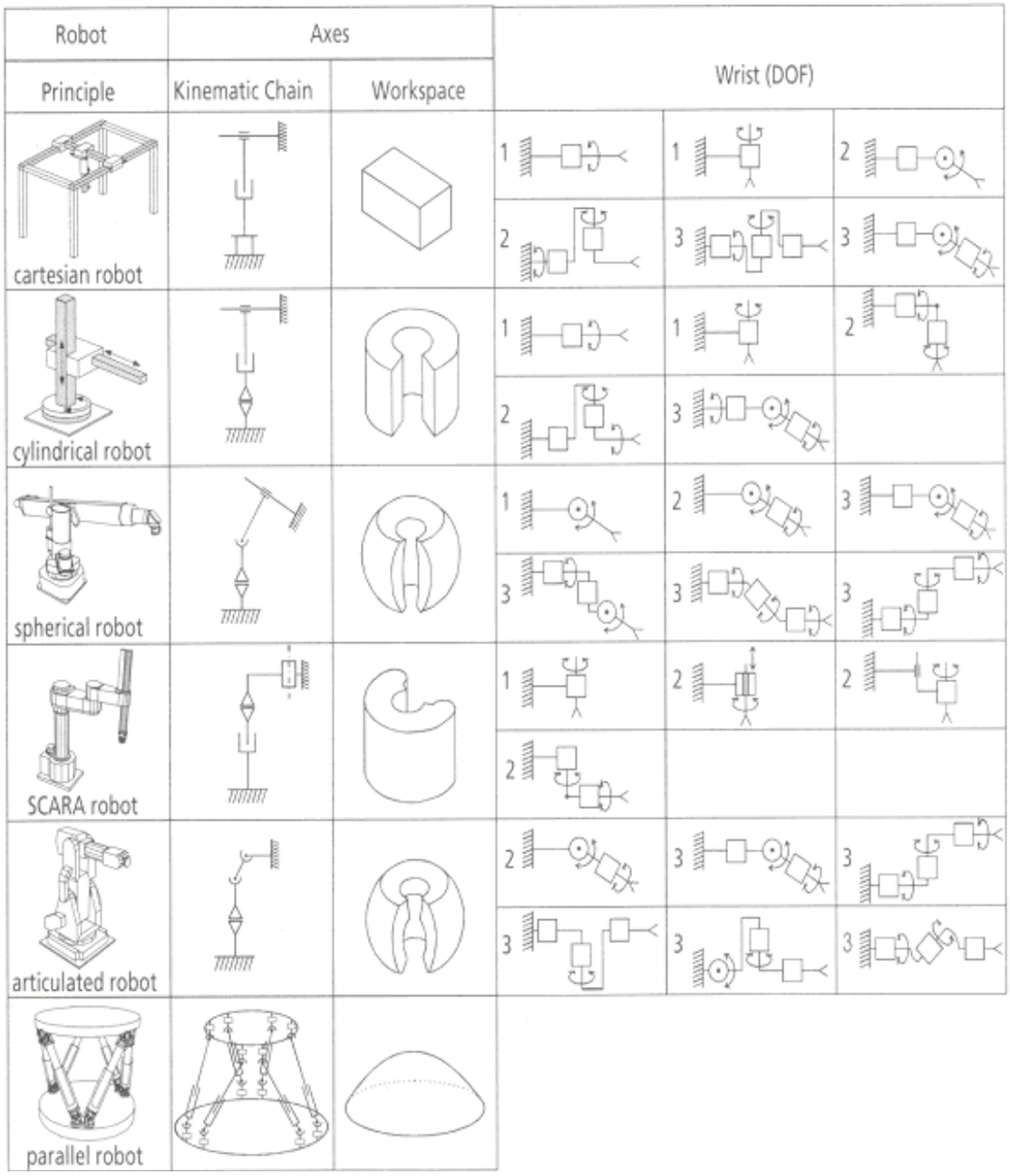


Figure 7 Typical Arms and Wrist Configuration of Industrial Robots (Handbook of Industrial Robotics)

2. Tau Configuration Design

Hybrid manipulators are parallel-serial connection robots that give rise to a multitude of highly articulate robotic manipulators. The robotic manipulators have a strength-to-weight ratio many times larger than the value currently available with industrial or research manipulators. This is due to the fact that these hybrid manipulators are stress compensated and ultralight in weight, yet are extremely stiff due to the fact that the force distribution in their structures is mostly axial.
Serially connected robot manipulators in the form of an open-loop kinematic chain with computer-controlled joint actuation have been utilized extensively in robot industry. For parallel manipulators, a classic example is the Stewart platform, which has been kinematically and, to some extent, dynamically investigated by many researchers.
The major advantages of existing parallel robots, compared with serial robots, are smaller mass and higher stiffness of the arm system. This is very important to achieve a shorter cycle time with lower actuator power together with more accurate movement.
The disadvantage is a relative small workspace in relation to the volume of the arm system. In the process of improving the robotic performance by diminishing the disadvantages, the basic features in design should include the following:

1. All the actuators are mounted on a fixed platform, which minimizes the mass of the moving arm system.
2. The links connected to the actuated platform are two-force members transmitting only compression and tensile forces and do not carry bending and twisting loads, which makes it easy to achieve a moving arm system of high in stiffness and low in mass.
3. The joints can be implemented as ball and socket bearings, which makes it possible to obtain high precision in addition to high stiffness and low mass for the joint arrangement.
4. The actuated platform is positioned with 3 translational DOFs in a parallel fashion without angular displacement.

2.1 The Link Clustering Design Approach

Systematic clustering of the links connected to the actuated robot platform has been studied. Based on this design approach new parallel arm structures have been identified and some new robot concepts have been found (Brogardh, T, 2000).
Figure 8 shows schematically the basic components needed to achieve the

Delta parallel arm robot with the kinematic features listed above. The actuated platform is connected to 6 links of type A by means of ball and socket joints that each has 3 DOFs. Type A means that the links are designed to be stiff only for forces along their axial direction in the structure. This force loading characteristic in the links of type A is guaranteed since a ball and socket joint cannot transmit bending moment or twisting torque to the link it is connected to.
The actuators in Figure 8 are mounted on the fixed platform and the moving part of the actuators is connected to the links of type A via links of type B. The type B links are designed to be stiff against also bending moment and twisting torque. All the links of type B do not need to be connected to actuators, but 3 of them must, otherwise the actuated platform cannot be manipulated in 3 DOFs.

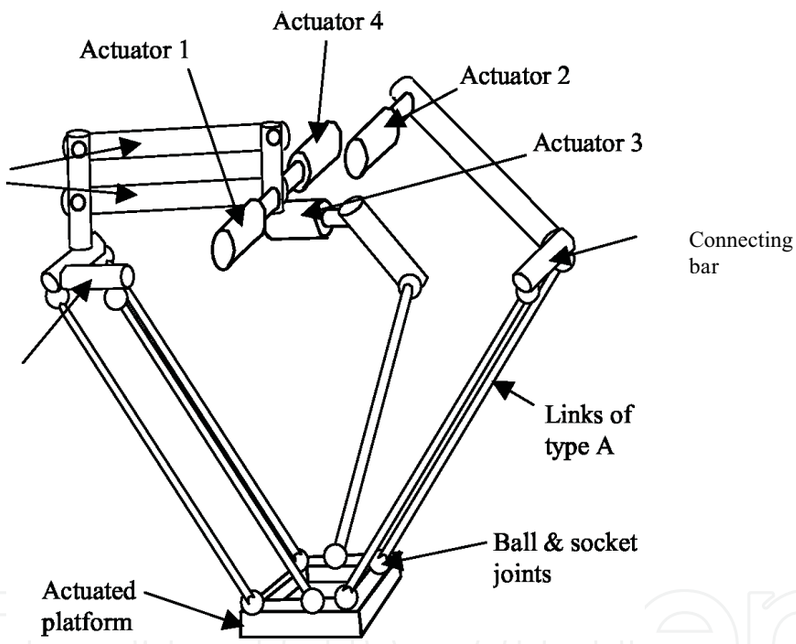


Figure 8. Components for the Design of Structures with the Same Features as the Delta Robot. (Courtesy of Brogardh, T, 2000).

Each of the links of type B (Figure 8) can be connected to one or more of the links of type A. One could say that each link of type B can be connected to a cluster of links of type A and it is possible to introduce a simple clustering scheme, where for example 2/2/2 means that the links of type A are clustered with 2 links to each of the 3 links of type B. To achieve parallel movements of the actuated platform (to preserve the tilt angles), type A links belonging to the same cluster must be parallel and have the same length. Moreover, to avoid a

collapsing parallel arm structure because of kinematic singularities, the placement of the type A link joints on the actuated platform must be optimized as well as the relative directions between the type A links of the different clusters. The 6 links of type A can be clustered in 3 ways: 2/2/2, 3/2/1 and 4/1/1. The 4/1/1 clustering will not fulfill the kinematic demands for a controllable structure and can be omitted. However, the 2/2/2 and the 3/2/1 clustering according to Figure 10 are kinematically useful. Using the 2/2/2 clustering scheme for the design will end up with the Delta structure. The optimized link placement in this case is achieved when the lines between the joints of each cluster on the actuated platform have an angle of 120 degrees between each other. The arm structure will collapse if the angle between two joint lines is 0 (180) degrees instead of 120 degrees. If instead the 3/2/1 clustering is used for the design of a parallel arm robot, the placement of the joints of the type A links on the platform surface is not critical. The only demand is that the 3 lower joints of cluster 1 are not allowed to be on a straight line on the platform. The optimum is achieved when these 3 joints of cluster 1 form a triangle with equal side length. This robustness with respect to the link placement on the actuated platform opens up new possibilities for the robot design.

In Figure 9 the actuated platform is considered to have a plane design, which means that the links of type A connect to the platform surface in a plane. However, the actuated platform could also be designed as a 3-D framework as depicted in Figure 10. This framework does not need to be a cube as in the figure, but the cube drawing makes it easier to see the configurations of the links. As in the case with a plane platform design, there are also in this case 2 useful clustering possibilities for an actuated 3-D platform: 2/2/2 and 3/2/1.

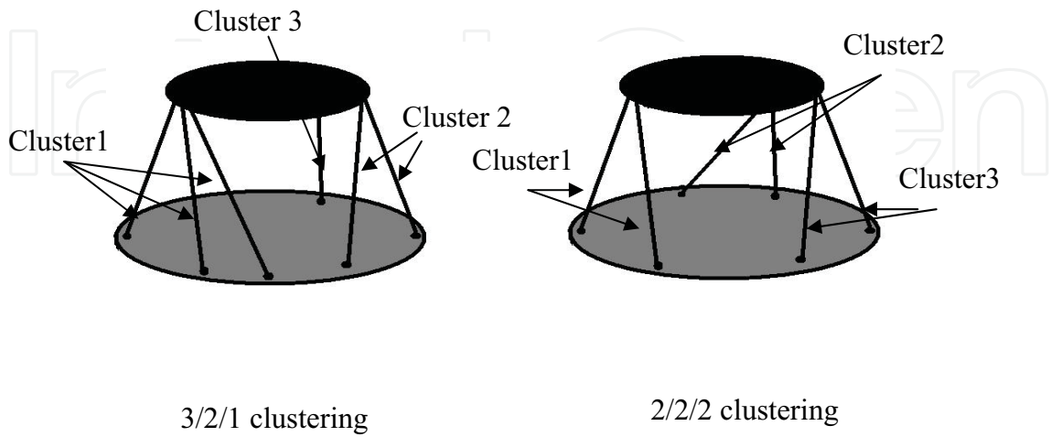


Figure 9. Useful Clustering Strategies When the Links of Type A Are Attached to the Actuator Platform in a 2-D Pattern.

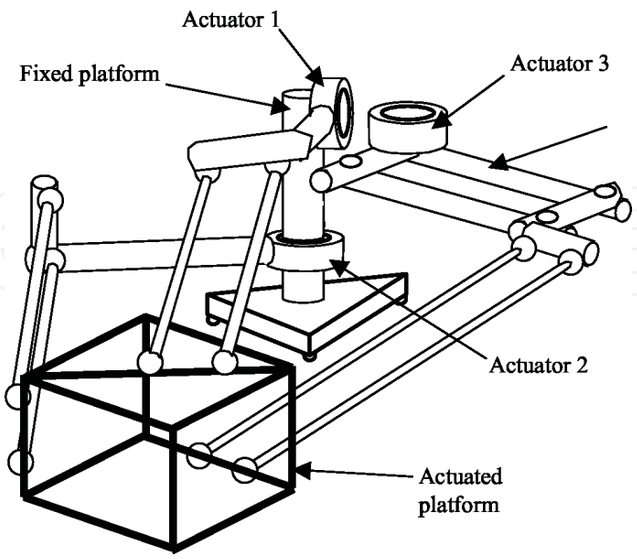


Figure 10. Useful 2/2/2 Clustering Strategies when the Links of Type A are Attached to the Actuator Platform in a 3-D pattern. (Courtesy of Brogardh, T, 2000).

2.2 TAU 3/2/1 Configuration

A new class of parallel robot, namely, TAU robot, has been created based on the 3/2/1 configuration. It combines the performance advantages of parallel arm mechanism (e.g., high stiffness, high accuracy) with the large workspace of serial robot.

As shown in Figure 11, the primary design of the TAU prototype robot has three actuators mounted on the base fixture and arranged in a line, which is called an I-configuration TAU. From bottom to top, actuators and upper arms (type B link) are numbered as 1, 2 and 3, and connected with 3, 2 and 1 lower arm(s) (type A link) respectively. This configuration basically performs a 3-DOF motion in its workspace. The 3-DOF parallel robot has a small footprint but with an enhanced stiffness.

The six links (lower arms) connected to the tool plate are driven by the three upper arms rotating around Z-axis. This structure has 3 DOFs in its work-space. With its geometric constraint, the DOF of a TAU robot is equal to (Tsai, 1999)

$$\text{DOF} = \lambda(n - j - 1) + \sum_{i=1} f_i$$

λ: degree of freedom of the space in which a mechanism is intended to function

n: number of links in a mechanism, including the fixed link
j: number of joints in a mechanism, assuming that all the joints are binary.
f_i: degree of relative motion permitted by joint i.

Joints between fixture and upper arms are 1-DOF rotational joints. Joints connecting upper and lower arms are 2-DOF universal joints. 3-DOF spherica-joints connect lower arms and moving plate.

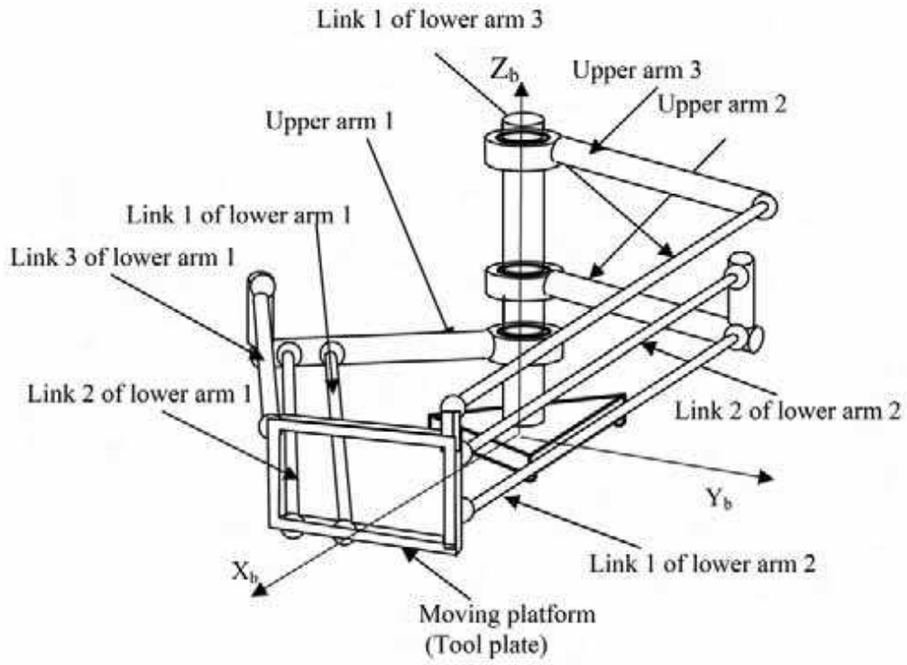


Figure 10.TAU Robot Based on Clustering Design Approach

2.3 Features of the TAU Configuration

The parallel robotic configuration for translational motion has a higher stiffness compared to the serial robotic configuration. It also has the following features: Large workspace, 360 degree around its base axis like a serial robot, analytic kinematic solution and analytic rigid-body dynamic solution.

Applications and Design Requirements

With these new features, the robot has the possibility to work with several conveyors and feeders placed around the robot. This is just one example of how a SCARA like parallel arm robot could be used to increase the productivity in an existing production line just by replacing conventional SCARA robots used today with its parallel arm cousins.

Typical Applications

Spot welding and painting are among the earliest application for industrial robot. Their payload is usually less than 50 kg. Repeatability requirement is in the range of 100 μm.

Pick and place and packaging have relatively low requirement on repeatability and stiffness. Payload varies from 1 to 500 kg. High speed is preferred for high productivity.

Machining or material removal including deburring, grinding, milling and sawing, requires high stiffness. Stiffness and accuracy of the robot decide the quality of the machined product.

Potential Applications

Laser cutting or welding, as a non-contact process requires an accuracy/repeatability less than 100 μm. Payload, which is the laser gun and accessories, is usually less than 50 kg. Speed required is not high in such applications.

Coordinate measuring function is typically performed by a CMM. It has a strict accuracy requirement of less than 50 μm for both static and path following at a low speed.

Fine material removal is as precision machining application now dominated by CNC machines. It requires the highest stiffness and system accuracy.

Design Objectives

The mechanism design is application orientated. Three typical future applications were selected and studied in the design phase: 2-D laser cutting, CMM for automobile vehicle and material removal applications. Each of them represents a typical application with certain requirements. Accuracy is a dominating factor reflecting the level of performance of any measurement system. The accuracy is low for current articulated robot arms. Material removal application requires high stiffness. Current serial configured CNC machines or parallel-configured machines have a limited workspace.

	TAU robot	Linear motor gantry
Work area	1.5 m * 3 m	1.5 m * 3 m
Repeatability	15 μm	17 μm
Path accuracy	30 μm	50 μm
Acceleration	5 g	2 - 4 g
Maximum positioning speed	180 m / min	100 - 200 m / min
Excitation frequency	> 40 Hz	13 Hz
Cost	< 250 KUSD	250 KUSD

Table 2. Performance Comparison with 2-D Laser Cutting Gantry Robot

Table 2 shows the performance comparison between the TAU robot and the gantry robot currently used in laser cutting application, which indicates the potential applications instead of using linear gantry robot. The performance of TAU covers all advantages of the Linear Motor Gantry.

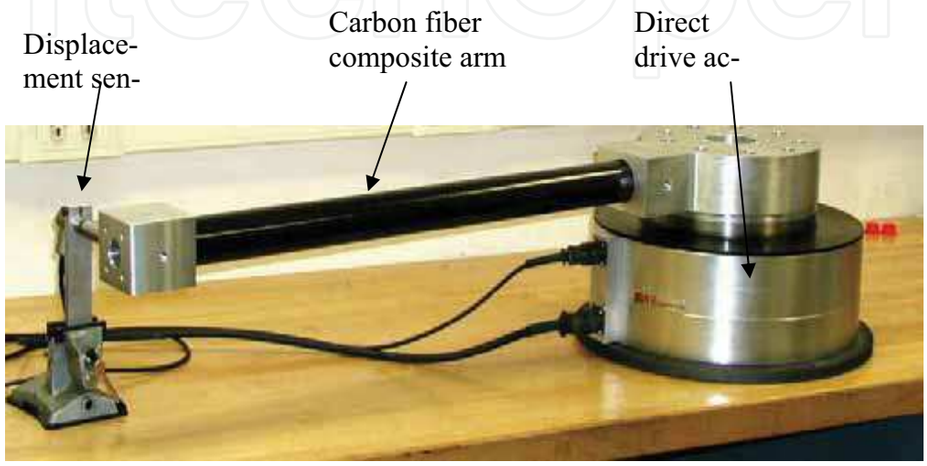


Figure 12. Single Arm Test Platform for Drive Motor Error Analysis

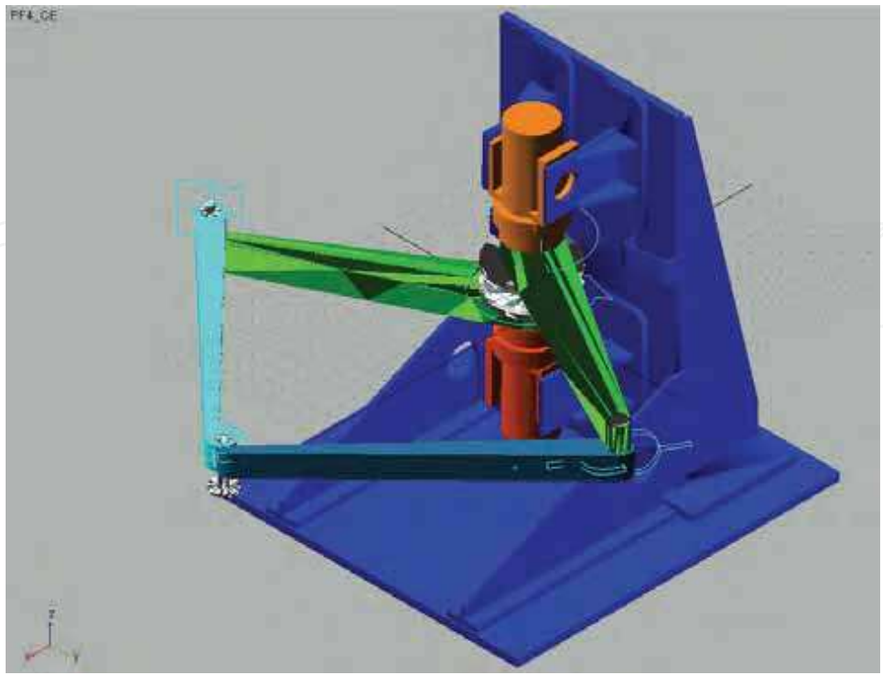


Figure 13. ADAMS Simulation Model for Two-Arm Test Platform



Figure 14. Two-Arm Test Platform (double SCARA structure)



Figure 15. TAU Prototype Design

3. Kinematics of Tau configuration

This chapter gives the nominal (no error) kinematics of the TAU robot. It is a general solution for this type of 3-DOF parallel-serial robots. For the two-arm test platform, a simple kinematic solution can be obtained based on its double SCARA configuration and it is not included in this chapter. The two-arm test platform kinematics was used in friction model identification and kinematic error calibration of the two-arm test platform. It will be introduced as needed in the related chapters.

To solve the kinematics of this 3-DOF TAU robot, three independent equations are needed. The three lower arm links, connected between the moving plate and upper arm 1, are designed to be parallel to each other and have the same length. Similarly, the two lower arms of upper arm 2 are also parallel and equal in length, which gives another length equation. The third equation comes from the lower arm 3.

Formulating these three equations all starts from point P in Figure 3.1, where three kinematic chains meet. Three basic equations for the kinematic problem are:

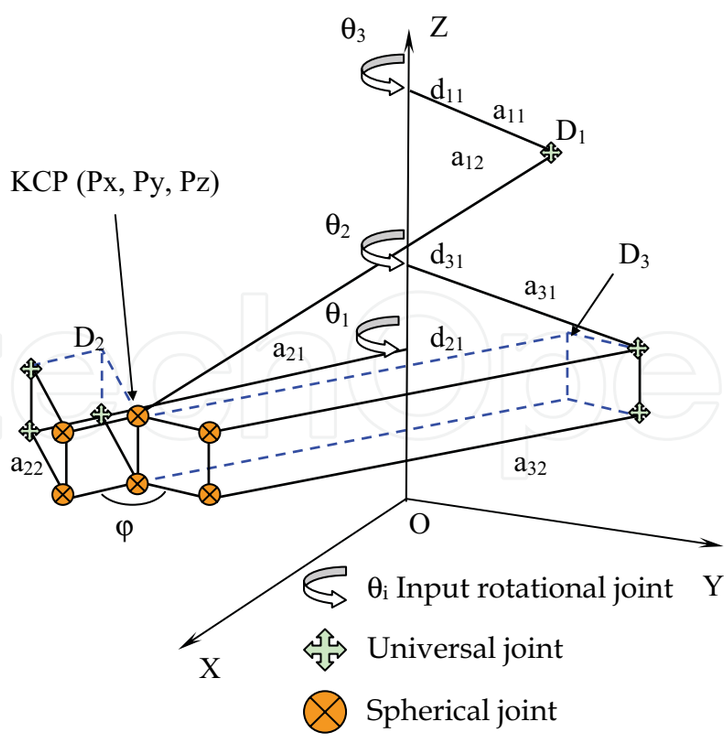


Figure 16.TAU Robot Kinematic Representation

For Point $D_{1:}$

$$D_{1x} = a_{11}\cos(\theta_3) - a_{33}\cos(120+\theta_1)$$
$$D_{1y} = a_{11}\sin(\theta_3) - a_{33}\sin(120+\theta_1)$$
$$D_{1z} = d_{11} - d4$$
$$dist(D_1 - P) = a_{12}$$

For Point $D_{2:}$

$$D_{2x} = a_{21}\cos(\theta_1)$$
$$D_{2y} = a_{21}\sin(\theta_1)$$
$$D_{2z} = d_{21} + d_{23}$$
$$dist(D_2 - P) = a_{22}$$

For Point $D_{3:}$

$$D_{3x} = a_{31}\cos(\theta_2) - a_{33}\cos(120+\theta_1)$$
$$D_{3y} = a_{31}\sin(\theta_2) - a_{33}\sin(120+\theta_1)$$
$$D_{3z} = d_{31}$$
$$dist(D_3 - P) = a_{32}$$

Basic equations

$$a_{12}^2 = (D_{1x} - Px)^2 + (D_{1y} - py)^2 + (D_{1z} - Pz)^2 \quad (1)$$
$$a_{22}^2 = (D_{2x} - Px)^2 + (D_{2y} - Py)^2 + (D_{2z} - Pz)^2 \quad (2)$$
$$a_{32}^2 = (D_{3x} - Px)^2 + (D_{3y} - Py)^2 + (D_{3z} - Pz)^2 \quad (3)$$

3.1 Inverse Kinematics

In an inverse kinematic problem, the Cartesian positioning information (Px, Py, Pz) is known. The unknowns are joint space position of active drive angles: θ_1, θ_2 and θ_3.

Substitute point D2 into Equation (2):

$$2a_{21}(Px\cos\theta_1 + Py\sin\theta_1) = a_{21}^2 - a_{22}^2 + Px^2 + Py^2 + (D_{2Z} - Pz)^2$$

Therefore, the first angle is obtained as:

$$\theta_1 = \cos^{-1}\frac{a_{21}^2 - a_{22}^2 + Px^2 + Py^2 + (D_{2Z} - Pz)^2}{2a_{21}\sqrt{Px^2 + Py^2}} + \tan^{-1}\frac{Py}{Px} \quad (4)$$

Substitute point D3 into Equation (3):

$$a_{32}^2 = (a_{31}\cos\theta_2 - a_{33}\cos(120+\theta_1) - Px)^2 + (a_{31}\sin\theta_2 - a_{33}\sin(120+\theta_1) - Py)^2 + (D_{3z} - Pz)^2$$

Therefore:

$$\theta_2 = \cos^{-1}\frac{a_{31}^2 - a_{32}^2 + C_{3x}{}^2 + C_{3y}{}^2 + (D_{3z} - C_{3z})^2}{2a_{31}\sqrt{C_{3x}{}^2 + C_{3y}{}^2}} + \tan^{-1}\frac{C_{3y}}{C_{3x}} \tag{5}$$

Where,

$$C_{3x} = Px + a_{33}\cos(120+\theta_1)$$
$$C_{3y} = Py + a_{33}\sin(120+\theta_1)$$
$$C_{3z} = Pz$$

Substitute point D1into Equation (1)

$$2a_{11}(C_{3x}\cos\theta_3 + C_{3y}\sin\theta_3) = a_{11}^2 - a_{12}^2 + C_{3x}{}^2 + C_{3y}{}^2 + (D_{1z} - C_{3z})^2$$

Therefore:

$$\theta_3 = \cos^{-1}\frac{a_{11}^2 - a_{12}^2 + C_{3x}{}^2 + C_{3y}{}^2 + (D_{1z} - C_{3z})^2}{2a_{11}\sqrt{C_{3x}{}^2 + C_{3y}{}^2}} + \tan^{-1}\frac{C_{3y}}{C_{3x}} \tag{6}$$

Equations (4), (5) and (6), therefore, are the inverse kinematics ended at point P on the moving platform. Noticed that point P is the kinematic calculation point, and additional inverse kinematic is needed to transfer TCP (Tool Center Point) to point P, when tool or wrist assembly is attached to the moving platform.

3.2 Forward Kinematics

The forward kinematic problem of a parallel configuration in general is more difficult than the inverse problem, for certain configurations there is no analytical solution admitted. For this TAU robot, the analytical forward kinematics is achievable. The Cartesian positioning information (Px, Py, Pz) is unknown in this case. The known are joint space position of active drive angles: θ_1, θ_2 and θ_3.

Change the format of Equations (1), (2) and (3) into:

$$D_{1x}^2 - 2D_{1x}Px + Px^2 + D_{1y}^2 - 2D_{1y}Py + Py^2 + D_{1z}^2 - 2D_{1z}Pz + Pz^2 = a_{12}^2 \quad (7)$$

$$D_{2x}^2 - 2D_{2x}Px + Px^2 + D_{2y}^2 - 2D_{2y}Py + Py^2 + D_{2z}^2 - 2D_{2z}Pz + Pz^2 = a_{22}^2 \quad (8)$$

$$D_{3x}^2 - 2D_{3x}Px + Px^2 + D_{3y}^2 - 2D_{3y}Py + Py^2 + D_{3z}^2 - 2D_{3z}Pz + Pz^2 = a_{32}^2 \quad (9)$$

Equation (7) – Equation (8)

$$(D_{1x}^2 + D_{1y}^2 + D_{1z}^2) - (D_{2x}^2 + D_{2y}^2 + D_{2z}^2) - 2(D_{1x} - D_{2x})Px - 2(D_{1y} - D_{2y})Py - 2(D_{1z} - D_{2z})Pz = a_{12}^2 - a_{22}^2 \quad (10)$$

Equation (7) – Equation (9)

$$(D_{1x}^2 + D_{1y}^2 + D_{1z}^2) - (D_{3x}^2 + D_{3y}^2 + D_{3z}^2) - 2(D_{1x} - D_{3x})Px - 2(D_{1y} - D_{3y})Py - 2(D_{1z} - D_{3z})Pz = a_{12}^2 - a_{32}^2 \quad (11)$$

Thus, define:

$$d_1 = D_{1x}^2 + D_{1y}^2 + D_{1z}^2$$
$$d_2 = D_{2x}^2 + D_{2y}^2 + D_{2z}^2$$
$$d_3 = D_{3x}^2 + D_{3y}^2 + D_{3z}^2$$

Equation (10) becomes

$$(D_{1x} - D_{2x})Px + (D_{1y} - D_{2y})Py + (D_{1z} - D_{2z})Pz = (a_{22}^2 - a_{12}^2 + d_1 - d_2)/2$$

Let

$$a_1 = D_{1x} - D_{2x}$$
$$b_1 = D_{1y} - D_{2y}$$
$$c_1 = D_{1z} - D_{2z}$$
$$e_1 = (a_{22}^2 - a_{12}^2 + d_1 - d_2)/2$$

Substitutes into Equation (10):

$$a_1 \cdot Px + b_1 \cdot Py + c_1 \cdot Pz = e_1$$

Similarly define

$$a_2 = D_{1x} - D_{3x}$$
$$b_2 = D_{1y} - D_{3y}$$
$$c_2 = D_{1z} - D_{3z}$$
$$e_2 = (a_{32}^2 - a_{12}^2 + d_1 - d_3)/2$$

Then Equation (11) as

$$a_2 \cdot Px + b_2 \cdot Py + c_2 \cdot Pz = e_2$$
$$a_1 \cdot Px + b_1 \cdot Py = e_1 - c_1 \cdot Pz$$
$$a_2 \cdot Px + b_2 \cdot Py = e_2 - c_2 \cdot Pz$$
$$\begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \end{bmatrix} \begin{bmatrix} Px \\ Py \end{bmatrix} = \begin{bmatrix} e_1 - c_1 \cdot Pz \\ e_2 - c_2 \cdot Pz \end{bmatrix}$$

Define $\Delta = a_1 b_2 - a_2 b_1$

For case 1, $\Delta \neq 0$:

$$\Delta x = (e_1 - c_1 \cdot Pz) b_2 - (e_2 - c_2 \cdot Pz) b_1$$
$$= (b_2 e_1 - b_1 e_2) + (b_1 c_2 - b_2 c_1) \cdot Pz$$
$$\Delta y = (e_2 - c_2 \cdot Pz) a_1 - (e_1 - c_1 \cdot Pz) a_2$$
$$= (a_1 e_2 - a_2 e_1) + (a_2 c_1 - a_1 c_2) \cdot Pz$$
$$Px = \frac{\Delta x}{\Delta} = \frac{b_2 e_1 - b_1 e_2}{\Delta} + \frac{b_1 c_2 - b_2 c_1}{\Delta} \cdot Pz$$
$$Py = \frac{\Delta y}{\Delta} = \frac{a_1 e_2 - a_2 e_1}{\Delta} + \frac{a_2 c_1 - a_1 c_2}{\Delta} \cdot Pz$$

Define:

$$f_1 = \frac{b_2 e_1 - b_1 e_2}{\Delta}$$
$$f_2 = \frac{a_1 e_2 - a_2 e_1}{\Delta}$$
$$f_x = \frac{b_1 c_2 - b_2 c_1}{\Delta}$$
$$f_y = \frac{a_2 c_1 - a_1 c_2}{\Delta}$$

Thus:

$$Px = f_1 + f_x Pz$$
$$Py = f_2 + f_y Pz \qquad (12)$$

Substitute Px, Py into Equation (3):

$$a_{32}^2 = (D_{3x} - f_1 - f_x Pz)^2 + (D_{3y} - f_2 - f_y Py)^2 + (D_{3z} - Pz)^2$$

Resort Equations above:

$$(1 + f_x^2 + f_y^2)Pz^2 + 2(f_{11}f_x + f_{22}f_y - D_{3z})Pz + (f_{11}^2 + f_{22}^2 + D_{3z}^2 - a_{32}^2) = 0$$

Where,

$$f_{11} = f_1 - D_{3x}$$
$$f_{22} = f_2 - D_{3y}$$

Then, let

$$A = 1 + f_x^2 + f_y^2$$
$$B = 2(f_{11}f_x + f_{22}f_y - D_{3z})$$
$$C = f_{11}^2 + f_{22}^2 + D_{3z}^2 - a_{32}^2$$

The solution of Equation $A \cdot Pz^2 + B \cdot Pz + C = 0$ is well known as:

$$Pz = \frac{-B \pm \sqrt{B^2 - 4 \cdot A \cdot C}}{2 \cdot A} \quad \text{..... (13)}$$

From Equation (12)

$$Px = f_1 + f_x Pz \quad \text{..... (14)}$$
$$Py = f_2 + f_y Pz \quad \text{..... (15)}$$

For case 1, Δ = 0, i.e. a2 = b2 = 0

$$\Delta = a_1 b_2 - a_2 b_1$$

In this case, $Pz = D_{3z}$ and only one Equation is available,

$$a_1 \cdot Px + b_1 \cdot Py = e_1 - c_1 \cdot Pz$$

i.e.

$$Py = (e_1 - c_1 \cdot Pz - a_1 \cdot Px) / b_1$$
$$= f_y Px + f_1 Pz + f_2$$

Where:

$$f_1 = -c_1 / b_1$$
$$f_2 = +e_1 / b_1$$
$$f_x = -a_1 / b_1$$

Substitute Px, Py into Equation (3) and resort the equation above:

$$(1 + f_x^2)Px^2 + 2(f_{11}f_x - D_{3x})Px + (f_{11}^2 + D_{3x}^2 - a_{32}^2) = 0$$

Where, $f_{11} = f_1 Pz + f_2 - D_{3y}$
Then, let

$$A = 1 + f_x^2$$
$$B = 2(f_{11}f_x - D_{3x})$$
$$C = f_{11}^2 + D_{3x}^2 - a_{32}^2$$

The solution of Equation $A \cdot Px^2 + B \cdot Px + C = 0$ is well known as:

$$Px = \frac{-B \pm \sqrt{B^2 - 4 \cdot A \cdot C}}{2 \cdot A}$$

From Equation (12)

$$Py = f_x Px + f_1 Pz + f_2$$
$$Pz = D_{3z}$$

For case b1, b2 =0
Equation $a_1 \cdot Px + b_1 \cdot Py = e_1 - c_1 \cdot Pz$ becomes

$$Px = (e_1 - c_1 \cdot Pz) / a_1$$

Py can be solved by one of those basic Equations, for example for Equation (1).

$$Py = D_{1y} \pm \sqrt{a_{12}^2 - (D_{1x} - Px)^2 - (D_{1z} - Pz)^2}$$

The sign is the same as d1y.

For case a1, a2 =0

Equation becomes

$$a_1 \cdot Px + b_1 \cdot Py = e_1 - c_1 \cdot Pz$$
$$Py = (e_1 - c_1 \cdot Pz) / b_1$$

Px can be solved by one of those basic Equations as

$$Px = D_{1x} \pm \sqrt{a_{12}^2 - (D_{1y} - Py)^2 - (D_{1z} - Pz)^2}$$

The sign is the same as D_{1x}.

Thus forward kinematic is solved based on geometry constrains. Like the inverse kinematics, additional mathematic work is needed for the kinematic chain from point P to final TCP depending on the configuration details of the tool or wrist.

3.3 Discriminant Analysis of Kinematic Solution

Mathematically neither forward nor inverse kinematics gives single solution. Forward kinematics usually has two solutions, because the passive joint angles formed between upper arm and lower arm are not determined by kinematic equations. When only arm 1 and arm 2 chains are considered, upper arm 1, lower arm1, upper arm 2 and lower arm 2 form a quadrilateral geometry. These two solutions form one convex and one concave quadrilateral and one and only one of them is allowed by mechanical constrains. The discriminating condition is the angle between arm 1 and arm 2. For inverse kinematics, the mathematic equations can give out up to 8 solutions for the same position input. Still the physical constrains limits the left arm can be only placed on the left side of right arm, together with the convex and concave condition, there is only one solution is reasonable for arm 1 and arm 2. However including arm 3 into consideration, if it can rotate freely around its axis, there are two solutions for the drive angle of arm 3 except for singularity point. However since the elbow joint of arm 3 physically limits arm 3 so that arm 3 can only move within one side. Therefore, combining mathematics and physical constrains together, within the reachable workspace, TAU robot kinematics gives single solution on each input for both forward and inverse routine.

4. Error Modeling and Jacobian Matrix with all variables

The purpose of error analysis is to minimize the error of robot system through assembly based on the comprehensive system error model. The reason is based

on the fact that all error source will either have a negative or positive influence on the system error, which is then possible to arrange them in a way that cancellation or at least error reduction will happen. The methodology is described as:

- Identifying the error effect of individual component using the established system error model.
- Identifying the dimensional ranges allowed in an assembly for each connection.
- Using the system error model to identify the negative or positive direction that a connection should be made within the ranges allowed.
- Predicting and minimizing system error using the model.
- Using proper error budget approach to minimize the system error.

4.1 Error Modeling

The assembly process is a process of error identification and more importantly, a process of error assignment in the way towards minimizing system error. During the process, error budget is completed, and more importantly, an accurate kinematic model should be established. The process is geared directly towards error control and compensation when a robot is in service. The process is also a redesign process for improved performance
Next attentions should be paid to:

- The direction and degree of influence of an error source on system error varies in the whole workspace.
- Random errors can not be dealt effectively.
- Effective fixture and measuring are important.
- The methodology reduces robotic system error and opens the door for more accurate error compensation.

For the TAU-robot, an important thing needed is the error analysis. One needs to assign an error limit or range to all components in order to obtain a given robotic system accuracy. The procedure is so called Error Budget.
Before the error budget, an important thing to accomplish is to establish and analyze the Jacobian Matrix. It is necessary to know Jacobian Matrix for all components before assigning error to all components. On the other hand one can also obtain the final accuracy with knowing Jacobian Matrix. Besides one can know which components are more important than others based on the Jacobian Matrix. Table 3 lists all the design variables for the TAU robot.

NO.	DESCRIPTION	MAME
1	drive 1	Joint 1
2	drive 2	Joint 2
3	drive 3	Joint 3
17	joint 1 and arm 1	a1
24		d1
4		sit1
10		afa1
18	joint_link11_arm 1	a2
19	short arm 1	a3
25		d3
5		sit3
11		afa3
20	joint 2 and arm 2	a4
26		d4
6		sit4
12		afa4
21	short arm 2	a5
27		d5
7		sit5
13		afa5
22	joint 3	a6
28		d6
8		sit6
14		afa6
23	arm 3	a7
29		d7
9		sit7
15		afa7
30	joint_link11_arm 1	x1
31		y1
32		z1
33	joint_link21_arm 1	x2
34		y2
35		z2
36	joint_link31_arm 1	x3
37		y3
38		z3
39	joint_link12_arm 2	x4
40		y4
41		z4

NO.	DESCRIPTION	MAME
42	joint_link22_arm2	x5
43		y5
44		z5
45	joint_link13_arm3	x6
46		y6
47		z6
48	joint_link11_platform	x11
49		y11
50		z11
51	joint_link31_platform	x22
52		y22
53		z22
54	joint_link21_platform	x33
55		y33
56		z33
57	joint_link12_platform	x44
58		y44
59		z44
60	joint_link22_platform	x55
61		y55
62		z55
63	joint-link13_platform	x66
64		y66
65		z66
16	height of the TCP	a
66	link 13	L0
67	link 11	L1
68	link 31	L2
69	link 21	L3
70	link 22	L4
71	link 12	L5

Table 3. Design Variables of TAU Robot

There are six kinematic chains from the base to the end-effector as:

Transfer Matrix: M1 Base->Joint1->Joint_link11_arm1
Transfer Matrix: M1*M3 Base->Joint1->Joint_link21_arm1
Transfer Matrix: M2 Base->Joint1->Joint_link31_arm1
Transfer Matrix: M4*M5 Base->Joint2->Joint_link12_arm2
Transfer Matrix: M4 Base->Joint2->Joint_link22_arm2
Transfer Matrix: M6*M7 Base->Joint3->Joint->Joint13_link_arm3

Where,

$$M_1 \rightarrow \begin{bmatrix} \cos(joint_1 + \Delta\theta_1) & -\sin(joint_1 + \Delta\theta_1)\cdot\cos(\Delta\alpha_1) & \sin(joint_1 + \Delta\theta_1)\cdot\sin(\Delta\alpha_1) & (700 + \Delta a_1)\cdot\cos(joint_1 + \Delta\theta_1) \\ \sin(joint_1 + \Delta\theta_1) & \cos(joint_1 + \Delta\theta_1)\cdot\cos(\Delta\alpha_1) & -\cos(joint_1 + \Delta\theta_1)\cdot\sin(\Delta\alpha_1) & (700 + \Delta a_1)\cdot\sin(joint_1 + \Delta\theta_1) \\ 0 & \sin(\Delta\alpha_1) & \cos(\Delta\alpha_1) & 750 + \Delta d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$M_2 \rightarrow \begin{bmatrix} \cos(joint_1 + \Delta\theta_2) & -\sin(joint_1 + \Delta\theta_2)\cdot\cos(\Delta\alpha_2) & \sin(joint_1 + \Delta\theta_2)\cdot\sin(\Delta\alpha_2) & (900 + \Delta a_2)\cdot\cos(joint_1 + \Delta\theta_2) \\ \sin(joint_1 + \Delta\theta_2) & \cos(joint_1 + \Delta\theta_2)\cdot\cos(\Delta\alpha_2) & -\cos(joint_1 + \Delta\theta_2)\cdot\sin(\Delta\alpha_2) & (900 + \Delta a_2)\cdot\sin(joint_1 + \Delta\theta_2) \\ 0 & \sin(\Delta\alpha_2) & \cos(\Delta\alpha_2) & 750 + \Delta d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$M_3 \rightarrow \begin{pmatrix} \cos(\Delta\theta_3) & -\sin(\Delta\theta_3)\cdot\cos(\Delta\alpha_3) & \sin(\Delta\theta_3)\cdot\sin(\Delta\alpha_3) & \Delta a_3\cdot\cos(\Delta\theta_3) \\ \sin(\Delta\theta_3) & \cos(\Delta\theta_3)\cdot\cos(\Delta\alpha_3) & -\cos(\Delta\theta_3)\cdot\sin(\Delta\alpha_3) & \Delta a_3\cdot\sin(\Delta\theta_3) \\ 0 & \sin(\Delta\alpha_3) & \cos(\Delta\alpha_3) & 200 + \Delta d_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$M_4 \rightarrow \begin{bmatrix} \cos(joint_2 + \Delta\theta_4) & -\sin(joint_2 + \Delta\theta_4)\cdot\cos(\Delta\alpha_4) & \sin(joint_2 + \Delta\theta_4)\cdot\sin(\Delta\alpha_4) & (900 + \Delta a_4)\cdot\cos(joint_2 + \Delta\theta_4) \\ \sin(joint_2 + \Delta\theta_4) & \cos(joint_2 + \Delta\theta_4)\cdot\cos(\Delta\alpha_4) & -\cos(joint_2 + \Delta\theta_4)\cdot\sin(\Delta\alpha_4) & (900 + \Delta a_4)\cdot\sin(joint_2 + \Delta\theta_4) \\ 0 & \sin(\Delta\alpha_4) & \cos(\Delta\alpha_4) & 950 + \Delta d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$M_5 \rightarrow \begin{pmatrix} \cos(\Delta\theta_5) & -\sin(\Delta\theta_5)\cdot\cos(\Delta\alpha_5) & \sin(\Delta\theta_5)\cdot\sin(\Delta\alpha_5) & \Delta a_5\cdot\cos(\Delta\theta_5) \\ \sin(\Delta\theta_5) & \cos(\Delta\theta_5)\cdot\cos(\Delta\alpha_5) & -\cos(\Delta\theta_5)\cdot\sin(\Delta\alpha_5) & \Delta a_5\cdot\sin(\Delta\theta_5) \\ 0 & \sin(\Delta\alpha_5) & \cos(\Delta\alpha_5) & -200 + \Delta d_5 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$M_6 \rightarrow \begin{pmatrix} \cos\left(\frac{1}{2}\cdot joint_1 + \frac{1}{2}\cdot joint_2 + \Delta\theta_6\right) & -\sin\left(\frac{1}{2}\cdot joint_1 + \frac{1}{2}\cdot joint_2 + \Delta\theta_6\right)\cdot\cos(-90 + \Delta\alpha_6) & \sin\left(\frac{1}{2}\cdot joint_1 + \frac{1}{2}\cdot joint_2 + \Delta\theta_6\right)\cdot\sin(-90 + \Delta\alpha_6) & \Delta a_6\cdot\cos\left(\frac{1}{2}\cdot joint_1 + \frac{1}{2}\cdot joint_2 + \Delta\theta_6\right) \\ \sin\left(\frac{1}{2}\cdot joint_1 + \frac{1}{2}\cdot joint_2 + \Delta\theta_6\right) & \cos\left(\frac{1}{2}\cdot joint_1 + \frac{1}{2}\cdot joint_2 + \Delta\theta_6\right)\cdot\cos(-90 + \Delta\alpha_6) & -\cos\left(\frac{1}{2}\cdot joint_1 + \frac{1}{2}\cdot joint_2 + \Delta\theta_6\right)\cdot\sin(-90 + \Delta\alpha_6) & \Delta a_6\cdot\sin\left(\frac{1}{2}\cdot joint_1 + \frac{1}{2}\cdot joint_2 + \Delta\theta_6\right) \\ 0 & \sin(-90 + \Delta\alpha_6) & \cos(-90 + \Delta\alpha_6) & 1700 + \Delta d_6 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$M_7 \rightarrow \begin{bmatrix} \cos(joint_3 + \Delta\theta_7) & -\sin(joint_3 + \Delta\theta_7)\cdot\cos(\Delta\alpha_7) & \sin(joint_3 + \Delta\theta_7)\cdot\sin(\Delta\alpha_7) & (900 + \Delta a_7)\cdot\cos(joint_3 + \Delta\theta_7) \\ \sin(joint_3 + \Delta\theta_7) & \cos(joint_3 + \Delta\theta_7)\cdot\cos(\Delta\alpha_7) & -\cos(joint_3 + \Delta\theta_7)\cdot\sin(\Delta\alpha_7) & (900 + \Delta a_7)\cdot\sin(joint_3 + \Delta\theta_7) \\ 0 & \sin(\Delta\alpha_7) & \cos(\Delta\alpha_7) & \Delta d_7 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

So, the six length equations can be obtained from matrices above.

4.2 Jacobian Matrix of TAU Robot with All Error Parameters

In error analysis, error sensitivity is represented by the Jacobian matrix. Derivations of the Jacobian matrix can be carried out after all the D-H models are established. For the TAU robot, the 3-DOF kinematic problem will become a 6-DOF kinematic problem. The kinematic problem becomes more complicated. In fact, the error sensitivity is formulated through $\frac{\partial x}{\partial g_i}$, $\frac{\partial y}{\partial g_i}$, $\frac{\partial z}{\partial g_i}$ where x, y, z represent the position of the tool plate and dg_i is the error source for each component. So the following equations can be obtained:

$$dx = \sum_{1}^{N} \frac{\partial x}{\partial l_i} dg_i \tag{16}$$

$$dy = \sum_{1}^{N} \frac{\partial y}{\partial l_i} dg_i \tag{17}$$

$$dz = \sum_{1}^{N} \frac{\partial z}{\partial l_i} dg_i \tag{18}$$

The error model is actually a 6-DOF model since all error sources have been considered. It includes both the position variables X, Y, Z and also rotational angles α, β, γ. From the six kinematic chains, the equations established based on D-H models are

$$f_1 = f_1(x, y, z, \alpha, \beta, \gamma, g) = 0$$
$$f_2 = f_2(x, y, z, \alpha, \beta, \gamma, g) = 0$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$f_6 = f_6(x, y, z, \alpha, \beta, \gamma, g) = 0$$

Differentiating all the equations against all the variables $x, y, z, \alpha, \beta, \gamma$ and g, where g is a vector including all geometric design parameters:

$$\frac{\partial f_i}{\partial x} \cdot dx + \frac{\partial f_i}{\partial y} \cdot dy + \frac{\partial f_i}{\partial z} \cdot dz + \frac{\partial f_i}{\partial \alpha} \cdot d\alpha + \frac{\partial f_i}{\partial \beta} \cdot d\beta + \frac{\partial f_i}{\partial \gamma} \cdot d\gamma + \sum_j \frac{\partial f_i}{\partial g_j} \cdot dg_j = 0 \tag{19}$$

Rewrite it in matrix as

$$\begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial z} & \frac{\partial f_1}{\partial \alpha} & \frac{\partial f_1}{\partial \beta} & \frac{\partial f_1}{\partial \gamma} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} & \frac{\partial f_2}{\partial z} & \frac{\partial f_2}{\partial \alpha} & \frac{\partial f_2}{\partial \beta} & \frac{\partial f_2}{\partial \gamma} \\ \frac{\partial f_3}{\partial x} & \frac{\partial f_3}{\partial y} & \frac{\partial f_3}{\partial z} & \frac{\partial f_3}{\partial \alpha} & \frac{\partial f_3}{\partial \beta} & \frac{\partial f_3}{\partial \gamma} \\ \frac{\partial f_4}{\partial x} & \frac{\partial f_4}{\partial y} & \frac{\partial f_4}{\partial z} & \frac{\partial f_4}{\partial \alpha} & \frac{\partial f_4}{\partial \beta} & \frac{\partial f_4}{\partial \gamma} \\ \frac{\partial f_5}{\partial x} & \frac{\partial f_5}{\partial y} & \frac{\partial f_5}{\partial z} & \frac{\partial f_5}{\partial \alpha} & \frac{\partial f_5}{\partial \beta} & \frac{\partial f_5}{\partial \gamma} \\ \frac{\partial f_6}{\partial x} & \frac{\partial f_6}{\partial y} & \frac{\partial f_6}{\partial z} & \frac{\partial f_6}{\partial \alpha} & \frac{\partial f_6}{\partial \beta} & \frac{\partial f_6}{\partial \gamma} \end{bmatrix} \bullet \begin{bmatrix} dx \\ dy \\ dz \\ d\alpha \\ d\beta \\ d\gamma \end{bmatrix} = \begin{bmatrix} \sum_j \frac{-\partial f_1}{\partial g_j} dg_j \\ \sum_j \frac{-\partial f_2}{\partial g_j} dg_j \\ \sum_j \frac{-\partial f_3}{\partial g_j} dg_j \\ \sum_j \frac{-\partial f_4}{\partial g_j} dg_j \\ \sum_j \frac{-\partial f_5}{\partial g_j} dg_j \\ \sum_j \frac{-\partial f_6}{\partial g_j} dg_j \end{bmatrix} \tag{20}$$

In a compact form, it becomes

$$J_1 dX = dG \tag{21}$$

Where

$$\text{dG} = \begin{bmatrix} \sum_j \frac{-\partial f_1}{\partial g_j} dg_j \\ \sum_j \frac{-\partial f_2}{\partial g_j} dg_j \\ \sum_j \frac{-\partial f_3}{\partial g_j} dg_j \\ \sum_j \frac{-\partial f_4}{\partial g_j} dg_j \\ \sum_j \frac{-\partial f_5}{\partial g_j} dg_j \\ \sum_j \frac{-\partial f_6}{\partial g_j} dg_j \end{bmatrix} = - \begin{bmatrix} \frac{\partial f_1}{\partial g_1} & \frac{\partial f_1}{\partial g_2} & \cdots & \frac{\partial f_1}{\partial g_N} \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \frac{\partial f_6}{\partial g_1} & \frac{\partial f_6}{\partial g_2} & \cdots & \frac{\partial f_6}{\partial g_N} \end{bmatrix}_{6\times N} \bullet \begin{bmatrix} dg_1 \\ dg_2 \\ \cdot \\ \cdot \\ \cdot \\ dg_N \end{bmatrix}_{N\times 1} \tag{22}$$

From Equation (22) above, we have,

$$dG = J_2 dg \tag{23}$$

Substitute Equation (21) into Equation (23) to obtain

$$J_1 dX = J_2 dg \tag{24}$$

$$dX = (J_1^{-1} J_2) dg \tag{25}$$

The Jacobian matrix is obtained as $J_1^{-1} \cdot J_2$

$$J = J_1^{-1} \cdot J_2 = \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial z} & \frac{\partial f_1}{\partial \alpha} & \frac{\partial f_1}{\partial \beta} & \frac{\partial f_1}{\partial \gamma} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} & \frac{\partial f_2}{\partial z} & \frac{\partial f_2}{\partial \alpha} & \frac{\partial f_2}{\partial \beta} & \frac{\partial f_2}{\partial \gamma} \\ \frac{\partial f_3}{\partial x} & \frac{\partial f_3}{\partial y} & \frac{\partial f_3}{\partial z} & \frac{\partial f_3}{\partial \alpha} & \frac{\partial f_3}{\partial \beta} & \frac{\partial f_3}{\partial \gamma} \\ \frac{\partial f_4}{\partial x} & \frac{\partial f_4}{\partial y} & \frac{\partial f_4}{\partial z} & \frac{\partial f_4}{\partial \alpha} & \frac{\partial f_4}{\partial \beta} & \frac{\partial f_4}{\partial \gamma} \\ \frac{\partial f_5}{\partial x} & \frac{\partial f_5}{\partial y} & \frac{\partial f_5}{\partial z} & \frac{\partial f_5}{\partial \alpha} & \frac{\partial f_5}{\partial \beta} & \frac{\partial f_5}{\partial \gamma} \\ \frac{\partial f_6}{\partial x} & \frac{\partial f_6}{\partial y} & \frac{\partial f_6}{\partial z} & \frac{\partial f_6}{\partial \alpha} & \frac{\partial f_6}{\partial \beta} & \frac{\partial f_6}{\partial \gamma} \end{bmatrix}^{-1} \bullet \begin{bmatrix} -\frac{\partial f_1}{\partial g_1} & -\frac{\partial f_1}{\partial g_2} & \cdots & -\frac{\partial f_1}{\partial g_N} \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ -\frac{\partial f_6}{\partial g_1} & -\frac{\partial f_6}{\partial g_2} & \cdots & -\frac{\partial f_6}{\partial g_N} \end{bmatrix} \tag{26}$$

For a prototype of the TAU robotic design, the dimension of the Jacobian matrix is 6 by 71. An analytical solution can be obtained and is used in error analysis.

4.3 Newton-Raphson Numerical Method

Because of the number of parameters involved as well as the number of error sources involved, the kinematic problem becomes very complicated. No analytical solution can be obtained but numerical solution. The TAU configuration, however, as a hybrid or a special case of parallel robots, its forward kinematic problem is, therefore, very complicated. The Newton-Raphson method as an effective numerical method can be applied to calculate the forward problem of the TAU robot, with an accurate Jacobian matrix obtained.
The Newton-Raphson method is represented by

$$X_{n+1} = X_n - [F'(X_n)]^{-1} \cdot F(X_n) \tag{27}$$

With the six chain equations obtained before, the following can be obtained

$$[F'(X_n)]^{-1} = \text{Inv} \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial z} & \frac{\partial f_1}{\partial \alpha} & \frac{\partial f_1}{\partial \beta} & \frac{\partial f_1}{\partial \gamma} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} & \frac{\partial f_2}{\partial z} & \frac{\partial f_2}{\partial \alpha} & \frac{\partial f_2}{\partial \beta} & \frac{\partial f_2}{\partial \gamma} \\ \frac{\partial f_3}{\partial x} & \frac{\partial f_3}{\partial y} & \frac{\partial f_3}{\partial z} & \frac{\partial f_3}{\partial \alpha} & \frac{\partial f_3}{\partial \beta} & \frac{\partial f_3}{\partial \gamma} \\ \frac{\partial f_4}{\partial x} & \frac{\partial f_4}{\partial y} & \frac{\partial f_4}{\partial z} & \frac{\partial f_4}{\partial \alpha} & \frac{\partial f_4}{\partial \beta} & \frac{\partial f_4}{\partial \gamma} \\ \frac{\partial f_5}{\partial x} & \frac{\partial f_5}{\partial y} & \frac{\partial f_5}{\partial z} & \frac{\partial f_5}{\partial \alpha} & \frac{\partial f_5}{\partial \beta} & \frac{\partial f_5}{\partial \gamma} \\ \frac{\partial f_6}{\partial x} & \frac{\partial f_6}{\partial y} & \frac{\partial f_6}{\partial z} & \frac{\partial f_6}{\partial \alpha} & \frac{\partial f_6}{\partial \beta} & \frac{\partial f_6}{\partial \gamma} \end{bmatrix} \tag{28}$$

This equation is used later to calculate the forward kinematic problem, and it is also compared with the method described in the next section.

4.4 Jacobian Approximation Method

A quick and efficient analytical solution is still necessary even though an accurate result has been obtained by the N-R method. The N-R result is produced based on iteration of numerical calculation, instead of from an analytical closed form solution. The N-R method is too slow in calculation to be used in on-line real time control. No certain solution is guaranteed in the N-R method. So the Jacobian approximation method is established. Using this method, error analysis, calibration, compensation, and on-line control model can be in turn established. As the TAU robot is based on a 3-DOF configuration, instead of a general Stewart platform, the Jacobian approximate modification can be obtained based the 3-DOF analytical solution without any errors. The mathematical description of the Jacobian approximation method can be described as follows.

For forward kinematics,

$$\begin{aligned} X &= F(\theta,\varepsilon) \\ X &= F(\theta,0)+J_{FORWARD}\cdot d\varepsilon \end{aligned} \tag{29}$$

Where $J_{FORWARD} = F^{'}(\theta,\varepsilon)$ and ε represents error. Thus, the analytical solution $F(\theta,0)$ and $F(X,0)$, is obtained. Therefore, the Jacobian Approximation as an analytical solution is obtained and is used to solve nonlinear equations instead of using N-R method.

4.5 Jacobian Matrix with a probe

A real tool should be attached on the wrist of robots as robots are used for any application. Here a probe means a real tool.
From the six kinematic chains, the equations established based on D-H models are

$$\begin{aligned} f_1 &= f_1(x,y,z,\alpha,\beta,\gamma,g) = 0 \\ f_2 &= f_2(x,y,z,\alpha,\beta,\gamma,g) = 0 \\ &\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ f_6 &= f_6(x,y,z,\alpha,\beta,\gamma,g) = 0 \end{aligned} \tag{30}$$

Differentiating all the equations against all the variables $x, y, z, \alpha, \beta, \gamma$ and g, where g is a vector including all geometric design parameters:

$$\frac{\partial f_i}{\partial x}\cdot dx+\frac{\partial f_i}{\partial y}\cdot dy+\frac{\partial f_i}{\partial z}\cdot dz+\frac{\partial f_i}{\partial \alpha}\cdot d\alpha+\frac{\partial f_i}{\partial \beta}\cdot d\beta+\frac{\partial f_i}{\partial \gamma}\cdot d\gamma+\sum_j \frac{\partial f_i}{\partial g_j}\cdot dg_j=0 \tag{19}$$

Rewrite it in matrix as

$$\begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial z} & \frac{\partial f_1}{\partial \alpha} & \frac{\partial f_1}{\partial \beta} & \frac{\partial f_1}{\partial \gamma} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} & \frac{\partial f_2}{\partial z} & \frac{\partial f_2}{\partial \alpha} & \frac{\partial f_2}{\partial \beta} & \frac{\partial f_2}{\partial \gamma} \\ \frac{\partial f_3}{\partial x} & \frac{\partial f_3}{\partial y} & \frac{\partial f_3}{\partial z} & \frac{\partial f_3}{\partial \alpha} & \frac{\partial f_3}{\partial \beta} & \frac{\partial f_3}{\partial \gamma} \\ \frac{\partial f_4}{\partial x} & \frac{\partial f_4}{\partial y} & \frac{\partial f_4}{\partial z} & \frac{\partial f_4}{\partial \alpha} & \frac{\partial f_4}{\partial \beta} & \frac{\partial f_4}{\partial \gamma} \\ \frac{\partial f_5}{\partial x} & \frac{\partial f_5}{\partial y} & \frac{\partial f_5}{\partial z} & \frac{\partial f_5}{\partial \alpha} & \frac{\partial f_5}{\partial \beta} & \frac{\partial f_5}{\partial \gamma} \\ \frac{\partial f_6}{\partial x} & \frac{\partial f_6}{\partial y} & \frac{\partial f_6}{\partial z} & \frac{\partial f_6}{\partial \alpha} & \frac{\partial f_6}{\partial \beta} & \frac{\partial f_6}{\partial \gamma} \end{bmatrix} \bullet \begin{bmatrix} dx \\ dy \\ dz \\ d\alpha \\ d\beta \\ d\gamma \end{bmatrix} = \begin{bmatrix} \sum_j \frac{-\partial f_1}{\partial g_j} dg_j \\ \sum_j \frac{-\partial f_2}{\partial g_j} dg_j \\ \sum_j \frac{-\partial f_3}{\partial g_j} dg_j \\ \sum_j \frac{-\partial f_4}{\partial g_j} dg_j \\ \sum_j \frac{-\partial f_5}{\partial g_j} dg_j \\ \sum_j \frac{-\partial f_6}{\partial g_j} dg_j \end{bmatrix} \tag{20}$$

In a compact form, it becomes

$$J_1 dX = dG \tag{21}$$

Where

$$\mathrm{dG}=\begin{bmatrix} \sum_j \frac{-\partial f_1}{\partial g_j} dg_j \\ \sum_j \frac{-\partial f_2}{\partial g_j} dg_j \\ \sum_j \frac{-\partial f_3}{\partial g_j} dg_j \\ \sum_j \frac{-\partial f_4}{\partial g_j} dg_j \\ \sum_j \frac{-\partial f_5}{\partial g_j} dg_j \\ \sum_j \frac{-\partial f_6}{\partial g_j} dg_j \end{bmatrix} = -\begin{bmatrix} \frac{\partial f_1}{\partial g_1} & \frac{\partial f_1}{\partial g_2} & \cdots & \frac{\partial f_1}{\partial g_N} \\ \vdots & \vdots & \cdots & \vdots \\ \vdots & \vdots & \cdots & \vdots \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial f_6}{\partial g_1} & \frac{\partial f_6}{\partial g_2} & \cdots & \frac{\partial f_6}{\partial g_N} \end{bmatrix}_{6\times N} \bullet \begin{bmatrix} dg_1 \\ dg_2 \\ \vdots \\ \vdots \\ dg_N \end{bmatrix}_{N\times 1} \tag{22}$$

From Equation (22) above, we have

$$dG = J_2 dg \tag{23}$$

Substitute Equation (21) into Equation (23) to obtain

$$J_1 dX = J_2 dg \tag{24}$$

$$dX = (J_1^{-1} J_2) dg \tag{25}$$

The Jacobian matrix is obtained as $J_1^{-1} \cdot J_2$

$$J = J_1^{-1} \cdot J_2 = \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial z} & \frac{\partial f_1}{\partial \alpha} & \frac{\partial f_1}{\partial \beta} & \frac{\partial f_1}{\partial \gamma} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} & \frac{\partial f_2}{\partial z} & \frac{\partial f_2}{\partial \alpha} & \frac{\partial f_2}{\partial \beta} & \frac{\partial f_2}{\partial \gamma} \\ \frac{\partial f_3}{\partial x} & \frac{\partial f_3}{\partial y} & \frac{\partial f_3}{\partial z} & \frac{\partial f_3}{\partial \alpha} & \frac{\partial f_3}{\partial \beta} & \frac{\partial f_3}{\partial \gamma} \\ \frac{\partial f_4}{\partial x} & \frac{\partial f_4}{\partial y} & \frac{\partial f_4}{\partial z} & \frac{\partial f_4}{\partial \alpha} & \frac{\partial f_4}{\partial \beta} & \frac{\partial f_4}{\partial \gamma} \\ \frac{\partial f_5}{\partial x} & \frac{\partial f_5}{\partial y} & \frac{\partial f_5}{\partial z} & \frac{\partial f_5}{\partial \alpha} & \frac{\partial f_5}{\partial \beta} & \frac{\partial f_5}{\partial \gamma} \\ \frac{\partial f_6}{\partial x} & \frac{\partial f_6}{\partial y} & \frac{\partial f_6}{\partial z} & \frac{\partial f_6}{\partial \alpha} & \frac{\partial f_6}{\partial \beta} & \frac{\partial f_6}{\partial \gamma} \end{bmatrix}^{-1} \bullet \tag{26}$$

$$\begin{bmatrix} -\frac{\partial f_1}{\partial g_1} & -\frac{\partial f_1}{\partial g_2} & \cdots & -\frac{\partial f_1}{\partial g_N} \\ \vdots & \vdots & \cdots & \vdots \\ \vdots & \vdots & \cdots & \vdots \\ \vdots & \vdots & \cdots & \vdots \\ -\frac{\partial f_6}{\partial g_1} & -\frac{\partial f_6}{\partial g_2} & \cdots & -\frac{\partial f_6}{\partial g_N} \end{bmatrix}$$

In the case with a probe on the end effecter:

From the Jacobian matrix ,$dX = (J_1^{-1} J_2) dL$, transfer the coordinate of TCP into the probe coordinates Xp, Yp and Zp as

$$\begin{bmatrix} X_p \\ Y_p \\ Z_p \\ 1 \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & x \\ R_{21} & R_{22} & R_{23} & y \\ R_{31} & R_{32} & R_{33} & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_L \\ y_L \\ z_L \\ 1 \end{bmatrix} \tag{27}$$

Differentiating Equation (20), one can obtain:

$$\begin{bmatrix} dX_P \\ dY_P \\ dZ_P \end{bmatrix} = DR \cdot \begin{bmatrix} x_L \\ y_L \\ z_L \end{bmatrix} + R \cdot \begin{bmatrix} dx_L \\ dy_L \\ dz_L \end{bmatrix} + \begin{bmatrix} dx \\ dy \\ dz \end{bmatrix} \tag{28}$$

Where

$$DR_{i,j} = R^{\alpha}_{i,j} \cdot d\alpha + R^{\beta}_{i,j} \cdot d\beta + R^{\gamma}_{i,j} \cdot d\gamma$$

$$R^{\alpha}_{i,j} = \frac{dR_{i,j}}{d\alpha} \qquad R^{\beta}_{i,j} = \frac{dR_{i,j}}{d\beta} \qquad R^{\gamma}_{i,j} = \frac{dR_{i,j}}{d\gamma} \tag{31}$$

Rewrite the equation into following forms,

$$\begin{bmatrix} dX_P \\ dY_P \\ dZ_P \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & M_{11} & M_{12} & M_{13} \\ 0 & 1 & 0 & M_{21} & M_{22} & M_{23} \\ 0 & 0 & 1 & M_{31} & M_{32} & M_{33} \end{bmatrix} \begin{bmatrix} dx \\ dy \\ dz \\ d\alpha \\ d\beta \\ d\gamma \end{bmatrix} + R \cdot \begin{bmatrix} dx_L \\ dy_L \\ dz_L \end{bmatrix} \tag{32}$$

Where $M_{ij} = DR \cdot \begin{bmatrix} x_L \\ y_L \\ z_L \end{bmatrix}$ then substitute $dX = (J_1^{-1} J_2) dL$ into Equation (32)

Finally

$$\begin{bmatrix} dX_P \\ dY_P \\ dZ_P \end{bmatrix} = \left(\begin{bmatrix} 1 & 0 & 0 & M_{11} & M_{12} & M_{13} \\ 0 & 1 & 0 & M_{21} & M_{22} & M_{23} \\ 0 & 0 & 1 & M_{31} & M_{32} & M_{33} \end{bmatrix} \cdot J \qquad R \right) \cdot \begin{bmatrix} dL_1 \\ dL_2 \\ . \\ . \\ dL_N \\ dx_L \\ dy_L \\ dz_L \end{bmatrix}$$

The final Jacobian matrix with a probe is $\begin{bmatrix} \begin{bmatrix} 1 & 0 & 0 & M_{11} & M_{12} & M_{13} \\ 0 & 1 & 0 & M_{21} & M_{22} & M_{23} \\ 0 & 0 & 1 & M_{31} & M_{32} & M_{33} \end{bmatrix} \cdot J & R \end{bmatrix}$

IntechOpen

4.6 Inverse Jacobian Matrix with a Probe

From 6 link length equations below:

$$\begin{aligned} f_1 &= f_1(\theta_1,\theta_2,\theta_3,x,y,z,\alpha,\beta,\gamma,g_1) = 0 \\ f_2 &= f_2(\theta_1,\theta_2,\theta_3,x,y,z,\alpha,\beta,\gamma,g_1) = 0 \\ &\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ f_6 &= f_6(\theta_1,\theta_2,\theta_3,x,y,z,\alpha,\beta,\gamma,g_1) = 0 \end{aligned} \tag{33}$$

where $\theta_1,\theta_2,\theta_3,$ are drive angles from actuators or motors and $x,y,z,\alpha,\beta,\gamma$ is the pose of TCP.
with the probe, one can obtain the next three equations from Euler transformation.

$$\begin{aligned} f_7 &= f_7(x,y,z,\alpha,\beta,\gamma,g_2) = 0 \\ f_8 &= f_8(x,y,z,\alpha,\beta,\gamma,g_2) = 0 \\ f_9 &= f_9(x,y,z,\alpha,\beta,\gamma,g_2) = 0 \end{aligned} \quad \text{from} \quad \begin{bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{bmatrix} = \begin{bmatrix} & & x \\ R & & y \\ & & z \\ 0 & 1 & \end{bmatrix} \cdot \begin{bmatrix} x_L \\ y_L \\ z_L \\ 1 \end{bmatrix} \tag{34}$$

Differentiate with respect to all the variables $\theta_1,\theta_2,\theta_3,x,y,z,\alpha,\beta,\gamma$ for Equation (30), where g_i is a vector including all design variables:

$df_i = 0$ or

$$\frac{\partial f_i}{\partial \theta_1}\cdot d\theta_1 + \frac{\partial f_i}{\partial \theta_2}\cdot d\theta_2 + \frac{\partial f_i}{\partial \theta_3}\cdot d\theta_3 + \frac{\partial f_i}{\partial x}\cdot dx + \frac{\partial f_i}{\partial y}\cdot dy + \frac{\partial f_i}{\partial z}\cdot dz + \frac{\partial f_i}{\partial \alpha}\cdot d\alpha + \frac{\partial f_i}{\partial \beta}\cdot d\beta + \frac{\partial f_i}{\partial \gamma}\cdot d\gamma + \frac{\partial f_i}{\partial g_1}\cdot dg_1 = 0 \tag{35}$$

Rewrite it in a matrix form

$$J_1 d\theta + J_2 dX = -\partial G_1 \tag{36}$$

$$\text{Where } J_1 = \begin{bmatrix} \frac{\partial f_1}{\partial \theta_1} & \frac{\partial f_1}{\partial \theta_2} & \frac{\partial f_1}{\partial \theta_3} \\ \frac{\partial \dot{f}_6}{\partial \theta_1} & \frac{\partial \dot{f}_6}{\partial \theta_2} & \frac{\partial \dot{f}_6}{\partial \theta_3} \end{bmatrix} \tag{37}$$

$$J_2 = \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial z} & \frac{\partial f_1}{\partial \alpha} & \frac{\partial f_1}{\partial \beta} & \frac{\partial f_1}{\partial \gamma} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} & \frac{\partial f_2}{\partial z} & \frac{\partial f_2}{\partial \alpha} & \frac{\partial f_2}{\partial \beta} & \frac{\partial f_2}{\partial \gamma} \\ \frac{\partial f_3}{\partial x} & \frac{\partial f_3}{\partial y} & \frac{\partial f_3}{\partial z} & \frac{\partial f_3}{\partial \alpha} & \frac{\partial f_3}{\partial \beta} & \frac{\partial f_3}{\partial \gamma} \\ \frac{\partial f_4}{\partial x} & \frac{\partial f_4}{\partial y} & \frac{\partial f_4}{\partial z} & \frac{\partial f_4}{\partial \alpha} & \frac{\partial f_4}{\partial \beta} & \frac{\partial f_4}{\partial \gamma} \\ \frac{\partial f_5}{\partial x} & \frac{\partial f_5}{\partial y} & \frac{\partial f_5}{\partial z} & \frac{\partial f_5}{\partial \alpha} & \frac{\partial f_5}{\partial \beta} & \frac{\partial f_5}{\partial \gamma} \\ \frac{\partial f_6}{\partial x} & \frac{\partial f_6}{\partial y} & \frac{\partial f_6}{\partial z} & \frac{\partial f_6}{\partial \alpha} & \frac{\partial f_6}{\partial \beta} & \frac{\partial f_6}{\partial \gamma} \end{bmatrix} \tag{38}$$

$$-\partial G_1 = \begin{bmatrix} \frac{-\partial f_1}{\partial g_1} dg_1 \\ \frac{-\partial f_2}{\partial g_1} dg_1 \\ \frac{-\partial f_3}{\partial g_1} dg_1 \\ \frac{-\partial f_4}{\partial g_1} dg_1 \\ \frac{-\partial f_5}{\partial g_1} dg_1 \\ \frac{-\partial f_6}{\partial g_1} dg_1 \end{bmatrix} \tag{39}$$

From Equation (33), one can obtain next formulation.

$$J_3 dX = -\partial G_2 \tag{40}$$

$$\text{Where } J_3 = \begin{bmatrix} \frac{\partial f_7}{\partial x} & \frac{\partial f_7}{\partial y} & \frac{\partial f_7}{\partial z} & \frac{\partial f_7}{\partial \alpha} & \frac{\partial f_7}{\partial \beta} & \frac{\partial f_7}{\partial \gamma} \\ \frac{\partial f_8}{\partial x} & \frac{\partial f_8}{\partial y} & \frac{\partial f_8}{\partial z} & \frac{\partial f_8}{\partial \alpha} & \frac{\partial f_8}{\partial \beta} & \frac{\partial f_8}{\partial \gamma} \\ \frac{\partial f_9}{\partial x} & \frac{\partial f_9}{\partial y} & \frac{\partial f_9}{\partial z} & \frac{\partial f_9}{\partial \alpha} & \frac{\partial f_9}{\partial \beta} & \frac{\partial f_9}{\partial \gamma} \end{bmatrix} \tag{41}$$

$$-\partial G_2 = \begin{bmatrix} \frac{-\partial f_7}{\partial g_2} dg_2 \\ \frac{-\partial f_8}{\partial g_2} dg_2 \\ \frac{-\partial f_9}{\partial g_2} dg_2 \end{bmatrix} \tag{42}$$

From Equation (36)

$$dX = -J_2^{-1} \cdot \partial G_1 - J_2^{-1} \cdot J_1 d\theta \tag{43}$$

Substituting Equation (43) into the Equation (36)

$$J_3 \cdot J_2^{-1} \cdot J_1 \cdot d\theta = J_5 \cdot \begin{bmatrix} dx_L \\ dy_L \\ dz_L \end{bmatrix} - J_3 \cdot J_2^{-1} J_4 \cdot \begin{bmatrix} dL_1 \\ dL_2 \\ . \\ dL_N \end{bmatrix} \tag{44}$$

$$\text{Where } J_4 = \begin{bmatrix} \frac{\partial f_1}{\partial L_1} & \cdots & \frac{\partial f_1}{\partial L_N} \\ \cdots & \cdots & \cdots \\ \frac{\partial f_6}{\partial L_1} & \cdots & \frac{\partial f_6}{\partial L_N} \end{bmatrix} \tag{45}$$

and

$$J_5 = \begin{bmatrix} \frac{\partial f_7}{\partial x_L} & \frac{\partial f_7}{\partial y_L} & \frac{\partial f_7}{\partial z_L} \\ \frac{\partial f_8}{\partial x_L} & \frac{\partial f_8}{\partial y_L} & \frac{\partial f_8}{\partial z_L} \\ \frac{\partial f_9}{\partial x_L} & \frac{\partial f_9}{\partial y_L} & \frac{\partial f_9}{\partial z_L} \end{bmatrix} \tag{46}$$

Finally

$$d\theta = [J_3 \cdot J_2^{-1} \cdot J_1]^{-1} \cdot [-J_3 \cdot J_2^{-1} \cdot J_4 \vdots J_5] \cdot \begin{bmatrix} dL_1 \\ . \\ . \\ dL_N \\ dx_L \\ dy_L \\ dz_L \end{bmatrix} \tag{47}$$

$$J_{INVERSE} = [J_3 \cdot J_2^{-1} \cdot J_1]^{-1} \cdot [-J_3 \cdot J_2^{-1} \cdot J_4 \vdots J_5]$$

4.7 Determination of Independent Design Variables Using SVD Method

With the reality that all the parts of a robot have manufacturing errors and misalignment errors as well as thermal errors, errors should be considered for any of the components in order to accurately model the accuracy of the robotic system. Error budget is carried out in the study and error sensitivity of robot kinematics with respect to any of the parameters can be obtained based on error modeling. This is realized through the established Jacobian matrix.
To find those parameters in the error model that are linearly dependent and those parameters that are difficult to observe, the Jacobian matrix is analyzed. SVD method (Singular Value Decomposition) is used in such an analysis.
A methodical way of determining which parameters are redundant is to investigate the singular vectors. An investigation of the last column of the V vector will reveal that some elements are dominant in order of magnitude. This implies that corresponding columns in the Jacobian matrix are linearly dependent. The work of reducing the number of error parameters must continue until no singularities exist and the condition number has reached an acceptable value.
A total of 31 redundant design variables of the 71 design parameters are eliminated by observing the numerical Jacobian matrix obtained. Table 7 in Section 6 lists the remaining calibration parameters.

4.8 Error Budget

When the SVD is completed and a linearly independent set of error model parameters determined, the Error Budget can be determined. The mathematical description of the error budget is as follows:

$$\begin{aligned} J &= U \bullet S \bullet V^T \\ dX &= J \bullet dg = U \bullet S \bullet V^T \bullet dg \\ U^T &\bullet dX = S \bullet V^T \bullet dg \end{aligned} \tag{48}$$

Assume $U^T \bullet dX = d\overline{X}$ and $V^T \bullet dg = d\overline{g}$. So we have $d\overline{g} = d\overline{X} / S_{ii}$, finally,

$$dg = (V \bullet U^T \bullet dX) / S_{ii} \tag{49}$$

Thus if the dX is given as the accuracy of the TAU robot, the error budget dg can be determined.
Given the D-H parameters for all three upper arms and the main column, the locations of the joints located at each of the three upper arms can be known accurately. The six chain equations are created for the six link lengths, as follows:

$$F = \begin{Bmatrix} f1(upperarm_point, TCP_point) \\ f2(upperarm_point, TCP_point) \\ f3(upperarm_point, TCP_point) \\ f4(upperarm_point, TCP_point) \\ f5(upperarm_point, TCP_point) \\ f6(upperarm_point, TCP_point) \end{Bmatrix}$$

Where

$$TCP_point = f(px, py, pz, \alpha, \beta, \gamma)$$

$$Upperarm_point = f(\varepsilon)$$

and ε is a collection of all the design parameters. Thus,

$$F = \begin{Bmatrix} F1(\varepsilon, px, py, pz, \alpha, \beta, \gamma) \\ F2(\varepsilon, px, py, pz, \alpha, \beta, \gamma) \\ F3(\varepsilon, px, py, pz, \alpha, \beta, \gamma) \\ F4(\varepsilon, px, py, pz, \alpha, \beta, \gamma) \\ F5(\varepsilon, px, py, pz, \alpha, \beta, \gamma) \\ F6(\varepsilon, px, py, pz, \alpha, \beta, \gamma) \end{Bmatrix} \tag{50}$$

An error model is developed based on the system of equations as described above. A total of 71 parameters are defined to represent the entire system. The 71 parameters include all the D-H parameters for the 3 upper arms, as well as the coordinates (x, y, z) of the 6 points at both ends of the 6 links, respectively. Table 8 in Section 6 presents the error budget.

4.9 Dexterity Analysis

From the inverse kinematics ,

$$S_i = RP_i^h \tag{51}$$

Where P_i^h denotes the position of the center on the end plate in local coordinate. R is the

transfer matrix of coordinate. So the link vector

$$L_i = P_h + S_i - P_i^b \tag{52}$$

P_h is the position coordinate of the center on the end plate. From the end plate velocities

to link velocities, We define the Jacobian matrix by

$$\dot{L} = J\dot{X} \tag{53}$$

Where $\dot{L}$ is the vector of link velocities and $\dot{X} = [\dot{P}_h^T, \omega^T]^T$ is the velocity vector

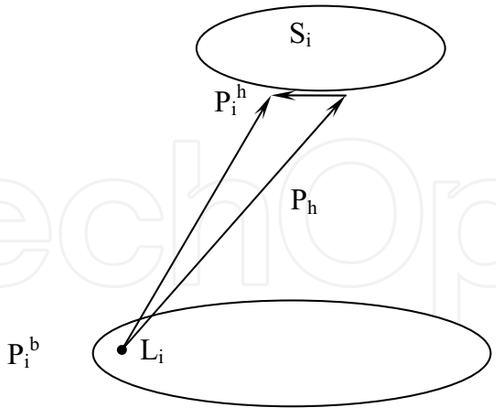


Figure 17. Vectors

Differentiating Equation (52) we can get

$$\dot{l}_i z_i + l_i \dot{z}_i = \dot{p}_h + (\omega \times S_i) \quad \text{(Zi is the unit vector of Li vector)} \tag{54}$$

Taking the inner product with Zi yields

$$\dot{l}_i = z_i \bullet \dot{p}_h + (S_i \times z_i) \bullet \omega \quad (\text{ from the } z_i \bullet (\omega \times s_i) = (s_i \times z_i) \bullet \omega \quad) \tag{55}$$

Thus we can obtain the Jacobian matrix as

$$J = \begin{bmatrix} z_1^T & (s_1 \times z_1)^T \\ .. & .. \\ z_6^T & (s_6 \times z_6)^T \end{bmatrix} \tag{56}$$

The dexterity is defined as

$$A = \frac{\lambda_{\max}(J^{-1})}{\lambda_{\min}(J^{-1})} \tag{57}$$

where λ is the eigen-value of the Jacobian matrix.

5. System Stiffness

The stiffness of the robot is a very important performance, which will have a significant influence on the robotic applications like cutting, milling, grinding

etc. In this chapter, general formulations for the stiffness of robotic system and the stiffness measurement result are presented, TCP stiffness is calculated based on theoretical analysis and modeling. In the stiffness analysis, the stiffness of individual component in related directions will be the output of stiffness model.

5.1 The Measurement of the Robot Stiffness

Based on the designed robot with certain component errors, Error modeling will be used to map the robot error over its working space. Thermal model will also be established. Deflection under load will be part of the modeling too. This comprehensive error model is the base for error analysis and robotic product design. It will also be used, or partly used for error compensation. For error compensation, however, suitable sensors will have to be used.
As measurement is concerned, it is important is to choose the suitable performance evaluation standard. The type of sensors will be selected based on the evaluation method. In selecting the sensors, resolution, repeatability, and accuracy under certain environments will be the key to consider. The factors of price and user-friendliness will also be weighted heavily. Measurement procedure will be carefully generated and measurement will be performed using certified metrology equipment only to ensure the results.

5.2 Formulations of the Robotic System Stiffness

A solution to the inverse kinematics problem is required for stiffness calculation. It is briefly described below. Referring to standard Stewart Platform the *i-th* leg length *li* is given by

$$li = gi(R,d) \tag{58}$$

where $d = [x,y,z]$, is the position vector of the platform coordinate system's origin in the base coordinate system, *li* is the length of the *i-th* leg and *gi* is only a function of *R* and *d* for constant geometric the *i-th* leg parameters.

$$R = \begin{bmatrix} \cos\phi vos\theta & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi \\ \sin\phi\sin\theta & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi \\ -\sin\phi & \cos\theta\sin\psi & \cos\theta\cos\psi \end{bmatrix} \tag{59}$$

Above is the rotation matrix relating the platform's coordinate system, to the base co-ordinate system, Here *R* is constructed using Roll-Pitch-Yaw (RPY) angle rotations, where *R (roll)* = ϕ around the z axis, P (pitch) = θ around the y axis, and Y (yaw) = ψ around the x axis.

Thus, R is a rotation about the x axis of ψ, followed by θ, a rotation around y axis, and ending with a rotation of ϕ around z axis.

Equation (58) represents the inverse kinematic solution. For some *R* and *d*, the *i*-th leg length (*li*) can be easily calculated.

If Equation (58) is expanded using Taylor series expansion, and the first order term considered only, the change in leg length, Δli, is obtained as a row vector *Ji*, multiplied by the column twist vector Δp as given below:

$$\Delta p = Ji\ \Delta li \tag{60}$$

where

$$J_i = \left[\frac{\partial x}{\partial g_i}, \frac{\partial y}{\partial g_i}, \frac{\partial z}{\partial g_i}, \frac{\partial \psi}{\partial g_i}, \frac{\partial \theta}{\partial g_i}, \frac{\partial \phi}{\partial g_i} \right] \tag{61}$$

And

$$\Delta p = [\Delta x, \Delta y, \Delta z, \Delta \psi, \Delta \theta, \Delta \phi]^T . \tag{62}$$

Assembling the equations for all the legs of the mechanism,

$$\Delta p = J \Delta q \tag{63}$$

where $\Delta q = [\Delta l_1, \Delta l_2, \Delta l_3, \Delta l_4, \Delta l_5, \Delta l_6]^T$.

From the principle of duality between the force/torque and velocity fields, or what is more commonly known as contragradience

$$f = J^T \tau \tag{64}$$

where

$$\tau = [F_x, F_y, F_z, M_x, M_y, M_z]^T$$

is the end effector wrench, and

$$f = [f_1, f_2, f_3, f_4, f_5, f_6]^T$$

is the vector of forces experienced by the legs, and J^T is the transpose of the Jacobian *J,* (described earlier).

As previously mentioned, the static stiffness (or rigidity) of the mechanism can be a primary consideration in the design of a parallel link manipulator for certain applications (specifically, those involving large forces and high accuracy).

The static stiffness of the PLM is a function of:

- The limbs' structure and material.
- The joints' stiffnesses.
- The platform and base stiffness.
- The geometry of the structure.
- The topology of the structure.
- The end-effector position and orientation.

To ensure meeting the stiffness specifications, it becomes important to estimate the stiffness, particularly the lowest stiffness value and the direction in which it is experienced, for the manipulator in a given posture or configuration. In the following analysis, this problem is addressed. Algebraic expressions for stiffness (both the engineering and the general, to be defined later) are developed. The fact that the minimum stiffness is experienced in the direction of the eigenvector that corresponds to the minimum eigenvalue of the 'stiffness matrix' of the manipulator is shown. A corresponding result can be obtained for the maximum stiffness of the manipulator. Finally, expressions are developed for the stiffness of the manipulator in any direction.

The basic assumption for the theory developed is:

- The joints are frictionless.
- The weights of the legs or arms are negligible.

The rigidity of the platform and the base is much greater than that of the legs and, therefore, can be considered as infinite (or in general, the manipulator's joints are the least stiff elements in the structure, and hence, dictate the manipulator stiffness). If k is the axial or arm stiffness, then for the i-th leg or arm

$$f_i = k_i \Delta l_i \tag{65}$$

where f_i is the force needed to cause a Δl_i change of the i-th leg length. Assembling the equations for all the legs, Equation (65) becomes

$$f_i = k\Delta q \tag{66}$$

Substituting for Δq from Equation (63)

$$f = kJ^{-1}\Delta p \tag{67}$$

Multiplying both sides of Equation (67) with J^{-T} and substituting f with $J^T\tau$ from Equation (64) to obtain

$$\tau = J^{-T}kJ^{-1}\Delta p. \tag{68}$$

Equation (68) can be interpreted as τ is the wrench required to cause the platform to experience a twist of Δp. So the stiffness is obtained as

$$J^{-T}kJ^{-1} \tag{69}$$

5.3 Method for Measuring Joint Stiffness

From Equation (69), the stiffness if the robot can be obtained, including the component or joint stiffness K_i. In order to obtain the total stiffness of the robot, the joint stiffness has to be measured.
From Equation (68), the following Equation (70) can be obtained by finding the inverse of the matrix $J^{-T}kJ^{-1}$ as

$$\Delta p = JK_i^{-1}J^T\tau. \tag{70}$$

Equation (70) is very important for measuring the joint stiffness. Many different equations can be obtained by applying different force τ with different directions then measuring the deflections Δp. Least square method is applied to solve Equation (70). As variable $1/K_i$ is the unknown, one can simplify Equation (70) as linear equations since $K_i^{-1} = [1/k_i]$ is a diagonal matrix.

5.4 Results of the Stiffness Measurement

The instrument used in measuring includes:

- CMM ROMER 3000i Digitizer with an accuracy of 5μm
- Sphere with an accuracy of 0.02 mm

Pose measurement is carried out first as seen in Fig. 5.1. The conditions are

J1=84.7^0, J2=-3.6^0, J3=38.8^0, J4=-0.3^0, J5=50.6^0 and J6=-110.2^0.
Load F_x=-360N

And the measured deformation is

Δx = -0.69 mm, Δy = 0.37 mm, and Δz = -0.13 mm

Condition for Deflection Measurement:

- Measure robot translational deflections by the position of the center of the sphere, which is calculated based on the measurement result of the portable CMM ROMER.
- Motor servo is active during the measurement to take account of the controller stiffness.



Figure 18. Measurement Set-up

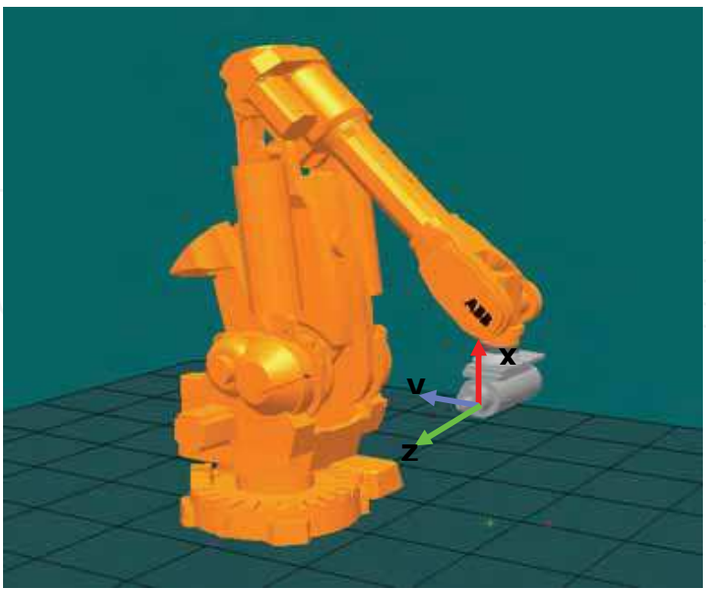
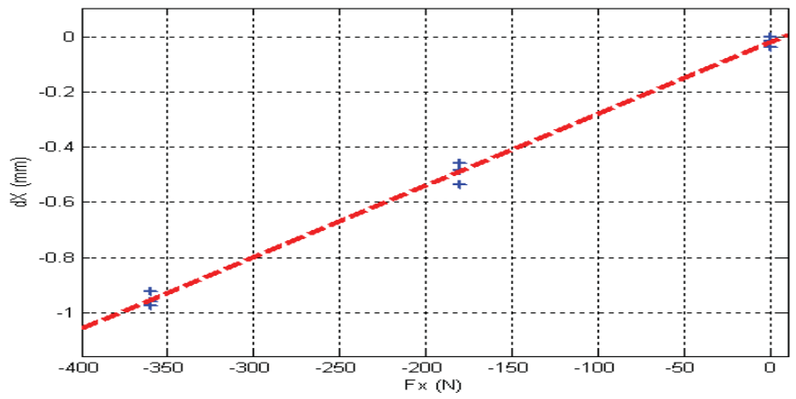
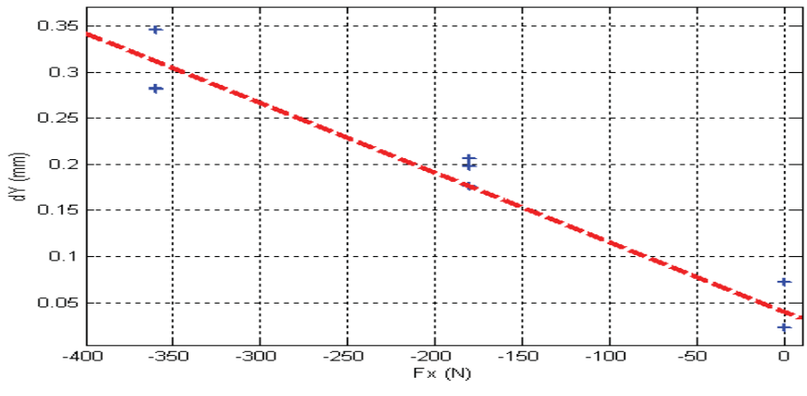


Figure 19 . Configuration of the IRB 4400 Robot





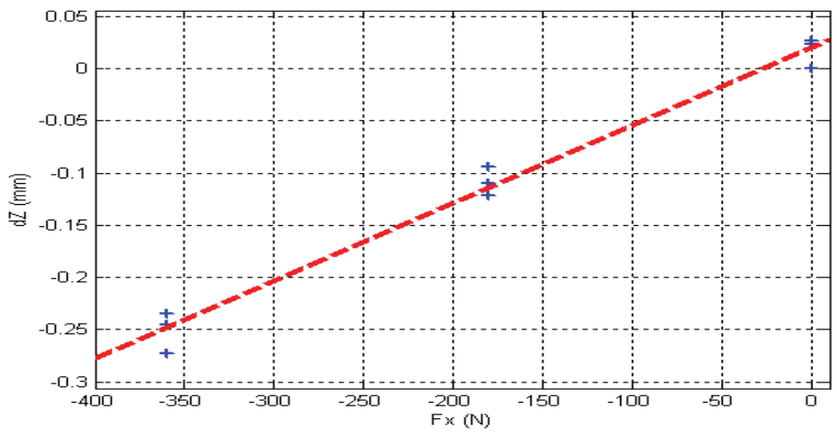


Figure 20. Measured Deflection/Deformation Dx, Dy, and Dz

Another measurement pose, as seen in Fig. 5.4 is J1=45.60, J2=-23.60, J3=37.20, J4=52.10, J5=52.10 and J6=-194.80. Load condition is
Load Fx=-360N
And the measured deformation is
Δx = -1.05 mm, Δy = -0.01 mm and Δz = -0.57 mm

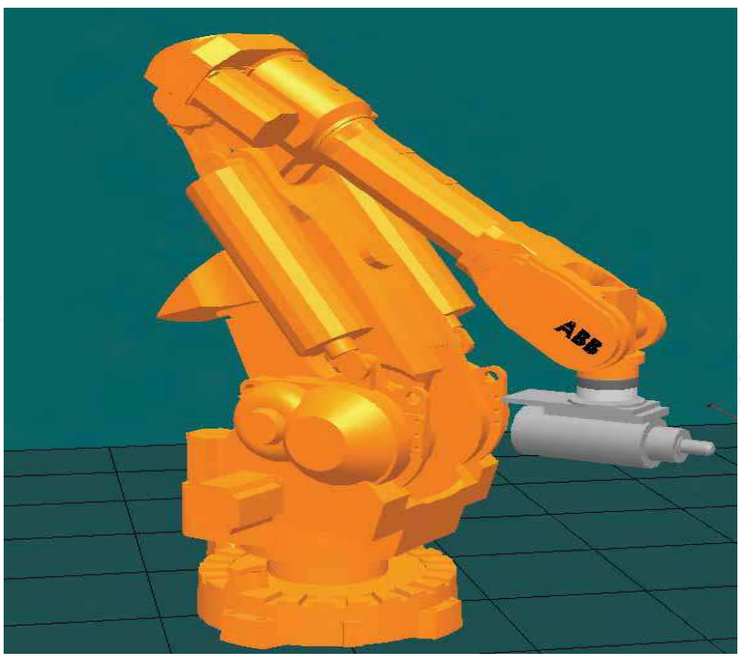


Figure 21. Configuration of the IRB 4400 Robot

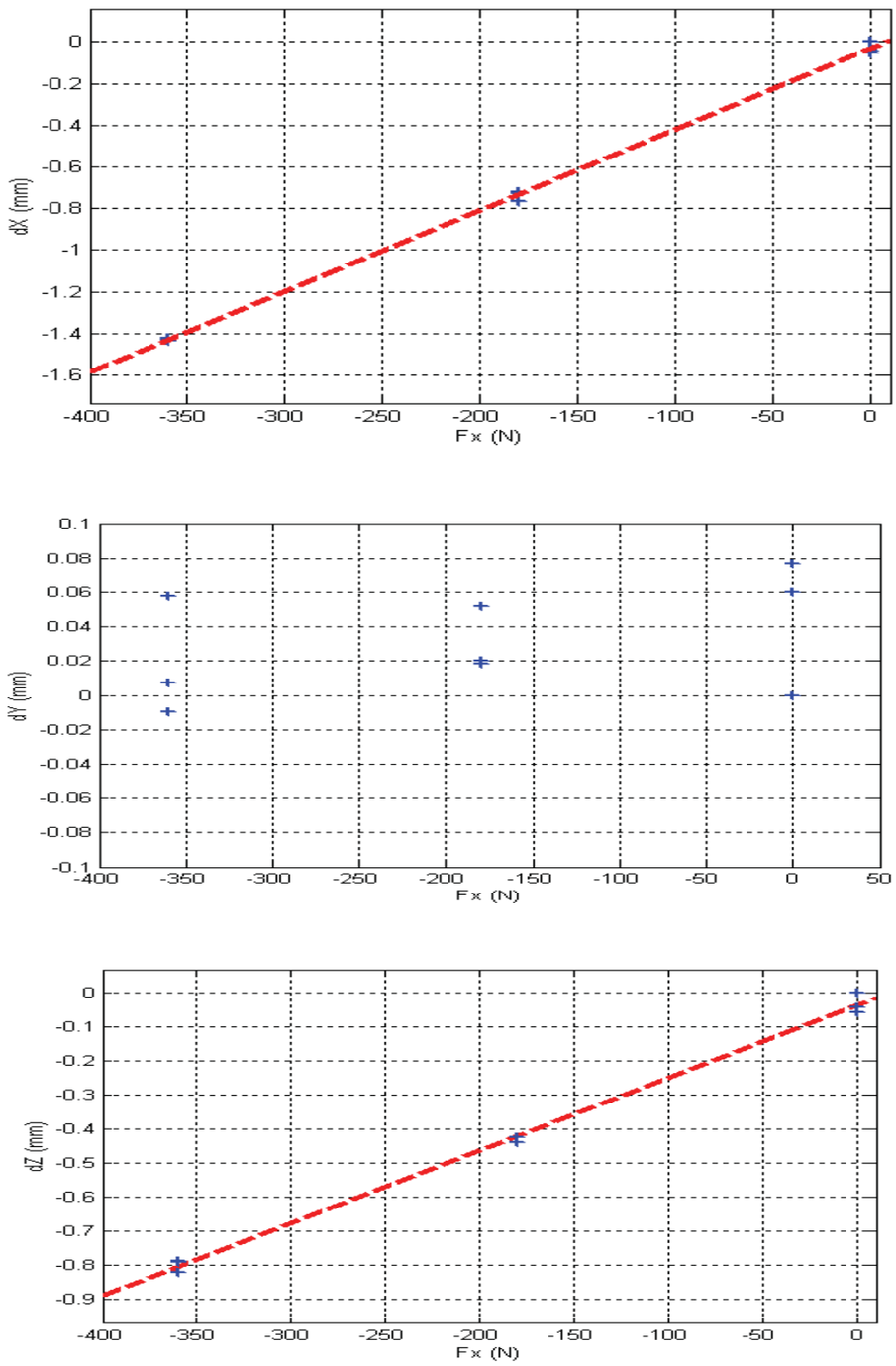


Figure 22. Measured Deflection/Deformation Dx, Dy, and Dz (Second pose)

Fx	Fy	Fz	dx	dy	dz
-180	0	0	-0.4561	0.1767	-0.1211
-360	0	0	-0.9232	0.2812	-0.2723
-360	0	0	-0.9604	0.2825	-0.2452
-180	0	0	-0.4822	0.1983	-0.0943
-180	0	0	-0.5359	0.2062	-0.1103
-360	0	0	-0.9775	0.3464	-0.2344
-180	0	0	-0.7276	0.0201	-0.4238
-360	0	0	-1.423	0.0073	-0.8206
-360	0	0	-1.4246	-0.0099	-0.7893
-180	0	0	-0.768	0.0184	-0.44
-180	0	0	-0.7194	0.0518	-0.4242
-360	0	0	-1.4357	0.0577	-0.7922
0	-275	25	0.0061	-0.8927	0.0336
0	-275	25	-0.0004	-0.9184	-0.0111
-40	-295	10	0.134	-1.1826	-0.0926
-40	-295	10	0.1308	-1.2146	-0.1407
-360	0	0	-0.9344	0.2758	-0.2987

Table 4. Measured Deformation Data

Then solve the K_q in Equation (70) $\Delta x = JK_q^{-1}J^T F$ with the least square method.

The final result is as follows:

	lsqr result	Nominal value
Axis 1:	19.03	22~80
Axis 2:	14.6	32~42
Axis 3:	45.83	25~39
Axis 4:	31.26	70
Axis 5:	15.16	50
Axis 6:	15	85

Table 5.Calculated Joint Stiffness

Fig. 5.6 also gives the standard deviation from the measurement data.

Based on the results, the measurement data can be trusted and the standard deviation of residual error is 0.042mm. Also, verification of solved stiffness agrees well. The stiffness model can provide a method for position compensation to reach a high level accuracy, with a force sensor measuring the process force in real time, the impact on position deformation can be estimated and compensated.

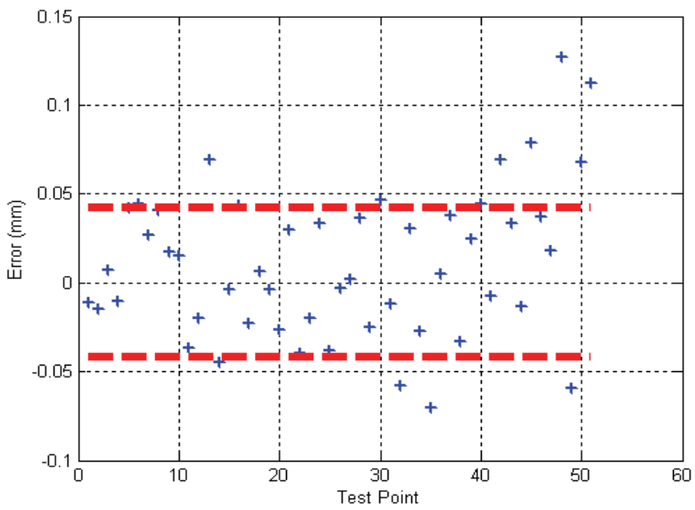


Figure 23. Residual Error Std = 0.042 mm

The same procedure can be applied to the TAU robot. The stiffness at TCP point was measured by applying a load at TCP and measuring the resulting displacements, see Figure 24. The results of the measurements are shown in Figure 25.



Figure 24. Setup of the TAU robot's Stiffness Measurement

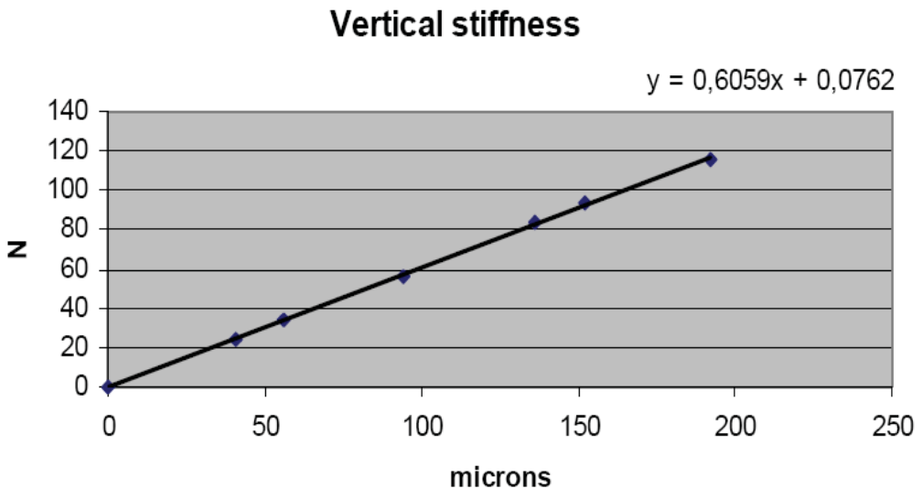


Figure 25. Measured Stiffness of the TAU robot

5.5 Application of the Robot Stiffness: Position Compensation

Position compensation can be made once the stiffness model is established. The application is to compensate the position error caused by the cutting force of milling processing.



Figure 26.Robotic Milling Setup

First, the surface quality of the aluminum block can be recorded as cutting without position compensation then cutting again to measure the surface quality with on line compensation. The surface will be measured via the laser. See Fig. 5.9 for the robotic milling setup. Based on the result shown in Figs. 5.10 and 5.11, the compensation procedure is effective reducing the error to less than 0.1 mm compared with the original error of 0.5 mm.

Conclusions:
Verification of solved stiffness agrees very well. The stiffness model can provide a method to model and test robot stiffness, with a force sensor measures the process force in real time, the impact on position deformation can be estimated and compensated.

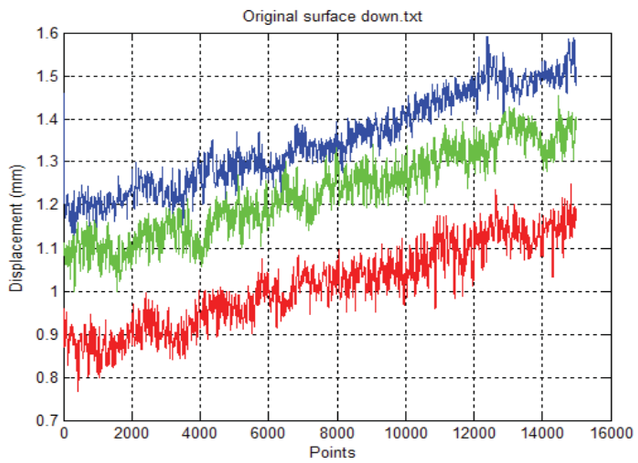
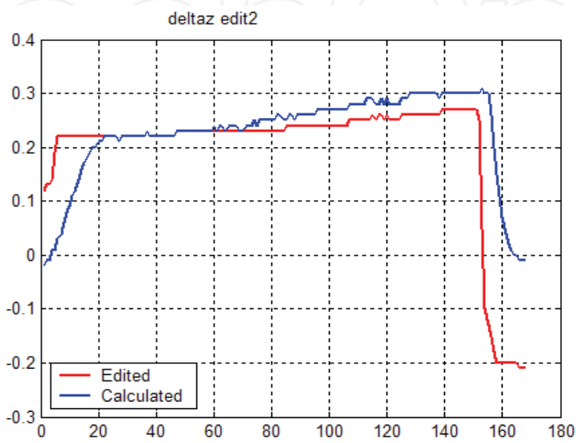


Figure 27. Surface Quality without Compensation



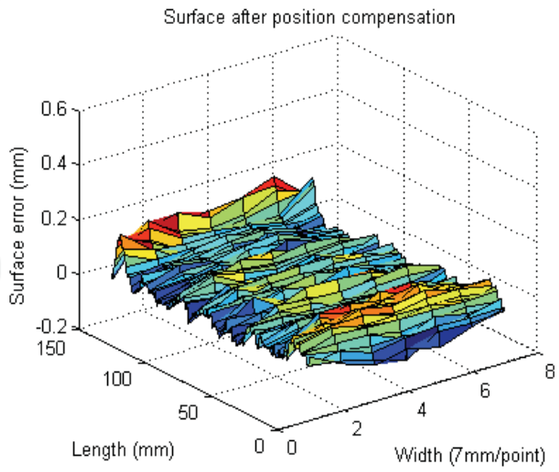


Figure 28. Surface Quality with Compensation, Mean Error < 0.1 mm

6 . Simulation and Experimental Results

The validation of the analytical model has been carried out, as well as the realization of control scheme. Besides the analytical result and data, additional results used in this chapter come from three sources:
Simulation results from ADAMS simulation software, see Figure 29 for details;
Test results from two-arm test platform, see Table 11;
Test results from the TAU prototype.

6.1 Validation of Jacobian Matrix and N-R Method

The Jacobian Matrix and N-R method need to be verified to guarantee their correctness. These simulations are made by ADAMS (commercial simulation software) see Figure 29.
The effect of the robot configurations were considered, all "verification points" are located in the whole work-space and with total different configurations. Figures 30, 6.3, and 6.4 show position differences between the N-R method and the ADAMS simulation, which indicates that accurate results have been obtained up to 0.06 um compared with ADAMS simulation results. These results guarantee the correctness of Jacobian Matrix and N-R method. Based on the simulation results, the N-R method with analytical Jacobian matrix can be used in error modeling, error budget, offline calibration.
Like most of the methods in this thesis this method suffers from a drawback: it can not be used in online position compensation and online control because it is an iteration method even with an analytical, full size Jacobian Matrix. Next

section will focus on the Jacobian Approximation Method (JAM), which is able to deal with the online compensation and online control problems.



Figure 29. Using Adams to Verify the Analytical and Error Model

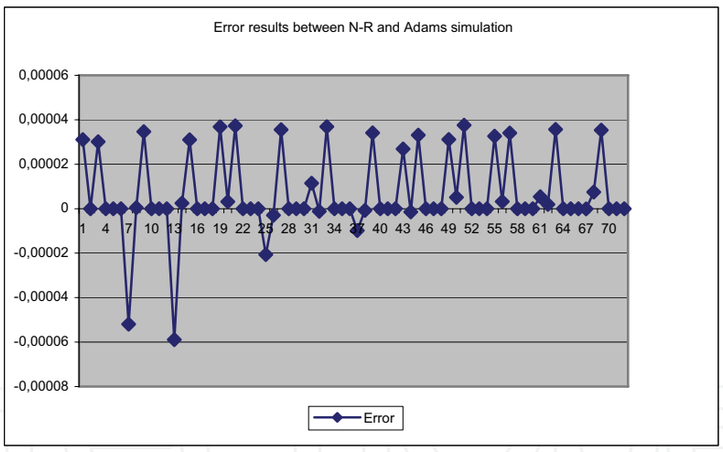
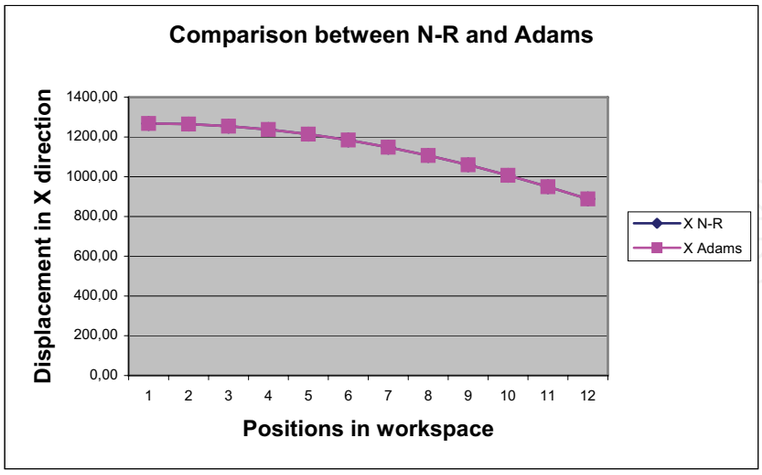
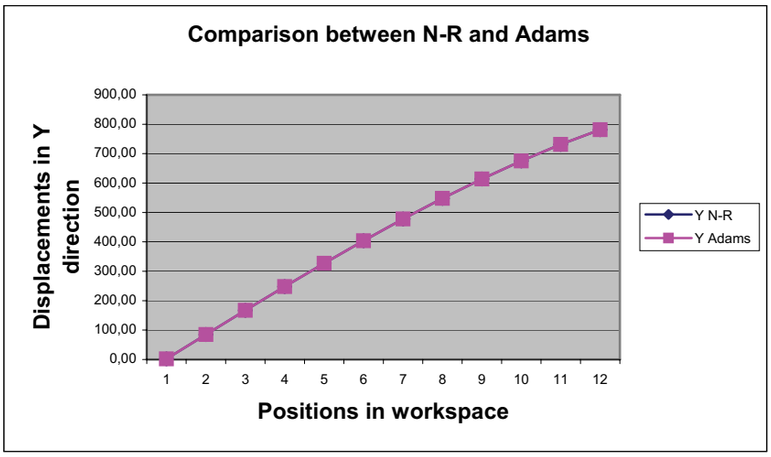
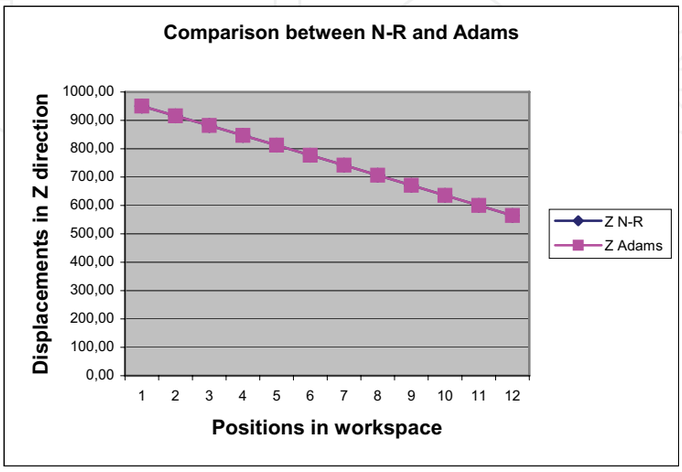


Figure 30. Position Error between the N-R Method and ADAMS Simulation


Comparison between N-R and Adams
Displacement in X direction
1400,00
1200,00
1000,00
800,00
600,00
400,00
200,00
0,00
1 2 3 4 5 6 7 8 9 10 11 12
Positions in workspace
X N-R
X Adams


Comparison between N-R and Adams
Displacements in Y direction
900,00
800,00
700,00
600,00
500,00
400,00
300,00
200,00
100,00
0,00
1 2 3 4 5 6 7 8 9 10 11 12
Positions in workspace
Y N-R
Y Adams


Comparison between N-R and Adams
Displacements in Z direction
1000,00
900,00
800,00
700,00
600,00
500,00
400,00
300,00
200,00
100,00
0,00
1 2 3 4 5 6 7 8 9 10 11 12
Positions in workspace
Z N-R
Z Adams

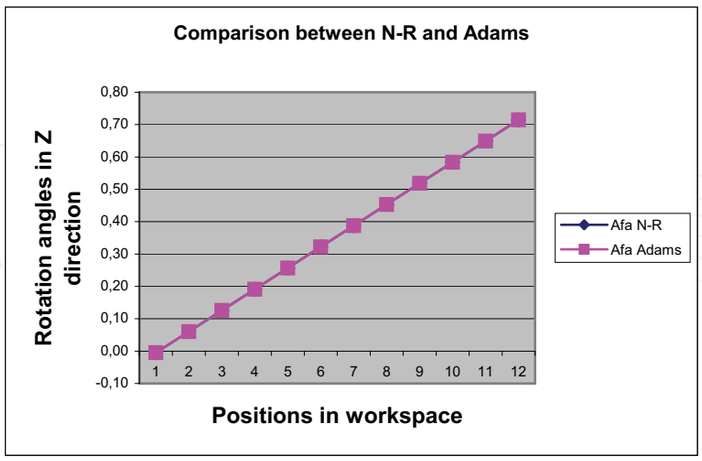


Figure 31. Results of N-R and ADAMS (Input Error Δ Link11=1mm)

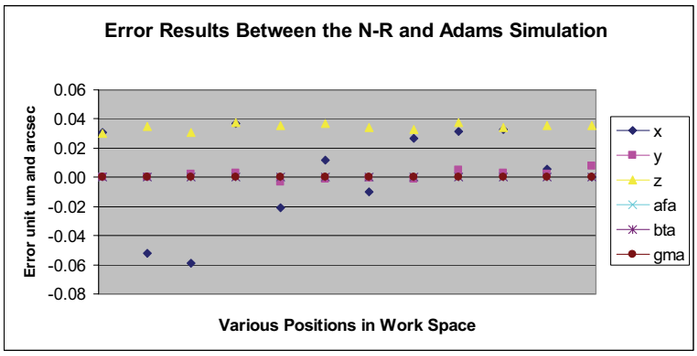


Figure 32. TCP Difference between ADAMS Simulation and N-R Method

6.2 Validation of Jacobian Approximation Method, Error Budget and Calibration

The Jacobian approximation method is verified by the following two different approaches:

(1) 6-DOF forward kinematic analysis (Newton-Raphson method), and
(2) ADAMS simulation results.

Based on the D-H model of TAU with all error parameters, inverse and forward kinematic models have been established. From the point of view of mathematics, the TAU kinematic problem is to solve 6 nonlinear equations using Newton-Raphson method with Jacobian matrix as the searching direction

and accurate results have been obtained up to 0.06 um compared with ADAMS simulation results.

It can be observed from the Figure 34, for data in detail, see Table 6, the JAM (Jacobian Approximation Method) is effective with an accuracy of 1.53 μm with an input error of 1 mm (Link 1 of lower arm 1). This was verified using ADAMS simulation results. Results from N-R method match very well with ADAMS simulation with a difference of only 0.06 μm.

The JAM can be used in on-line control and position compensation of the robot. For the TAU robot, a closed form solution of a forward kinematics problem is reached with a high accuracy instead of N-R numerical solution. The simulation results are almost perfect compared with that from ADAMS.

A series of results have been presented for error analysis. Figure 34 shows the results of SVD calibration. Which indicates the number of independent design variable is reduced from 71 to 31. A sudden drop can be observed from the Figure 34, which indicates other parameters behind variable #31 are not necessary and their effects on error model can be neglected. From Table 7, totally 40 redundant variables are removed also Table 10 gives the result of error budge.

Tables 8 and 9 give the actuator (driving motor) error and thermal error, which indicate the change of temperature should be controlled within $\pm 5^0$C otherwise the accuracy of system can not be reached to 50um. The resolution of drive motor should be at least < 10 arc second (1arc second=1/3600 degree).

SVD calibration is carried out for three parameters that contribute to the final position error, see Table 11 and 12. These parameters are Arm3 length, link13 length, and link12 length. Calibration process is completed for only1 iteration. Based on the Table 12 the accuracy of calibration is 4um for Link12 and others are below 1um, which indicates the calibration method and error model are correct.

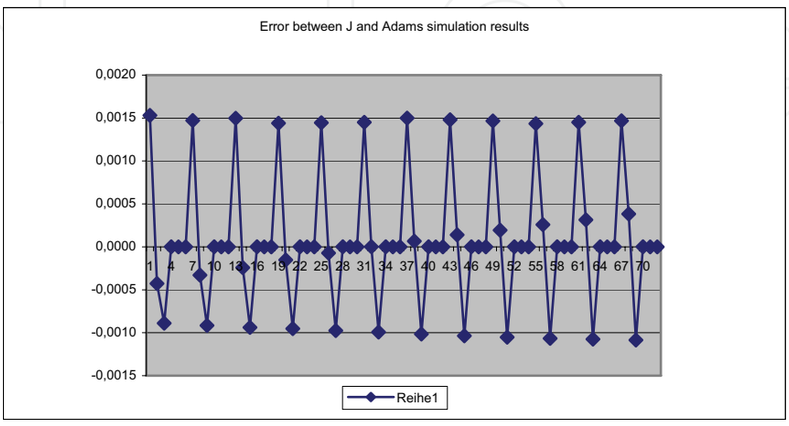


Figure 33. Position Error between Jacobian Approximation Method and ADAMS

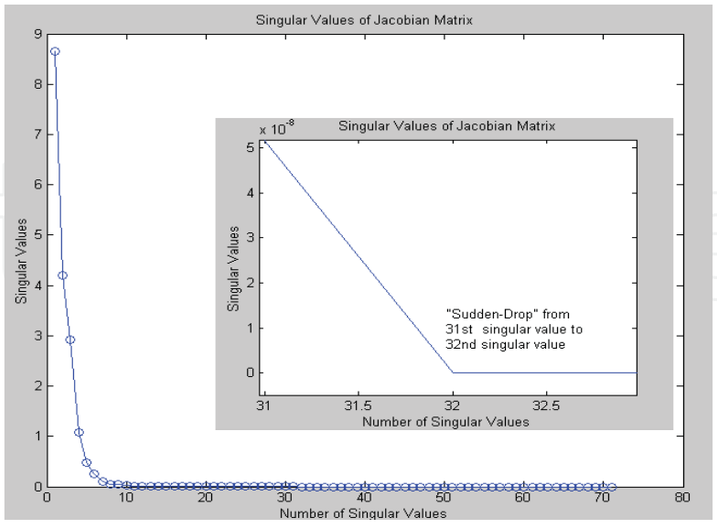


Figure 34. SVD Calibration of TAU Robot

Drive Angles	TCP Pose	Jacobian	Newton_raphson	Error between J and N
joint1=0 joint2=0 joint3=0	X	0,00E+00	1,53E-03	0,001531339
	Y	-1,81E+00	-1,81E+00	-0,0049559
	Z	-1,61E-16	-9,20E-04	-0,000919889
	afa	5,01E-03	5,01E-03	2,634E-07
	bta	-9,32E-19	-9,33E-19	-1,00679E-21
	gma	-9,32E-19	-9,32E-19	-1,5976E-22
joint1=3.75 joint2=3.75 joint3=2	X	1,19E-01	1,20E-01	0,00119916
	Y	-1,81E+00	-1,81E+00	-0,0009736
	Z	-2,09E-16	-9,45E-04	-0,000945048
	afa	5,01E-03	5,01E-03	2,7566E-06
	bta	0,00E+00	9,46E-16	9,45683E-16
	gma	0,00E+00	-4,84E-16	-4,84153E-16
joint1=7.5 joint2=7.5 joint3=4	X	2,37E-01	2,38E-01	0,00135537
	Y	-1,80E+00	-1,80E+00	0,0007562
	Z	-1,79E-16	-9,69E-04	-0,000968876
	afa	5,02E-03	5,02E-03	3,547E-07
	bta	0,00E+00	3,15E-16	3,14853E-16
	gma	0,00E+00	-4,82E-16	-4,82129E-16
joint1=11.25 joint2=11.25 joint3=6	X	3,54E-01	3,55E-01	0,00149511
	Y	-1,78E+00	-1,78E+00	0,0001837
	Z	-1,79E-16	-9,91E-04	-0,000991397
	afa	5,03E-03	5,03E-03	3,263E-06
	bta	0,00E+00	-3,10E-18	-3,10077E-18
	gma	-9,32E-19	1,15E-18	2,0782E-18
joint1=15 joint2=15 joint3=8	X	4,70E-01	4,71E-01	0,00111796
	Y	-1,75E+00	-1,75E+00	-0,0027737
	Z	-5,96E-17	-1,01E-03	-0,001012624
	afa	5,05E-03	5,05E-03	1,7286E-06
	bta	0,00E+00	0,00E+00	0
	gma	0,00E+00	0,00E+00	0
joint1=18.75 joint2=18.75 joint3=10	X	5,83E-01	5,85E-01	0,00173003
	Y	-1,72E+00	-1,72E+00	0,0017688
	Z	-5,96E-17	-1,03E-03	-0,001032565
	afa	5,07E-03	5,08E-03	6,0465E-06
	bta	4,66E-19	-6,39E-16	-6,39425E-16
	gma	-9,32E-19	9,59E-16	9,6015E-16
joint1=22.5 joint2=22.5 joint3=12	X	6,94E-01	6,96E-01	0,00184612
	Y	-1,68E+00	-1,68E+00	0,0036642
	Z	2,09E-16	-1,05E-03	-0,00105122
	afa	5,11E-03	5,11E-03	-3,4323E-06
	bta	0,00E+00	-8,47E-22	-8,47033E-22
	gma	0,00E+00	8,47E-22	8,47033E-22
joint1=26.25 joint2=26.25 joint3=14	X	8,03E-01	8,04E-01	0,00099179
	Y	-1,63E+00	-1,63E+00	0,002734
	Z	0,00E+00	-1,07E-03	-0,001068582
	afa	5,14E-03	5,14E-03	3,7091E-06
	bta	0,00E+00	3,26E-16	3,25672E-16
	gma	0,00E+00	-4,78E-16	-4,77901E-16
joint1=30 joint2=30 joint3=16	X	9,07E-01	9,09E-01	0,00170544
	Y	-1,57E+00	-1,57E+00	-0,0012306
	Z	-2,09E-16	-1,08E-03	-0,001084643
	afa	5,19E-03	5,19E-03	-2,0346E-06
	bta	0,00E+00	8,47E-22	8,47033E-22
	gma	0,00E+00	0,00E+00	0
joint1=33.75 joint2=33.75 joint3=18	X	1,01E+00	1,01E+00	-0,0004597
	Y	-1,51E+00	-1,51E+00	0,0015319
	Z	1,49E-16	-1,10E-03	-0,001099391
	afa	5,24E-03	5,24E-03	-7,54E-08
	bta	0,00E+00	-6,75E-16	-6,74923E-16
	gma	0,00E+00	4,55E-18	4,54772E-18
joint1=37.5 joint2=37.5 joint3=18	X	1,10E+00	1,11E+00	0,0060663
	Y	-1,44E+00	-1,44E+00	0,0007547
	Z	2,98E-17	-1,11E-03	-0,001112819
	afa	5,30E-03	5,30E-03	2,869E-07
	bta	0,00E+00	0,00E+00	0
	gma	0,00E+00	0,00E+00	0
joint1=41.25 joint2=41.25 joint3=22	X	1,20E+00	1,20E+00	-0,002128
	Y	-1,36E+00	-1,36E+00	-0,0038563
	Z	-2,98E-17	-1,12E-03	-0,001124931
	afa	5,37E-03	5,37E-03	-1,1E-07
	bta	0,00E+00	0,00E+00	0
	gma	0,00E+00	0,00E+00	0

Table 6. Comparison between the Results of JAM and N-R Method

Parameter Number	Parameter Definition	Parameter
16	height of the TCP	a
22	joint 3	a6
23	arm3	a7
24	joint 1 & arm 1	d1
25	short arm 1	d3
28	joint3	d6
31	joint_link11_arm1	y1
34	joint_link21_arm1	y2
37	joint_link31_arm1	y3
40	joint_link12_arm2	y4
43	joint_link22_arm2	y5
46	joint_link13_arm3	y6
48	joint_link11p	x11
49	joint_link11p	y11
51	joint_link31p	x22
52	joint_link31p	y22
54	joint_link21p	x33
55	joint_link21p	y33
56	joint_link21p	z33
57	joint_link12p	x44
58	joint_link12p	y44
59	joint_link12p	z44
60	joint_link22p	x55
61	joint_link22p	y55
62	joint_link22p	z55
63	joint_link13p	x66
64	joint_link13p	y66
67	link11	L1
68	link31	L2
69	link21	L3
70	link22	L4

Table 7. List of the Independent Design Variables

Actuator Error			X=1731mm	Y=0 mm	Z=1125mm
$\Delta\theta_1$	$\Delta\theta_2$	$\Delta\theta_3$	ΔX	ΔY	ΔZ
0	0	0	0	0	0
+/-100 arcsec	0	0	-0.1154 0.1149	0.2126 -0.2126	-0.7599 0.7598
0	+/-100 arcsec	0	0.3677 -0.3678	0.1605 -0.1605	0.2435 -0.2433
0	0	+/-100 arcsec	0 0	0.8392 -0.8392	0 0
+/-100 arcsec	+/-100 arcsec	0	0.2524 -0.2528	0.3732 -0.370	-0.5165 0.5164
0	+/-100 arcsec	+/-100 arcsec	0.3675 -0.3681	0.9999 -0.9995	0.2435 0.2433
+/-100 arcsec	+/-100 arcsec	+/-100 arcsec	0.2520 -0.2531	1.2125 -1.2121	-0.5165 0.5164

Table 8. Actuator Error

Temperature ΔT	X=1731mm ΔX	Y=0 mm ΔY	Z=1125mm ΔZ
+/-1°	0.0021 -0.0021	-0.0036 0.0036	0.0121 0.0121
+/-3°	0.0063 -0.0063	-0.0107 0.0107	0.0362 -0.0362
+/-5°	0.0106 -0.0106	-0.0178 0.0178	0.0603 -0.0603

Table 9. Thermal Error

Error Budget			
Variable No.	Description	Name	Budget
1	drive 1	Joint 1	32 arcsec
2	drive 2	Joint 2	ar6 arcsec
3	drive 3	Joint 3	1.2 arcsec
17	joint 1 and arm 1	a1	1.62 um
24		d1	363 um
4		sit1	10.4 arcsec
10		afa1	110 arcsec
18	joint_link11_arm 1	a2	373 um
19	short arm 1	a3	174 um
25		d3	449 um
5		sit3	9.24 arcsec
11		afa3	9.45 arcsec
20	joint 2 and arm 2	a4	1.9 mm
26		d4	485 um
6		sit4	1.22 arcsec
12		afa4	38.5 arcsec
21	short arm 2	a5	430 um
27		d5	D
7		sit5	11.2 arcsec
13		afa5	D
22	joint 3	a6	0
28		d6	D
8		sit6	4.64 arcsec
14		afa6	D
23	arm 3	a7	0
29		d7	D
9		sit7	6.14 arcsec
15		afa7	D
30	joint_link11_arm1	x1	D
31		y1	43 um

32		z1	123 um
33	joint_link21_arm1	x2	D
34		y2	49.4 um
35		z2	D
36	joint_link31_arm1	x3	115 um
37		y3	108 um
38		z3	D
39	joint_link12_arm2	x4	D
40		y4	1.28 mm
41		z4	D
42	joint_link22_arm2	x5	2.6 mm
43		y5	68.2 um
44		z5	D
45	joint_link13_arm3	x6	D
46		y6	21.6 um
47		z6	213 um
48	joint_link11_platform	x11	50 um
49		y11	50 um
50		z11	D
51	joint_link31_platform	x22	50 um
52		y22	50 um
53		z22	D
54	joint_link21_platform	x33	50 um
55		y33	50 um
56		z33	13.3 um
57	joint_link12_platform	x44	50 um
58		y44	50 um
59		z44	37.9 um
60	joint_link22_platform	x55	50 um
61		y55	50 um
62		z55	398 um
63	joint-link13_platform	x66	50 um
64		y66	50 um
65		z66	50 um
16	height of the TCP	a	436 um
66	link 13	L0	0
67	link 11	L1	88 um
68	link 31	L2	151 um
69	link 21	L3	54.3 um
70	link 22	L4	213 um
71	link 12	L5	1.47 mm

Table 10. Error Budget (Assigned System Error = 50 um)

Parameter Number	Parameter Name	Measured Parameter Errors	Calibration Results with SVD (mm/deg)
1	theta1	3	-0.731778837
2	theta2		2.934613648
3	L1	0.246	-0.065823708
4	alpha1	/	0.005595871
5	beta1	/	0.009767543
6	a20	0.639	-0.600433798
7	alpha20	0.004297187	-0.054380834
8	L2	0.022	-0.652730647
9	alpha3	/	0.100085204
10	Rx	NA	0.237556976
11	Ry	NA	-0.297084061
12	Rz	NA	86.49124257
13	tx	NA	-61.06910063
14	ty	NA	-1934.277556
15	tz	NA	510.5174107
16	xpl	NA	22.96695136
17	ypl	NA	-56.41477281

	SVD	LM - Nonlinear optimization	Gauss Newton - Nonlinear optimization
Average Absolute Accuracy (mm)	0.11718325	0.11395309	0.11395309
Average Standard Deviation (MM)	0.04774522	0.04849159	0.04849159
Elapsed Time (s)	300	175	175

Table 11. Calibration Results of 2D Testing Bench

Error Parameter	Error Assigned (mm)	Error from Calibration (mm)
Arm3	0.02	0.019969
	0.05	0.049904
Link13	0.01	0.012201
	0.02	0.024424
Link12	0.02	0.018469
	0.05	0.046093

Table 12. Calibration Result of TAU Robot

6.3 Approach Comparison and Summary

The results discussed above indicate that a closed form forward kinematic solution can be computed and finished much faster than the conventional iterative algorithms. The closed form solution is very difficult to obtain because the problem is highly nonlinear. The N-R method (iterative method) can give an accurate result but it usually takes an average of 4290 multiplications and 630 sine functions for the iterative N-R algorithm. The Polynomial Based method needs at least to solve a 16th-order polynomial equation, which is slow and solution is with spurious roots.
The proposed JAM algorithm can eliminate these drawbacks, and it has an effective closed-form solution with an accuracy of 1.53um.
Table 13 below summarizes the features of the methods proposed by the author in solving the parallel robotics problems involved. The methodology and approach are also used in other robotics applications to effectively increase system modeling, control and process accuracy.

Approaches	Description	Drawback	Accuracy
Polynomial Based	Reduces the resulting constraint equations into a high-order polynomial by the method of elimination.	Requires extremely complicated formulation procedures and has been known to be much slower than the numerical iteration such as the N-R method.	**60-70 um** Lee, H. S. and Han, M. 1999 IEEE
Newton-Raphson (N-R) Numerical Iteration	Among several iterative methods, it has been wisely employed due to its property of convergence.	Jacobian matrix obtained **numerically**, which is not efficient, and has a great influence on the convergence of numerical method.	**0.06 um** Researched in this paper
Extra-Sensor	Reduces the numbers of unknown variable by extra-sensor	Same as the polynomial-based method. Complicated hardware setup	**750 um** Geng, Z. and Haynes, L. 1994
Jacobian Approximation Method (JAM)	Analytic solution of Jacobian matrix + N-R method (one iteration)	**Analytical** Jacobian matrix is difficult to obtain for large-scale MIMO non-linear system.	**1.6 um** Closed-form solution Proposed in this paper

Table 13. Approaches in Parallel Robot Forward Kinematic Modeling

7. Conclusions

The TAU robot represents a new configuration of parallel robots. This robotic configuration is well adapted to perform with a high precision and high stiffness within a large working space compared with a serial robot. It has the advantages of both parallel robots and serial robots.

In this study, the kinematic modeling and error modeling are established with all errors considered using Jacobian matrix method for the robot. Meanwhile, a very effective Jacobian Approximation Method is introduced to calculate the forward kinematic problem instead of Newton-Raphson iteration method. It denotes that a closed form solution can be obtained instead of a numerical iteration solution. A full size Jacobian matrix is used in carrying out error analysis, error budget, and model parameter estimation and identification. Simulation results indicate that both Jacobian matrix and Jacobian Approximation Method are correct and with a level of accuracy of micron meters. ADAMS's simulation results are used in verifying the established models. Experimental results obtained based on both the lab prototype and industrial prototype show that the established models enabled the realization of high precision for the new class of robots.

The established models are also used in the development of other precision robotics systems. Precision robotic machining processes using existing serial robots have been realized successfully with industry partners involved. These precision processes include robotic milling of aluminum engine blocks, and belt grinding of complicated parts of curved surfaces such as engine blades, and human knee joint replacements.

Based on the analytical Jacobian matrix solution, SVD calibration is carried out for three parameters that contribute to the final position error, the accuracy of calibration is within 4um for individual components.

In the milling application of engine block, the position compensation procedure is proved, which reduces the error to < 0.1 mm compared with the original error of 0.5 mm.

8. Future work: error minimization and design optimization of tau robot

8.1 Problem Statement

To further increase the accuracy and performance of a robotic system, error minimization and parameter optimization in the design space will be a powerful tool. There are two spaces involved: the error space with numerous error sources, and the design parameter space with numerous robotic parameters associated with the D-H model. The current practice in robotic design optimization is to solve one of the problems, often the later. It is very difficult to solve both parameter problem and the error problem at the same time.
Based on the prior work about the Error modeling and Sensitivity analysis of Jacobian matrix, Position errors can be obtained in X, Y and Z directions as well as Jacobian matrix (error sensitivities). With these parameters for some given error sources, It is important that how to adjust the other error parameters so that the minimum global error can be obtained in whole workspace.
By analyzing error sensitivity results, the sensitivities vary according to different positions in whole workspace, so the optimal results have to satisfy whole workspace. It is a powerful tool for industrial robot design and development that a method capable of optimizing design parameters in two kinds of different optimization spaces through establishing an optimization criteria in two different independent and relative design and configuration (movement) spaces.

8.2 Problem Formulation – Proposed Object Function and Constrain Function

For Tau robot, the global error function is not only the function of component sizes but also the function of robot positions, and the global error comes from three directions (X, Y and Z) so this multi-objects optimal problem can be transformed into single object problem then combined global error function can be written as object function as follows:

$$F(X_i,\theta_i) = \omega_x F_x + \omega_y F_y + \omega_z F_z \tag{71}$$

Where θ_i is the position variables of robot, $\omega_x, \omega_y, \omega_z$ are weight factors denoted by designer according to the error budget, F_x, F_y and F_z are errors of X, Y, and Z directions, respectively, and

$$F_x = (C_x + \sum S_{ix} X_i)^2 \tag{72}$$

Where $F_y = (C_y + \sum S_{iy} X_i)^2$

$F_z = (C_z + \sum S_{iz} X_i)^2$

Where C_x, C_y, C_z are given error sources caused by manufacturing and assembly, and S_{ix}, S_{iy}, S_{iz} are just component's sensitivities calculated before in X, Y and Z directions. Here one attention point is that all parameters Ci and Si are functions of position parameters θ_i. The constrain function is $X_{i\min} \le X_i \le X_{i\max}$
From above one can transform the constrained optimal problem into unconstrained optimal problem by using the Lagrange multiplier Method as follows:

$$F(X_i, \theta_i) = \omega_x F_x + \omega_y F_y + \omega_z F_z + \sum_{j=1}^{k} \lambda_j g_j(X_i) \tag{73}$$

Where $g_j(X_i)$ is the constrain function.

In constrained optimization, the general aim is to transform the problem into an easier sub-problem that can then be solved and used as the basis of an iterative process. A characteristic of a large class of early methods is the translation of the constrained problem to a basic unconstrained problem by using a penalty function for constraints, which are near or beyond the constraint boundary. In this way the constrained problem is solved using a sequence of parameterized unconstrained optimizations, which in the limit (of the sequence) converge to the constrained problem. These methods are now considered relatively inefficient and have been replaced by methods that have focused on the solution of the Kuhn-Tucker (KT) equations. The KT equations are necessary conditions for optimality for a constrained optimization problem. If the problem is a so-called convex programming problem, then the KT equations are both necessary and sufficient for a global solution.
So from the Equation (73) next Kuhn-Tucker conditions should be satisfied as:

$$\begin{aligned} &\nabla F(X^*, \theta^*) + \sum_{j=1}^{k} \lambda_j \nabla g_j(X^*) = 0 \\ &\lambda_j g_j(X^*) = 0 \qquad (j = 1,2,...,k) \\ &\lambda_j \ge 0 \qquad (j = 1,2,...k) \end{aligned} \tag{74}$$

For a given configuration, by using Sequential Quadratic Programming (SQP) method, Equation (74) can be solved.

8.3 Future Work - Optimization in the Whole Robotics Workspace

Since from mentioned above the object functions have to satisfy whole workspace. A most serious configuration (movement position) has to be found so that one can optimize the object function in this situation.

One needs to utilize the other method to optimize the object function since the relationship between the object function and position variable is not explicit. Here "Pattern Search method" is adopted to search the most serious position of robot that means to maximum the object function. The final optimal result can be obtained by using the Lagrange multiplier Method as the most serious position is known.

"Pattern Search method" is consisted of two steps 'move', one is exploratory move and other is module move. The former is to obtain the useful direction by calculating the variations of object function the later is to get a better new "point" instead of old "point" in the useful direction, which is similar to the gradient direction.

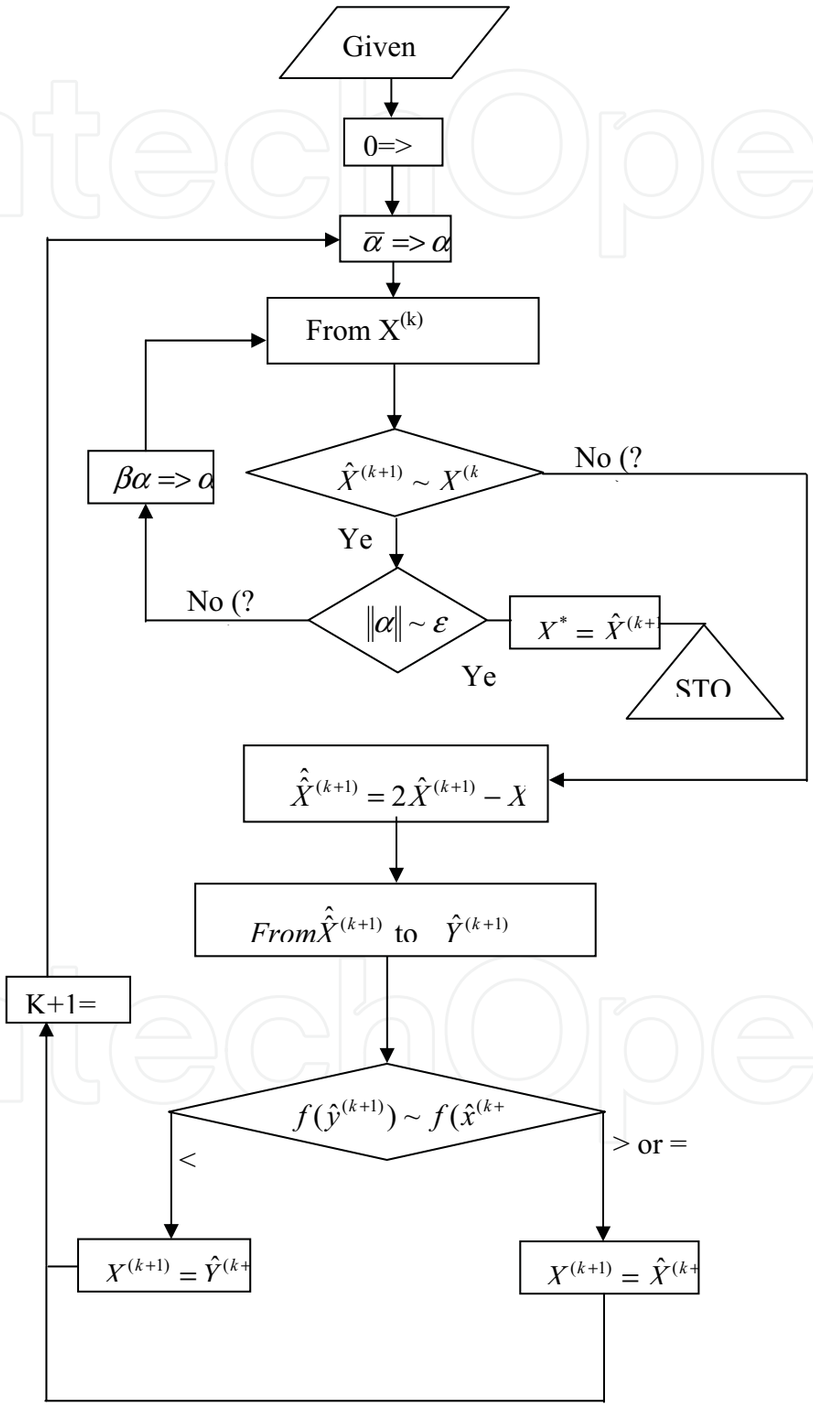
Description of Exploratory Move and Module Move:

Given initial point $X^{(0)}$, step length $\overline{\alpha} = (\overline{\alpha}_1, \overline{\alpha}_2, ..., \overline{\alpha}_n)^T$, $0 < \beta < 1$, error ε.

1. $0 \Rightarrow K$
2. Exploratory Move, $\overline{\alpha} \Rightarrow \alpha$
   - 2.1 $0 \Rightarrow i$, $X^{(k)} \Rightarrow \hat{X}^{(k,i)}$
   - 2.2 $\hat{X}^{(k,i)} + \alpha_{i+1} e_{i+1} \Rightarrow \widetilde{X}$, $\hat{X}^{(k,i)} - \alpha_{i+1} e_{i+1} \Rightarrow \overline{X}$, $\hat{X}^{(k,i)} \Rightarrow X$
   - 2.3 if $f(\widetilde{X}) < f(X)$, *then* $\widetilde{X} \Rightarrow \hat{X}^{(k,i+1)}$;
     if $f(\widetilde{X}) \geq f(X) > f(\overline{X})$, *then* $\overline{X} \Rightarrow \hat{X}^{(k,i+1)}$;
     if $f(\widetilde{X}) \leq f(X) \leq f(\overline{X})$, *then* $X \Rightarrow \hat{X}^{(k,i+1)}$
   - 2.4 $i+1 \Rightarrow i$
   - 2.5 if $i < n$, *then, do* 2.2; *if* $i = n$, *do* 3
3. if $\hat{X}^{(k,n)} \neq \hat{X}^{(k,0)}$, *then* $\hat{X}^{(k,n)} \Rightarrow \hat{X}^{(k+1)}$, *do* 5;
   if $\hat{X}^{(k,n)} = \hat{X}^{(k,0)}$, *then, do* 4.
4. if $\|\alpha\| \leq \varepsilon$, then solve the optimal solutions, $X^* = \hat{X}^{(k,n)}$. if $\|\alpha\| > \varepsilon$ and $\beta\alpha \Rightarrow \alpha$, go to 2.1
5. $2\hat{X}^{(k+1)} - X^{(k)} \Rightarrow \hat{\hat{X}}^{(k+1)}$, obtain $\hat{y}^{(k+1)}$ from $\hat{X}$ by exploratory move.
6. if $f(\hat{y}^{(k+1)}) < f(\hat{X}^{(k+1)})$, *then* $\hat{y}^{(k+1)} \Rightarrow X^{(k+1)}, k+1 \Rightarrow k$, go to 2;
   if $f(\hat{y}^{(k+1)}) \geq f(\hat{X}^{(k+1)})$, *then* $\hat{X}^{(k+1)} \Rightarrow X^{(k+1)}, k+1 \Rightarrow k$,
   $\beta\alpha \Rightarrow \alpha$ go to 2.1

So the calculation stops at the step 4.

Module move (Hooke-Jeeves) Flow Chart



9. References

Abderrahim, M. and Whittake, A. R., "Kinematic Model Identification of Industrial manipulators," Robotics and Computer Integrated Manufacturing 16 (2000), 1-8

Brogangrdh, T., "Design of high performance parallel arm robots for industrial applications," Proceedings of A symposium Commemorating the Legacy, Works, and Life of Sir Robert Stawell Ball Upon the 100th Anniversary of A Treatise on the Theory of Screws , University of Cambridge, Trinity College, July 9-11, 2000

Brogangrdh, T., et al, "Device for relative movement of two elements," United States Patent 6425303, July 30, 2002

Brogangrdh, T., et al, "Device for relative movement of two elements," United States Patent 6336374, January 8, 2002

Brogangrdh, T., et al, "Device for relative movement of two elements," United States Patent 6301988, October 16, 2001

Cui, H., Zhu, Z., Gan Z., and Brogardh, T., "Kinematics Analysis and Error Modeling of Tau Parallel Robot", Robotics and Computer Integrated Manufacturing: An International Journal, v 21, n 6, December, 2005, p 497-505

Denavit, J., Hartenberg, H., 1955, "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices", Journal of Applied Mechanics, pp. 215- 221

Dhingra, A. K., Almadi, A. N. and Kohli, D., "Closed-Form Displacement Analysis of 8, 9 and 10-Link Mechanisms, Part II" Mechanism and Machine Theory, 35, 2000, pp 851-869

Dhingra, A. K., Almadi,A. N. and Kohli, D., "A Grobner-Sylvester Hybrid Method For Closed-Form Displacement Analysis of Mechanisms," 1998 ASME Design Engineering Technical Conference, Atlanta, GA

Dhingra, A. K., Almadi,A. N. and Kohli, D., "Closed-Form Displacement Analysis of 8, 9 and 10-Link Mechanisms," Mechanism and Machine Theory, 35, 2000, pp 821-850

Didrit, Olivier; Petitot, Michel & Walter, Eric "Guaranteed Solution of Direct Kinematic Problems for General Configurations of Parallel Manipulators," IEEE Transactions on Robotics and Automation, Vol. 14, No.2, April 1998

Gong, Chunhe, Yuan, Jingxia, and Ni, J., "Nongeometric Error Identification and Compensation for Robotic System by Inverse Calibration," International Journal of Machine Tools & Manufacture 40 (2000) 2119-2137

Griffis,M. & Duffy, J. "A Forward Displacement Analysis of a Class of Stewart Platform," Journal of Robotic System 6 (6), 703-720 (1989) by John Wiley & Sons, Inc

Lee, hyung Sang and Myung-chul Han, "The estimation for forward kinematic solution of stewart platform using the neural network", Proceedings of the 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems

Lin,W.; Griffis, M.; & Duffy, J. "Forward Displacement Analyses of the 4-4 Stewart Platforms," Transaction of the ASME Vol. 114, September 1992, pp444-450

Patel, Amit J., Ehmann, K. F., "Calibration of a Hexapod Machine Tool Using a Redundant Leg," International Journal of Machine Tools & Manufacture 40 2000, 489-512

Prabjot Nanua, Kenneth J. Waldron, and Vasudeva Murthy, "Direct kinematic Solution of a Stewart platform," IEEE Transactions on Robotics and Automation, Vol. 6, No.4, August 1990

Raghavan, M., "The Stewart Platform of General Geometry Has 40 Configurations," ASME Journal of Mechanical Design, June 1993, Vol. 115, pp 277-282

Shi, Xiaolun; Fenton,R. G. "A Complete and General Solution to the Forward Kinematics Problem of Platform-Type Robotic Manipulators," IEEE, 1050-4729/ 1994, pp 3055-3062

Sreenivasan,S. V.; Waldron K. J. & Nanua,P. "Closed-Form Direct Displacement Analysis of a 6-6 Stewart Platform," Mech. Mach. Theory Vol. 29. No. 6, pp 855-864, 1994

Stewart, D., 1965, "A Platform with Six Degree-of-freedom", Proc, 1st. Mech. Eng. London, Volume 180, pp371-286

Tsai, L., 1999 "Robot Analysis-The Mechanics of Serial and Parallel Manipulators", John Wiley & Sons, Inc, ISBN: 0-471-32593-7

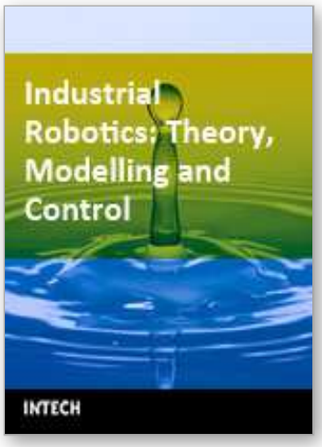
Wang, J. & Masory, O. "On the Accuracy of A Stewart Platform- Part I The Effect of Manufacturing Tolerances," IEEE, 1050-4729/ 1993, pp 114-120

Z. Jason Geng and Leonard S. Haynes, "A 3-2-1 Kinematic Configuration of a Stewart Platform and its Application to Six Degree of Freedom Pose Measurements," Robotics & Computer-Integrated Manufacture, Vol. 11, No. 1, pp23-34, 1994

Zhang, H., Wang, J., Zhang, G., Gan, Z., Pan, Z., Cui, H. and Zhu, Z. "Machining with flexible manipulator: Toward improving robotic machining performance", Proceedings of the 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM 2005, 2005, p 1127-1132

Zhang,Chang-de, Song, Shin-Min, "Forward Kinematics of a Class of Parallel (Stewart) Platform with Closed-Form Solutions," Proceedings of the 1991 IEEE International Conference on Robotics and Automation, Sacramento, California-April 1991

Zhu, Z.; Cui, H. "Six degree of freedom measuring system", Apr., 2003, U.S. patent (7040033)



Industrial Robotics: Theory, Modelling and Control

Edited by Sam Cubero

ISBN 3-86611-285-8

Hard cover, 964 pages

Publisher Pro Literatur Verlag, Germany / ARS, Austria

Published online 01, December, 2006

Published in print edition December, 2006

This book covers a wide range of topics relating to advanced industrial robotics, sensors and automation technologies. Although being highly technical and complex in nature, the papers presented in this book represent some of the latest cutting edge technologies and advancements in industrial robotics technology. This book covers topics such as networking, properties of manipulators, forward and inverse robot arm kinematics, motion path-planning, machine vision and many other practical topics too numerous to list here. The authors and editor of this book wish to inspire people, especially young ones, to get involved with robotic and mechatronic engineering technology and to develop new and exciting practical applications, perhaps using the ideas and concepts presented herein.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Hongliang Cui and Zhenqi Zhu (2006). Error Modeling and Accuracy of Parallel Industrial Robots, Industrial Robotics: Theory, Modelling and Control, Sam Cubero (Ed.), ISBN: 3-86611-285-8, InTech, Available from: http://www.intechopen.com/books/industrial_robotics_theory_modelling_and_control/error_modeling_and_accuracy_of_parallel_industrial_robots

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2006 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License, which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen