Kinematic Calibration Procedure for Serial Robots with Six Revolute Axes

M. J. D. HAYES, P. L. O'LEARY
Institut für Automation,
Montanuniversität Leoben
Peter-Tunner-Strasse 27, A-8700 Leoben, Austria
email: john.hayes@unileoben.ac.at

April 12, 2001

Abstract

In this paper an approach to the kinematic calibration of serial robots with six revolute axis in general, and the KUKA KR-15/2 in particular, is presented. We proceed by formulating how the pose of the robot (the position and orientation of the end effector) varies with the kinematic parameters which describe the robot geometry. Two models are presented, both based on the Denavit-Hartenberg parameters. The first can be used when measurements all six pose variables, or any sub-set are available. The second set can be used when only position measurements are available. The DH parameters are introduced in tutorial form before the calibration equations are derived. The Pose measurement is achieved by comparing relative linear robot motions to parallel standard ruled and flat straight edges. Images from a CCD camera mounted to the tool flange yield relative distance measurements along the length of the ruled straight edge while a stereo laser displacement sensor yields height measurements above the flat straight edge. The pose of a robot is specified by six independent position and orientation parameters. Because of this independence, subsets of these parameters can be used to formulate the system of calibration equations. Experimental results using the three x, y, z position coordinates are presented.

1 Introduction

Nothing can be manufactured so that nominally specified dimensions exactly match the actual ones. For a robot, this means the distances between the revolute axis centres and the relative orientation of the axes themselves will vary from nominal values. The result of these deviations is that there will be a difference between where the robot *thinks* it is and where it *really* is. For some robot tasks, such as spray painting or spot welding, this may not be too critical an issue. But, in applications where positions, or continuous trajectories are computed by a device external to that of the robot controler the deviations can be critical. Because striving for greater precision in machining and assembly is prohibitively costly, robot calibration has come to be thought of as necessary.

If the task permits, it may be that the robot can be implicitly calibrated with respect to the task itself, see [1] for example. If this is not possible, or practical, then a standard calibration procedure must be performed. We proceed by formulating how the end-effector (EE) pose varies with the Denavit-Hartenberg (DH) parameters. For kinematic calibration, the derivation of this relation begins with

$$\mathbf{x} \;=\; \mathbf{T}(\boldsymbol{\varphi}, \boldsymbol{\alpha}, \mathbf{a}, \mathbf{d}), \tag{1}$$

where $\mathbf{x}$ is the vector of pose variables (the x, y, z coordinates of an EE reference point and the ZYX Euler angles of the EE reference frame all with respect to the relatively fixed robot base frame, see Figure 1), $\mathbf{T}$ is the functional relationship depending upon the DH parameters: the vector of joint angles, $\boldsymbol{\varphi}$; the vector of skew angles between neighbouring joint axes, $\boldsymbol{\alpha}$; the vector of link lengths, $\mathbf{a}$; and the vector of joint offsets, $\mathbf{d}$. Variations of the EE pose with respect to the DH parameters can be determined by taking the first difference of the transformation $\mathbf{T}$ in Equation (1) with respect to the four DH parameters, giving

$$\Delta\mathbf{T} \;=\; \frac{\partial\mathbf{T}}{\partial\boldsymbol{\varphi}}\Delta\boldsymbol{\phi} + \frac{\partial\mathbf{T}}{\partial\boldsymbol{\alpha}}\Delta\boldsymbol{\alpha} + \frac{\partial\mathbf{T}}{\partial\mathbf{a}}\Delta\mathbf{a} + \frac{\partial\mathbf{T}}{\partial\mathbf{d}}\boldsymbol{\Delta}\mathbf{d}. \tag{2}$$

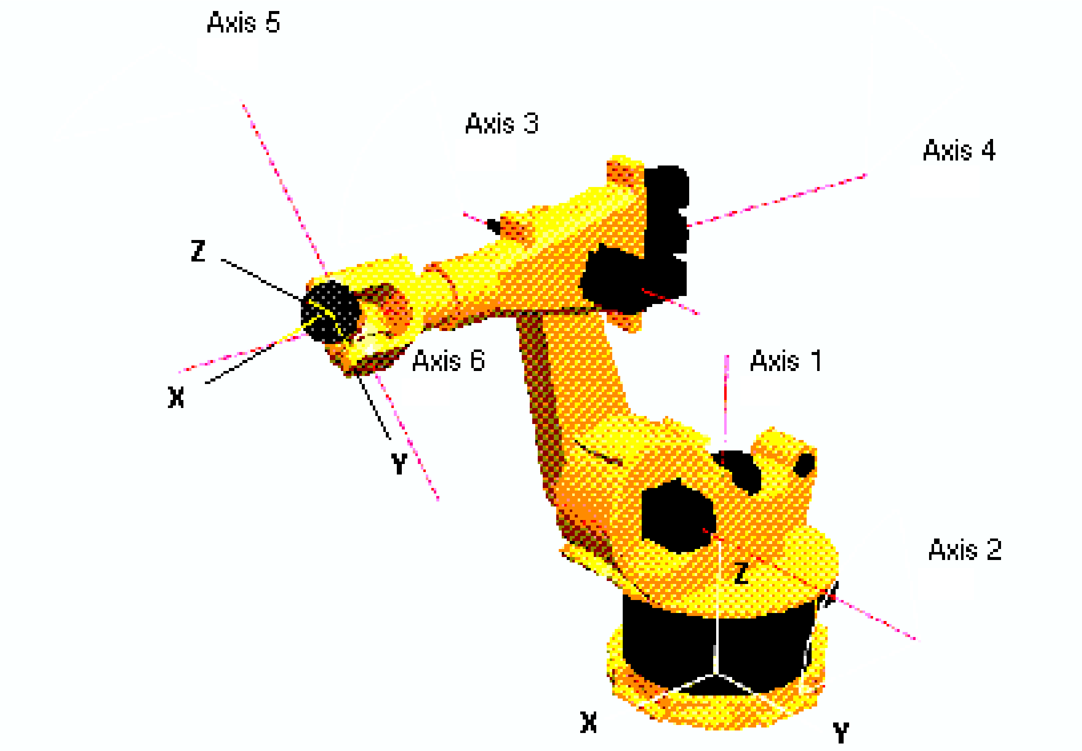

Figure 1: KUKA KR-15/2 six axis robot.

The partial derivatives are *Jacobians* with respect to the DH parameters. The differences $\Delta\boldsymbol{\varphi}$, $\Delta\boldsymbol{\alpha}$, $\Delta\mathbf{a}$ and $\Delta\mathbf{d}$ may be interpreted as the corrections to the DH parameters.

The corrections may be determined by first measuring *many*[1] poses of the robot in its workspace. Using Equation (2) a conventional Jacobian representation can be formulated as

$$\Delta\mathbf{x} \;=\; \mathbf{J}\boldsymbol{\Delta\rho}, \tag{3}$$

where $\Delta\boldsymbol{\rho}$ is the combined vector of DH parameter errors and $\mathbf{J}$ is the combined Jacobian matrix. For each measured pose, assuming the DH parameter errors to be constant, there will be one $\Delta\boldsymbol{\rho}$ vector and one $\mathbf{J}$ matrix. The difference $\Delta\mathbf{x}$ may be viewed as the error in the EE pose, obtained by subtracting the computed from the measured pose:

$$\Delta\mathbf{x} \;=\; \mathbf{x}_{\text{meas}} - \mathbf{x}_{\text{comp}}, \tag{4}$$

A suitably overconstrained set of Equations (3) can be solved, in a least-squares sense, for $\Delta\boldsymbol{\rho}$. Modification of this procedure for different kinematic models is easily accomplished. However, the most difficult aspect of kinematic calibration is obtaining accurate measurements of the EE pose.

2 Homogeneous Transformations

We will make extensive use of homogeneous coordinate transformations in the development of the calibration equations. A few words on their structure are warranted.

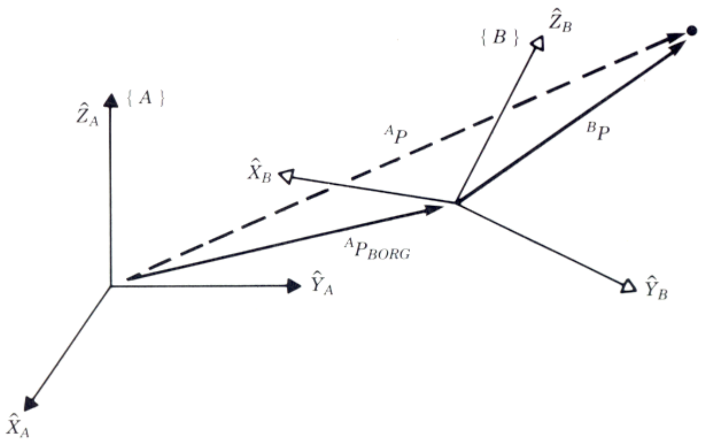

Figure 2: Coordinate transformation from $\{B\}$ to $\{A\}$.

The general case of mapping the coordinates of a point P represented in one coordinate reference frame $\{B\}$ to those of the same point in an arbitrarily displaced frame $\{A\}$ must account for the translation and rotation of $\{A\}$ with respect to $\{B\}$. Referring to Figure 2, we can first change the representation of the point in $\{B\}$, indicated by the position vector $^{B}\mathbf{p}$, to an intermediate frame with the same orientation as $\{A\}$, but whose

[1] *Many* in this context means enough measurements should be obtained to suitably over constrain the system of Equations (3).

coordinate origin is incident with $\{B\}$. Then account for the translation component of the displacement with simple vector addition:

$$^{A}\mathbf{p} \;=\; {}^{A}\mathbf{R}_{B}\,{}^{B}\mathbf{p} + {}^{A}\,\mathbf{p}_{BORG}, \tag{5}$$

where $^{A}\mathbf{p}$ is the position vector of P represented in $\{A\}$, $^{B}\mathbf{p}$ is the position vector of P represented in $\{B\}$, $^{A}\mathbf{R}_{B}$ is the rotation matrix changing the orientation from $\{B\}$ to $\{A\}$, and $^{A}\mathbf{p}_{BORG}$ is the position vector of the origin of reference frame $\{B\}$, O_B, represented in frame $\{A\}$.

Equation (5) describes a transformation of a position vector from its description in frame $\{B\}$ to its description in frame $\{A\}$. Our notation convention makes it easy to keep track of the transformation: it may be interpreted as the B's cancel-out. However, Equation (5) does not represent a linear transformation. This can lead to conceptual and computational problems if many such equations need to be concatenated to obtain a result. Additionally, its form is not as appealing as

$$^{A}\mathbf{p} \;=\; {}^{A}\mathbf{T}_{B}\,{}^{B}\mathbf{p}. \tag{6}$$

We can remedy this by combining the translation and rotation components, which are independent, into a single linear transformation:

$$\begin{bmatrix} 1 \\ ^{A}\mathbf{p} \end{bmatrix} \;=\; \begin{bmatrix} 1 & \mathbf{0} \\ ^{A}\mathbf{p}_{BORG} & ^{A}\mathbf{R}_{B} \end{bmatrix} \begin{bmatrix} 1 \\ ^{B}\mathbf{p} \end{bmatrix}. \tag{7}$$

The 4x4 matrix is called a homogeneous transform, $\mathbf{T}$. The term homogeneous refers to the fact that a *homogenising* coordinate has been introduced so that the matrix combining the rotation and translation is dimensionally compatible with the vectors. Hence, we write:

$$^{A}\mathbf{p} \;=\; {}^{A}\mathbf{T}_{B}\,{}^{B}\mathbf{p}. \tag{8}$$

Again, note that the convention of sub and super-scripts notation makes algebraic manipulations easy to follow. In general:

$$^{0}\mathbf{T}_{n} \;=\; {}^{0}\mathbf{T}_{1}\,{}^{1}\mathbf{T}_{2}\cdots{}^{n-1}\,\mathbf{T}_{n}. \tag{9}$$

3 Nominal Denavit-Hartenberg Parameters

Assuming the transformations used by the controller to determine pose data from joint encoder readings is known, the first step in the calibration procedure is to model the robot. Perhaps the most fundamental problem in describing the working environment of a robot is how to represent the relative positions and orientations of various components. The most common approach is to systematically assign coordinate reference frames to fixed and moving components in the workspace. From this, an algebraic model of the geometry of the environment can be constructed.

To construct robot kinematic and dynamic models the Denavit-Hartenberg parameters are usually employed. However, even these *standard* frames have two main variants in the literature [2, 3, 4]. Serial robots are typically modeled by considering them as *kinematic chains*. These are sets of rigid bodies, called *links*, coupled together by kinematic pairs, called *joints*. The joints each have only one degree-of-freedom (DOF). For a robot with n joints numbered 1 to n, there are $n+1$ links numbered 0 to n. Link 0 represents the relatively fixed base and link n carries the EE. Joint i connects links i and $i-1$.

Since each link is a rigid body its shape defines the relationship between the two neighbouring joint axes associated with it, see Figure 3. The DH parameters for the i^{th} link are its *length*, a_i, and *twist*, α_i. Each joint is also described by two DH parameters: the *link offset*, d_i, and the *joint angle*, ϑ_i.

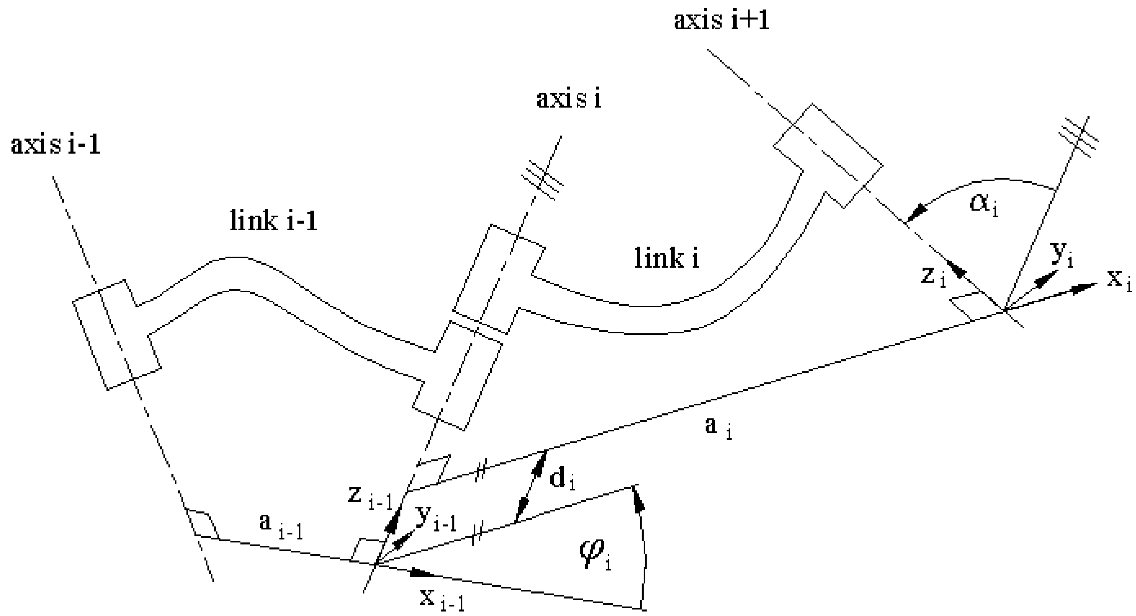

Figure 3: The DH parameters.

3.1 Conventions for Attaching Frames to Links

To use the DH parameters coordinate frames must be assigned in a systematic way, although there are two main conventions for doing so. This stems from the fact that the DH parameters for the first and last links are arbitrary. The two main philosophies for assigning the reference frames origins are:

1. The origin of frame i, O_i, is located on the axis of joint $i+1$ [5, 2, 4, 6, 7]. It is usually called the *Denavit-Hartenberg* (DH) form.

2. The origin of frame i, O_i, is located on the axis of joint i. It has been referred to as the *modified Denavit-Hartenberg* (MDH) form [3, 4, 6].

The DH form is suited to the kinematic calibration formulation, while the MDH form is, because of the mixed indices, better suited to deriving the equations of motion using an itterative approach [3].

Slight deviations from parallelism between two consecutive parallel axes can cause the kinematic calibration closure equations to be ill conditioned. The modification to the DH form proposed by Hayati [8] can then be used. We will call this set the *nearly parallel axes*, or NPA-DH form. Additionally, the DH and NPA-DH forms can be combined [9] into a single, general set called the *general* DH (GDH) form. These forms, with the exception of the MDH, will be briefly discussed in the following.

3.1.1 The DH Form

Item 1 in Section 3.1 above is the convention originally put forward in 1955 by Denavit and Hartenberg [5]. With reference to Figure 3, the details for assigning coordinate origin and axes for link i are as follows:

1. Identify all the joint axes. Consider the neighbouring ones, $i-1$, i and $i+1$.
2. Identify the common perpendicular between the two axes i and $i+1$, or their point of intersection. At the point of intersection, or where the common perpendicular meets the $(i+1)^{th}$ axis, assign the link frame origin, O_i.
3. For $\{0\}$ and $\{1\}$, ensure the axes are aligned when $\varphi_1 = 0$.
4. For $\{n-1\}$ and $\{n\}$, ensure the axes are aligned when $\varphi_n = 0$.
5. Assign the z_i axis to point along joint axis $i+1$. It is located at the distal end of link i connecting it to link $i+1$.
6. Assign the x_i axis to point along the common normal between joint axes $i-1$ and i. If the axes are parallel, any convenient normal can be selected. If the axes intersect, set x_i normal to the plane containing z_{i-1} and z_i.
7. Assign the y_i-axis to complete a right-handed coordinate system.

Note, the frame assignment is not necessarily unique. For instance, when the z_i-axis is aligned with the $(i+1)^{th}$ joint axis there is a choice of direction in which z_i points. As long as the procedure is strictly adhered to the uniqueness issue has no effect on the kinematics. That is, the resulting transformation matrices will always yield geometrically correct results with respect to the assigned reference frames.

3.2 First and Last Links: Base and Tool Frames

The first and last links in the robot kinematic chain require special attention. For the first link (the ground), there is no $i-1$ axis, and for the last link there is no $i+1$ axis. We can attach frame $\{0\}$ to the relatively non-moving robot base, link 0. In the case of the KUKA KR-15/2 we have positioned it so the origin is on the intersection of axis 1 with the plane of the floor. Additionally, we must have that when $\varphi_1 = 0$, then x_0 and

x_1 both have the same direction, and because of the assignment rules both intersect axis 1, but separated by a distance d_1. Moreover the origins of $\{0\}$ and $\{1\}$ are offset by a distance a_1.

Frame $\{n-1\}$ is attached last joint axis. We therefore need one more frame, connected to the last axis, to account for tool offsets. This frame is also selected so that the x_n-axis aligns with the x_{n-1}-axis.

3.3 DH Parameter Definitions

The four DH parameters describe the relative positions and orientations of the links and relative positions of the joints. Refering to Figure 3, they are:

- φ_i *joint angle*: the angle from x_{i-1} to x_i measured about z_{i-1}.
- α_i *link twist*: the angle from z_{i-1} to z_i measured about x_i.
- a_i *link length*: the distance from z_{i-1} to z_i measured along x_i.
- d_i *link offset*: the distance from x_{i-1}, to x_i measured along z_{i-1}.

Figure 4 shows the DH parameter assignment for the Mitsubishi RV-M2 and the KUKA KR-15/2.

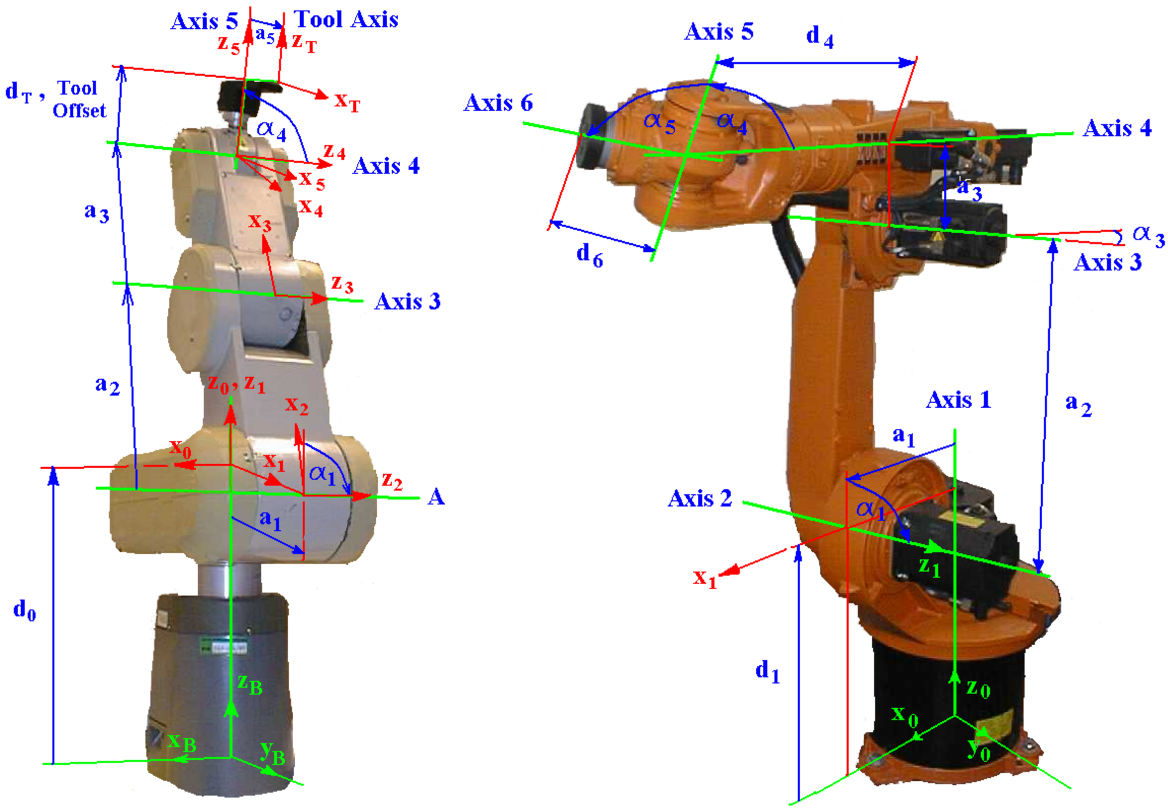

Figure 4: DH parameters assigned to the Mitsubishi RV-M2 and the KUKA KR-15/2.

Now we need a functional relationship that relates the pose of reference frame i to frame $i-1$ in terms of the DH parameters. This function can be thought of as a transformation that changes the coordinates of points in frame i to those of the same points in frame $i-1$, ${}^{i-1}\mathbf{T}_i$. Thus, for an n axis arm we have divided the forward kinematics problem into n sub-problems, namely the ${}^{i-1}\mathbf{T}_i$. To derive this transformation, we can further divide each of the n transformations into four sub-sub-problems. Each of these four transformations will be a function of only one link parameter and will be simple enough to write down *by inspection*.

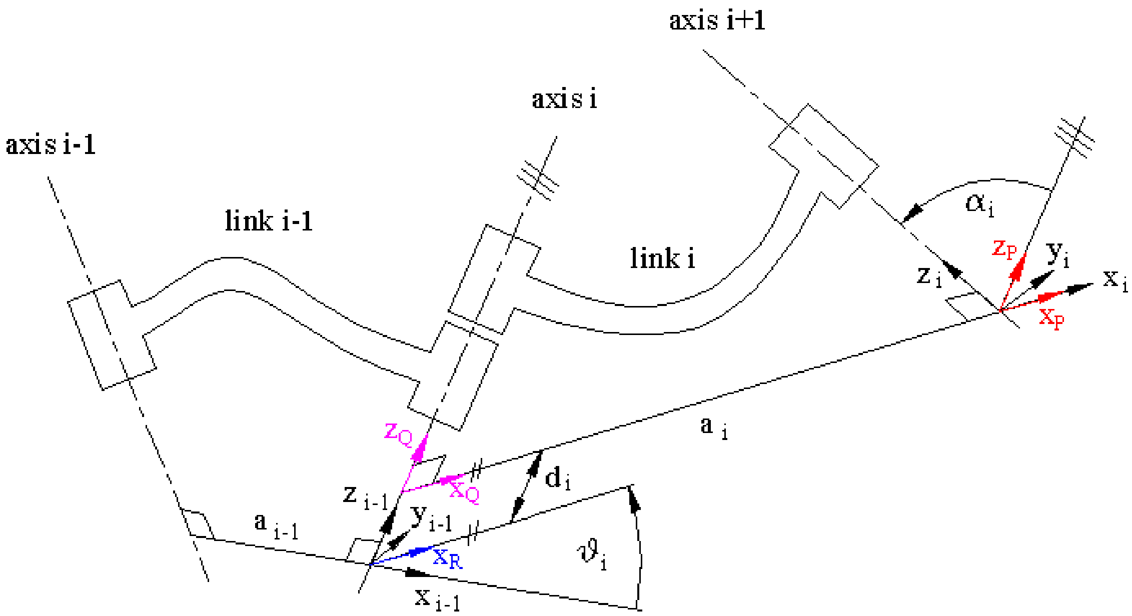

Figure 5: The intermediate reference frames $\{P\}$, $\{Q\}$ and $\{R\}$.

We begin by defining three intermediate frames for each link: $\{P\}$, $\{Q\}$ and $\{R\}$, see Figure 5. Transforming coordinates from i to $i-1$ may be thought of as a composition of the following homogeneous displacement transformations:

${}^{i-1}\mathbf{T}_R$:	rotate about z_R by φ_i.
${}^{R}\mathbf{T}_Q$:	translate along z_Q (z_{i-1}) a distance d_i.
${}^{Q}\mathbf{T}_P$:	translate along the x_p (x_i) a distance a_i.
${}^{P}\mathbf{T}_i$:	rotate about x_i by α_i.

This gives

$$ {}^{i-1}\mathbf{T}_i \;=\; {}^{i-1}\mathbf{T}_R \, {}^{R}\mathbf{T}_Q \, {}^{Q}\mathbf{T}_P \, {}^{P}\mathbf{T}_i, \qquad (10) $$

or explicitly

$$
{}^{i-1}\mathbf{T}_i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\varphi_i & -s\varphi_i & 0 \\ 0 & s\varphi_i & c\varphi_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \\ d_i & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ a_i & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & c\alpha_i & -s\alpha_i \\ 0 & 0 & s\alpha_i & c\alpha_i \end{bmatrix}, \quad (11)
$$

where $c\varphi_i$ means $\cos(\varphi_i)$, $s\varphi_i$ means $\sin(\varphi_i)$ with φ_i being the angle the x_i-axis makes relative to the x_{i-1}-axis. Multiplying gives

$$
{}^{i-1}\mathbf{T}_i \quad = \quad \begin{bmatrix} 1 & 0 & 0 & 0 \\ a_i c\varphi_i & c\varphi_i & -s\varphi_i c\alpha_i & s\varphi_i s\alpha_i \\ a_i s\varphi_i & s\varphi_i & c\varphi_i c\alpha_i & -c\varphi_i s\alpha_i \\ d_i & 0 & s\alpha_i & c\alpha_i \end{bmatrix}. \quad (12)
$$

For a serial arm with n joints there will be n such transformations. It is these DH transformations that determine the functional relationship mapping the EE pose to the DH parameters. Transforming the coordinates from frame n to the fixed base frame 0 is achieved by multiplying together the individual joint transforms. In general we have

$$
{}^{0}\mathbf{T}_n \quad = \quad {}^{0}\mathbf{T}_1 \, {}^{1}\mathbf{T}_2 \cdots {}^{n-2}\mathbf{T}_{n-1} \, {}^{n-1}\mathbf{T}_n. \quad (13)
$$

For the KUKA KR-15/2 we have assigned the the parameters listed in Table 1, which correspond to those illustrated in Figure 4.

i	φ_i	α_i (deg.)	a_i (m)	d_i (m)
1	φ_1	90	0.300	0.675
2	φ_2	0	0.650	0
3	φ_3	90	0.155	0
4	φ_4	-90	0	0.600
5	φ_5	90	0	0
6	φ_6	0	0	0.140

Table 1: DH parameter assignments.

3.3.1 The NPA-DH Form

The following modification of the DH parameters, proposed by Hayati [8], address the issues associated with deviation from parallelism. Rather than using the common normal between adjacent axes, a plane perpendicular to axis i contains the origin of frame $i-1$, O_{i-1}, is defined, see Figure 6. The intersection of this plane with joint axis $i+1$ defines the origin of frame i, O_i. The line between O_{i-1} and O_i defines the direction of x_i. The z_i axis lies along joint axis $i+1$.

The four NPA-DH parameters are:

φ_i	*joint angle*: the angle from x_{i-1} to x_i measured about z_{i-1}.
α_i	*link x-twist*: the angle from z_{i-1} to z_i measured about x_i.
β_i	*link y-twist*: the angle from z_{i-1} to z_i measured about y_i.
a_i	*link length*: the distance from z_{i-1} to z_i measured along x_i.

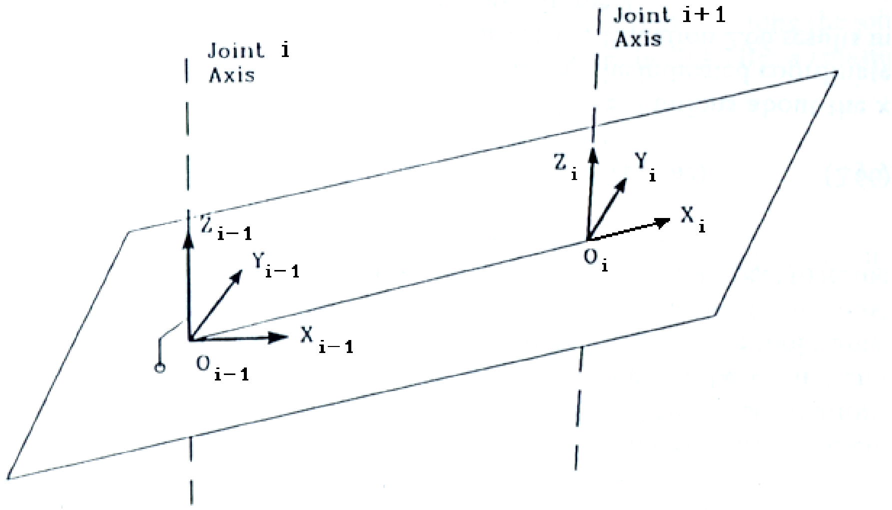

Figure 6: The NPA-DH parameters.

The transformation between adjacent reference frames can be derived in a way similar to that for the DH form. Transforming coordinates from i to $i-1$ may be thought of as a composition of the following:

${}^{i-1}\mathbf{T}_R$:	rotate about z_{i-1} by φ_i.
${}^{R}\mathbf{T}_Q$:	translate along x_{i-1} a distance a_i.
${}^{Q}\mathbf{T}_P$:	rotate about x_i by α_i.
${}^{P}\mathbf{T}_i$:	rotate about x_i by β_i.

When multiplied out, the above transformations yield

$$
{}^{i-1}\mathbf{T}_i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ a_i c\varphi_i & -s\alpha_i s\beta_i s\varphi_i + c\beta_i c\varphi_i & -s\varphi_i c\alpha_i & s\alpha_i c\beta_i s\varphi_i + s\beta_i c\varphi_i \\ a_i s\varphi_i & s\alpha_i s\beta_i c\varphi_i + c\beta_i s\varphi_i & c\varphi_i c\alpha_i & -s\alpha_i c\beta_i c\varphi_i + s\beta_i s\varphi_i \\ 0 & -c\alpha_i s\beta_i & s\alpha_i & c\alpha_i c\beta_i \end{bmatrix}. \quad (14)
$$

3.3.2 The GDH Form

The DH and NPA-DH forms can be combined into a single form that is immune to the numerical problem associated with the DH form. The GDH form has the advantage that only one form is required for a serial robot with revolute joints. Thus, the GDH parameters are

φ_i	*joint angle*: the angle from x_{i-1} to x_i measured about z_{i-1}.
α_i	*link x-twist*: the angle from z_{i-1} to z_i measured about x_i.
β_i	*link y-twist*: the angle from z_{i-1} to z_i measured about y_i.
a_i	*link length*: the distance from z_{i-1} to z_i measured along x_i.
d_i	*link offset*: the distance from x_{i-1} to x_i measured along z_{i-1}.

The GDH transformation is

$$ {}^{i-1}\mathbf{T}_i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ a_i c\varphi_i & -s\alpha_i s\beta_i s\varphi_i + c\beta_i c\varphi_i & -s\varphi_i c\alpha_i & s\alpha_i c\beta_i s\varphi_i + s\beta_i c\varphi_i \\ a_i s\varphi_i & s\alpha_i s\beta_i c\varphi_i + c\beta_i s\varphi_i & c\varphi_i c\alpha_i & -s\alpha_i c\beta_i c\varphi_i + s\beta_i s\varphi_i \\ d_i c\beta_i & -c\alpha_i s\beta_i & s\alpha_i & c\alpha_i c\beta_i \end{bmatrix}. \tag{15} $$

Note that when $\beta_i = 0$ the DH form is recovered, when $d_i = 0$ the NPA-DH form is recovered.

4 Kinematic Error Model for a Single Joint

After selecting an appropriate kinematic model, we can formulate the difference equations needed in the parameter identification problem, as in Equation (2). The aim is to ultimately use the GDH form, however for these preliminary stages in the development of our calibration procedure we shall use only the DH form. Moreover, no stochastic measures will be employed. Only when the procedure is up-and-running will more sophistication be added.

Recall that variations of the coordinate transformation between coordinate frames attached to neighbouring links with respect to the DH parameters can be determined by taking the first difference of $\mathbf{T}$ from Equation (1) with respect to the four DH parameters, giving Equation (2), which is reproduced below for for a single joint relating link i to $i-1$:

$$ \Delta\, {}^{i-1}\mathbf{T}_i = \frac{\partial\, {}^{i-1}\mathbf{T}_i}{\partial \varphi_i}\Delta\varphi_i + \frac{\partial\, {}^{i-1}\mathbf{T}_i}{\partial \alpha_i}\Delta\alpha_i + \frac{\partial\, {}^{i-1}\mathbf{T}_i}{\partial a_i}\Delta a_i + \frac{\partial\, {}^{i-1}\mathbf{T}_i}{\partial d_i}\Delta d_i. \tag{16} $$

Differentiating Equation (12) with respect to φ_i yields the first term in Equation (16). We then set this derivative equal to the product of the original DH transformation with another:

$$ \frac{\partial\, {}^{i-1}\mathbf{T}_i}{\partial \varphi_i} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ -a_i s\varphi_i & -s\varphi_i & -c\varphi_i c\alpha_i & c\varphi_i s\alpha_i \\ a_i c\varphi_i & c\varphi_i & -s\varphi_i c\alpha_i & s\varphi_i s\alpha_i \\ 0 & 0 & 0 & 0 \end{bmatrix} = {}^{i-1}\mathbf{T}_i \mathbf{Q}_{\varphi_i}. \tag{17} $$

An expression for the matrix $\mathbf{Q}_{\varphi_i}$ comes from

$$ {}^{i-1}\mathbf{T}_i \mathbf{Q}_{\varphi_i} = \frac{\partial\, {}^{i-1}\mathbf{T}_i}{\partial \varphi_i} \Rightarrow \mathbf{Q}_{\varphi_i} = {}^{i-1}\mathbf{T}_i^{-1} \frac{\partial\, {}^{i-1}\mathbf{T}_i}{\partial \varphi_i}. \tag{18} $$

We obtain

$$ \mathbf{Q}_{\varphi_i} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -c\alpha_i & s\alpha_i \\ a_i c\alpha_i & c\alpha_i & 0 & 0 \\ -a_i s\alpha_i & -s\alpha_i & 0 & 0 \end{bmatrix}. \tag{19} $$

For the other three DH parameters, we similarly obtain

$$\mathbf{Q}_{\alpha_i} \;=\; {}^{i-1}\mathbf{T}_i^{-1}\,\frac{\partial\,{}^{i-1}\mathbf{T}_i}{\partial \alpha_i} \;=\; \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{20}$$

$$\mathbf{Q}_{a_i} \;=\; {}^{i-1}\mathbf{T}_i^{-1}\,\frac{\partial\,{}^{i-1}\mathbf{T}_i}{\partial a_i} \;=\; \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{21}$$

$$\mathbf{Q}_{d_i} \;=\; {}^{i-1}\mathbf{T}_i^{-1}\,\frac{\partial\,{}^{i-1}\mathbf{T}_i}{\partial d_i} \;=\; \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ s\alpha_i & 0 & 0 & 0 \\ c\alpha_i & 0 & 0 & 0 \end{bmatrix}, \tag{22}$$

Substituting the results of Equations (19)-(22) back into Equation (16) gives

$$\Delta^{i-1}\mathbf{T}_i \;=\; {}^{i-1}\mathbf{T}_i(\mathbf{Q}_{\varphi_i}\Delta\varphi_i + \mathbf{Q}_{\alpha_i}\Delta\alpha_i + \mathbf{Q}_{a_i}\Delta a_i + \mathbf{Q}_{d_i}\,\Delta d_i). \tag{23}$$

Proceeding according to Mooring, *et al* [4], we define the error matrix $\delta\mathbf{T}$ such that

$$\Delta^{i-1}\mathbf{T}_i \;=\; {}^{i-1}\mathbf{T}_i\delta^{i-1}\mathbf{T}_i, \tag{24}$$

which means

$$\delta^{i-1}\mathbf{T}_i = \mathbf{Q}_{\varphi_i}\Delta\varphi_i + \mathbf{Q}_{\alpha_i}\Delta\alpha_i + \mathbf{Q}_{a_i}\Delta a_i + \mathbf{Q}_{d_i}\Delta d_i. \tag{25}$$

Expanding Equation (25) we obtain the structure for the error matrix:

$$\delta^{i-1}\mathbf{T}_i \;=\; \begin{bmatrix} 0 & 0 & 0 & 0 \\ \Delta a_i & 0 & -c\alpha_i\Delta\varphi_i & s\alpha_i\Delta\varphi_i \\ a_i c\alpha_i\Delta\varphi_i + s\alpha_i\Delta d_i & c\alpha_i\Delta\varphi_i & 0 & -\Delta\alpha_i \\ -a_i s\alpha_i\Delta\varphi_i + c\alpha_i\Delta d_i & -s\alpha_i\Delta\varphi_i & \Delta\alpha_i & 0 \end{bmatrix}. \tag{26}$$

In essence, the error matrix $\delta^{i-1}\mathbf{T}_i$ represents a *velocity* matrix, as found in [3, 7]. This is because it represents a constrained motion from the nominal pose to the actual pose that occurs in the limit as the change in time tends to zero. As such, the proper orthonormal rotation matrix embedded in ${}^{i-1}\mathbf{T}_i$ has been differentiated once with respect to time. What is the time derivative of a proper orthonormal matrix? Suppose we have a proper orthonormal matrix $\mathbf{P}$. We know, by definition, that

$$\mathbf{P}\mathbf{P}^T \;=\; \mathbf{P}\mathbf{P}^{-1} \;=\; \mathbf{I}, \tag{27}$$

the product of $\mathbf{P}$ and its transpose yields the identity matrix of same dimension as $\mathbf{P}$. Differentiating Equation (27) with respect to time gives

$$\mathbf{P}\dot{\mathbf{P}}^T + \dot{\mathbf{P}}\mathbf{P}^T \;=\; \mathbf{0}. \tag{28}$$

Equation (28) may be rearranged, yielding

$$\mathbf{P}\dot{\mathbf{P}}^T + (\mathbf{P}\dot{\mathbf{P}}^T)^T = \mathbf{0}. \tag{29}$$

Let's define the product $\mathbf{P}\dot{\mathbf{P}}^T \equiv \boldsymbol{\Omega}$ and substitute it back into Equation (29):

$$\boldsymbol{\Omega} + \boldsymbol{\Omega}^T = \mathbf{0}. \tag{30}$$

The quantity $\boldsymbol{\Omega}$ in Equation (30) is, by definition, a skew symmetric matrix. Furthermore, by the definition of $\boldsymbol{\Omega}$ we immediately have

$$\dot{\mathbf{P}} = \mathbf{P}\boldsymbol{\Omega}^T. \tag{31}$$

Because Equation (26) is a differential displacement transformation the rotation and translation errors are embedded in it. The rotation component, $\mathbf{R}$, is in general skew symmetric [10]. It must have the same form as $\boldsymbol{\Omega}$:

$$\boldsymbol{\Omega} \Rightarrow \delta^{i-1}\mathbf{R}_i = \begin{bmatrix} 0 & -\delta_z & \delta_y \\ \delta_z & 0 & -\delta_x \\ -\delta_y & \delta_x & 0 \end{bmatrix}. \tag{32}$$

Hence, we compose the orientation error vector, $\boldsymbol{\delta}$, of the orienting errors with respect to the robot base coordinate axes x, y and z, respectively. This gives for the ith coordinate transformation

$$\begin{aligned} \boldsymbol{\delta}_i &= \begin{bmatrix} \Delta\alpha_i \\ s\alpha_i\Delta\varphi_i \\ c\alpha_i\Delta\varphi_i \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ s\alpha_i \\ c\alpha_i \end{bmatrix} \Delta\varphi_i + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \Delta\alpha_i. \end{aligned} \tag{33}$$

Moreover, the displacement error, $\mathbf{d}_i$, with respect to the robot base coordinate axes x, y and z, respectively is extracted from the first column of Equation (26):

$$\begin{aligned} \mathbf{d}_i &= \begin{bmatrix} \Delta a_i \\ a_i c\alpha_i\Delta\varphi_i + s\alpha_i\Delta d_i \\ -a_i s\alpha_i\Delta\varphi_i + c\alpha_i\Delta d_i \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ a_i c\alpha_i \\ -a_i s\alpha_i \end{bmatrix} \Delta\varphi_i + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \Delta a_i + \begin{bmatrix} 0 \\ s\alpha_i \\ c\alpha_i \end{bmatrix} \Delta d_i. \end{aligned} \tag{34}$$

Thus, the robot pose error vector is the concatenation of $\mathbf{d}_i$ and $\boldsymbol{\delta}_i$:

$$\Delta\mathbf{x}_i = \begin{bmatrix} \mathbf{d}_i \\ \boldsymbol{\delta}_i \end{bmatrix}. \tag{35}$$

The pose error, $\Delta\mathbf{x}_i$, is the vector associated with the linear and skew symmetric rotation components of the error matrix, $\delta\,^{i-1}\mathbf{T}_i$. It is due to the errors in the DH parameters which describe the kinematic geometry of links $i-1$ and i coupled by joint i. It can be expressed by the single linear equation

$$\Delta \mathbf{x}_i = \begin{bmatrix} \mathbf{d}_i \\ \boldsymbol{\delta}_i \end{bmatrix} = \begin{bmatrix} \mathbf{k}_i^1 \Delta\varphi_i + \mathbf{k}_i^2 \Delta a_i + \mathbf{k}_i^3 \Delta d_i \\ \mathbf{k}_i^3 \Delta\varphi_i + \mathbf{k}_i^2 \Delta\alpha_i \end{bmatrix}, \tag{36}$$

where

$$\mathbf{k}_i^1 = \begin{bmatrix} 0 \\ a_i c\alpha_i \\ -a_i s\alpha_i \end{bmatrix}, \quad \mathbf{k}_i^2 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{k}_i^3 = \begin{bmatrix} 0 \\ s\alpha_i \\ c\alpha_i \end{bmatrix}. \tag{37}$$

Note the superscripts are indices and not exponents.

5 Kinematic Error Model for the Entire Robot

The next aim is to generalize the preceding analysis to the entire robot. That is, to express the positioning and orienting errors at the EE frame, relative to the fixed base frame, in terms of the DH parameters. For a robot with n joints the kinematic error model will be composed of at least $4n$ unknown DH error parameters. We can express the deviation from the computed DH transform using the *additive differential transformation*, $\Delta\mathbf{T}$, of $\mathbf{T}$ [10], as derived in the last section for a single joint:

$$\begin{aligned} {}^0\mathbf{T}_n + \Delta {}^0\mathbf{T}_n &= ({}^0\mathbf{T}_1 + \Delta {}^0\mathbf{T}_1)({}^1\mathbf{T}_2 + \Delta {}^1\mathbf{T}_2) \cdots ({}^{n-1}\mathbf{T}_n + \Delta {}^{n-1}\mathbf{T}_n) \\ &= \prod_{i=1}^{n} ({}^{i-1}\mathbf{T}_i + \Delta {}^{i-1}\mathbf{T}_i). \end{aligned} \tag{38}$$

Expanding Equation (38), ignoring second order products, yields

$${}^0\mathbf{T}_n + \Delta {}^0\mathbf{T}_n = {}^0\mathbf{T}_n + \sum_{i=1}^{n} ({}^0\mathbf{T}_1 \cdots {}^{i-2}\mathbf{T}_{i-1}) \Delta {}^{i-1}\mathbf{T}_i ({}^i\mathbf{T}_{i+1} \cdots {}^{n-1}\mathbf{T}_n). \tag{39}$$

Next, substitute the relation

$$\sum_{i=1}^{n} {}^0\mathbf{T}_n ({}^i\mathbf{T}_{i+1} \cdots {}^{n-1}\mathbf{T}_n)^{-1} = \sum_{i=1}^{n} ({}^0\mathbf{T}_1 \cdots {}^{i-1}\mathbf{T}_i), \tag{40}$$

together with Equation (24) into Equation (39):

$$\Delta {}^0\mathbf{T}_n = \sum_{i=1}^{n} {}^0\mathbf{T}_n ({}^i\mathbf{T}_{i+1} \cdots {}^{n-1}\mathbf{T}_n)^{-1} \delta^{i-1}\mathbf{T}_i ({}^i\mathbf{T}_{i+1} \cdots {}^{n-1}\mathbf{T}_n). \tag{41}$$

Now, define the matrix $\mathbf{U}_k$ to be the product of the $\mathbf{T}$ matrices from k to n, the current joint to the end of the manipulator:

$$\mathbf{U}_k = \prod_{i=k}^{n} {}^{i-1}\mathbf{T}_i, \tag{42}$$

we can write

$$\Delta^0\mathbf{T}_n = {}^0\mathbf{T}_n\left(\sum_{i=1}^{n}\mathbf{U}_{i+1}^{-1}\,\delta^{i-1}\mathbf{T}_i\mathbf{U}_{i+1}\right). \tag{43}$$

Recalling Equation (24), it is evident that

$$\delta^0\mathbf{T}_n = \sum_{i=1}^{n}\mathbf{U}_{i+1}^{-1}\,\delta^{i-1}\mathbf{T}_i\mathbf{U}_{i+1}, \tag{44}$$

for the index $i = n$, $\mathbf{U}_{n+1}$ is defined to be the identity matrix, $\mathbf{U}_{n+1} \equiv \mathbf{I}$.

Comparing Equations (24), (31) and (43) we must conclude that the rotation component of $\delta^0\mathbf{T}_n$ is skew symmetric and has the following form [4]:

$$\delta^0\mathbf{T}_n = \begin{bmatrix} 0 & 0 & 0 & 0 \\ d_x & 0 & -\delta_z & \delta_y \\ d_y & \delta_z & 0 & -\delta_x \\ d_z & -\delta_y & \delta_x & 0 \end{bmatrix}. \tag{45}$$

This follows from the fact that $\delta^0\mathbf{T}_n$ is a differential displacement transformation.

Rewriting the general form of Equation (24) as

$${}^0\mathbf{T}_n^{-1}\Delta^0\mathbf{T}_n = \delta^0\mathbf{T}_n, \tag{46}$$

we can now equate pose errors to the quantity ${}^0\mathbf{T}_n^{-1}\Delta^0\mathbf{T}_n$. It is known that [4]

$${}^0\mathbf{T}_n^{-1}\Delta^0\mathbf{T}_n = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \mathbf{n}\cdot\mathbf{d} + (\mathbf{p}\times\mathbf{n})\cdot\boldsymbol{\delta} & 0 & -\boldsymbol{\delta}\cdot(\mathbf{n}\times\mathbf{o}) & \boldsymbol{\delta}\cdot(\mathbf{a}\times\mathbf{n}) \\ \mathbf{o}\cdot\mathbf{d} + (\mathbf{p}\times\mathbf{o})\cdot\boldsymbol{\delta} & \boldsymbol{\delta}\cdot(\mathbf{n}\times\mathbf{o}) & 0 & \boldsymbol{\delta}\cdot(\mathbf{o}\times\mathbf{a}) \\ \mathbf{a}\cdot\mathbf{d} + (\mathbf{p}\times\mathbf{a})\cdot\boldsymbol{\delta} & -\boldsymbol{\delta}\cdot(\mathbf{a}\times\mathbf{n}) & \boldsymbol{\delta}\cdot(\mathbf{o}\times\mathbf{a}) & 0 \end{bmatrix}, \tag{47}$$

where $\mathbf{n}$, $\mathbf{o}$ and $\mathbf{a}$ are orthogonal unit vectors. Because of their orthogonality

$$\begin{aligned} \mathbf{o}\times\mathbf{a} &= \mathbf{n}, \\ \mathbf{a}\times\mathbf{n} &= \mathbf{o}, \\ \mathbf{n}\times\mathbf{o} &= \mathbf{a}. \end{aligned} \tag{48}$$

By virtue of its structure, the general $\mathbf{T}$ matrix may be equated to the concatenation of the vector $\mathbf{p}$ and the three orthonormal vectors $\mathbf{n}$, $\mathbf{o}$ and $\mathbf{a}$, thus:

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \mathbf{p} & \mathbf{n} & \mathbf{o} & \mathbf{a} \end{bmatrix}. \tag{49}$$

Expanding Equation (44) employing Equations (45), (47) and (48) yields the following closed form, explicit relations for the pose error:

$$
\begin{aligned}
d_{x_n} &= \sum_{i=1}^{n} \left[\mathbf{n}^u_{i+1} \cdot \mathbf{d}_i + (\mathbf{p}^u_{i+1} \times \mathbf{n}^u_{i+1}) \cdot \boldsymbol{\delta}_i\right], \\
d_{y_n} &= \sum_{i=1}^{n} \left[\mathbf{o}^u_{i+1} \cdot \mathbf{d}_i + (\mathbf{p}^u_{i+1} \times \mathbf{o}^u_{i+1}) \cdot \boldsymbol{\delta}_i\right], \\
d_{z_n} &= \sum_{i=1}^{n} \left[\mathbf{a}^u_{i+1} \cdot \mathbf{d}_i + (\mathbf{p}^u_{i+1} \times \mathbf{a}^u_{i+1}) \cdot \boldsymbol{\delta}_i\right], \\
\delta_{x_n} &= \sum_{i=1}^{n} \left(\mathbf{n}^u_{i+1} \cdot \boldsymbol{\delta}_i\right), \\
\delta_{y_n} &= \sum_{i=1}^{n} \left(\mathbf{o}^u_{i+1} \cdot \boldsymbol{\delta}_i\right), \\
\delta_{z_n} &= \sum_{i=1}^{n} \left(\mathbf{a}^u_{i+1} \cdot \boldsymbol{\delta}_i\right),
\end{aligned}
\tag{50}
$$

where the vectors $\mathbf{p}^u_{i+1}$, $\mathbf{n}^u_{i+1}$, $\mathbf{o}^u_{i+1}$ and $\mathbf{a}^u_{i+1}$ comprise the $\mathbf{U}_{i+1}$ matrix:

$$
\mathbf{U}_{i+1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \mathbf{p}^u_{i+1} & \mathbf{n}^u_{i+1} & \mathbf{o}^u_{i+1} & \mathbf{a}^u_{i+1} \end{bmatrix}, \tag{51}
$$

with the superscript u indicating the vector is a column of $\mathbf{U}$. Additionally, the pose errors are

$$
\delta\, {}^{i-1}\mathbf{T}_i \;\Rightarrow\; \Delta\mathbf{x}_i = \begin{bmatrix} \mathbf{d}_i \\ \boldsymbol{\delta}_i \end{bmatrix}. \tag{52}
$$

Clearly, the $\mathbf{U}_{i+1}$ depend on the nominal DH parameters since they are products of the DH transformations. Whereas, $\mathbf{d}_i$ and $\boldsymbol{\delta}_i$ are functions of the DH error parameters $\Delta\varphi_i$, $\Delta\alpha_i$, Δa_i and Δd_i and by virtue of Equations (33) and (34). Substituting the relations from these two equations into Equations (50) gives the elements of the pose error in terms we can compute:

$$
\begin{aligned}
d_{x_n} &= \sum_{i=1}^{n} \left[(\mathbf{n}_{i+1}^u \cdot \mathbf{k}_i^1) + (\mathbf{p}_{i+1}^u \times \mathbf{n}_{i+1}^u) \cdot \mathbf{k}_i^3\right] \Delta\varphi_i + \left[(\mathbf{p}_{i+1}^u \times \mathbf{n}_{i+1}^u) \cdot \mathbf{k}_i^2\right] \Delta\alpha_i + \\
&\qquad (\mathbf{n}_{i+1}^u \cdot \mathbf{k}_i^2)\Delta a_i + (\mathbf{n}_{i+1}^u \cdot \mathbf{k}_i^3)\Delta d_i, \\
d_{y_n} &= \sum_{i=1}^{n} \left[(\mathbf{o}_{i+1}^u \cdot \mathbf{k}_i^1) + (\mathbf{p}_{i+1}^u \times \mathbf{o}_{i+1}^u) \cdot \mathbf{k}_i^3\right] \Delta\varphi_i + \left[(\mathbf{p}_{i+1}^u \times \mathbf{o}_{i+1}^u) \cdot \mathbf{k}_i^2\right] \Delta\alpha_i + \\
&\qquad (\mathbf{o}_{i+1}^u \cdot \mathbf{k}_i^2)\Delta a_i + (\mathbf{o}_{i+1}^u \cdot \mathbf{k}_i^3)\Delta d_i, \\
d_{z_n} &= \sum_{i=1}^{n} \left[(\mathbf{a}_{i+1}^u \cdot \mathbf{k}_i^1) + (\mathbf{p}_{i+1}^u \times \mathbf{a}_{i+1}^u) \cdot \mathbf{k}_i^3\right] \Delta\varphi_i + \left[(\mathbf{p}_{i+1}^u \times \mathbf{a}_{i+1}^u) \cdot \mathbf{k}_i^2\right] \Delta\alpha_i + \\
&\qquad (\mathbf{a}_{i+1}^u \cdot \mathbf{k}_i^2)\Delta a_i + (\mathbf{a}_{i+1}^u \cdot \mathbf{k}_i^3)\Delta d_i, \qquad (53) \\
\delta_{x_n} &= \sum_{i=1}^{n} (\mathbf{n}_{i+1}^u \cdot \mathbf{k}_i^3)\Delta\varphi_i + (\mathbf{n}_{i+1}^u \cdot \mathbf{k}_i^2)\Delta\alpha_i, \\
\delta_{y_n} &= \sum_{i=1}^{n} (\mathbf{o}_{i+1}^u \cdot \mathbf{k}_i^3)\Delta\varphi_i + (\mathbf{o}_{i+1}^u \cdot \mathbf{k}_i^2)\Delta\alpha_i, \\
\delta_{z_n} &= \sum_{i=1}^{n} (\mathbf{a}_{i+1}^u \cdot \mathbf{k}_i^3)\Delta\varphi_i + (\mathbf{a}_{i+1}^u \cdot \mathbf{k}_i^2)\Delta\alpha_i.
\end{aligned}
$$

Equations (53) can be formulated as a single linear equation

$$
\begin{bmatrix} \mathbf{d}_n \\ \boldsymbol{\delta}_n \end{bmatrix} = \begin{bmatrix} \mathbf{J}_1 & \mathbf{J}_2 & \mathbf{J}_3 & \mathbf{J}_4 \\ \mathbf{J}_4 & \mathbf{J}_3 & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{\varphi} \\ \Delta\boldsymbol{\alpha} \\ \Delta\mathbf{a} \\ \Delta\mathbf{d} \end{bmatrix}. \qquad (54)
$$

For a robot with six revolute axes, $n = 6$. The vector of DH parameter errors has dimension 24×1 and is comprised of the four concatenated 6×1 individual DH parameter error vectors, $\Delta\boldsymbol{\vartheta}$, $\Delta\boldsymbol{\alpha}$, $\Delta\mathbf{a}$ and $\Delta\mathbf{d}$. The four 3×6 $\mathbf{J}_i$ sub-matrices are the coefficients of the DH parameter errors. They are built-up as

$$
\mathbf{J}_1 = \begin{bmatrix} [\mathbf{n}_2^u \cdot \mathbf{k}_1^1 + (\mathbf{p}_2^u \times \mathbf{n}_2^1) \cdot \mathbf{k}_1^3] & \cdots & [\mathbf{n}_7^u \cdot \mathbf{k}_6^1 + (\mathbf{p}_7^u \times \mathbf{n}_7^u) \cdot \mathbf{k}_6^3] \\ [\mathbf{o}_2^u \cdot \mathbf{k}_1^1 + (\mathbf{p}_2^u \times \mathbf{o}_2^1) \cdot \mathbf{k}_1^3] & \cdots & [\mathbf{o}_7^u \cdot \mathbf{k}_6^1 + (\mathbf{p}_7^u \times \mathbf{o}_7^u) \cdot \mathbf{k}_6^3] \\ [\mathbf{a}_2^u \cdot \mathbf{k}_1^1 + (\mathbf{p}_2^u \times \mathbf{a}_2^1) \cdot \mathbf{k}_1^3] & \cdots & [\mathbf{a}_7^u \cdot \mathbf{k}_6^1 + (\mathbf{p}_7^u \times \mathbf{a}_7^u) \cdot \mathbf{k}_6^3] \end{bmatrix}, \qquad (55)
$$

$$
\mathbf{J}_2 = \begin{bmatrix} [(\mathbf{p}_2^u \times \mathbf{n}_2^1) \cdot \mathbf{k}_1^2] & \cdots & [(\mathbf{p}_7^u \times \mathbf{n}_7^u) \cdot \mathbf{k}_6^2] \\ [(\mathbf{p}_2^u \times \mathbf{o}_2^1) \cdot \mathbf{k}_1^2] & \cdots & [(\mathbf{p}_7^u \times \mathbf{o}_7^u) \cdot \mathbf{k}_6^2] \\ [(\mathbf{p}_2^u \times \mathbf{a}_2^1) \cdot \mathbf{k}_1^2] & \cdots & [(\mathbf{p}_7^u \times \mathbf{a}_7^u) \cdot \mathbf{k}_6^2] \end{bmatrix}, \qquad (56)
$$

$$
\mathbf{J}_3 = \begin{bmatrix} (\mathbf{n}_2^u \cdot \mathbf{k}_1^2) & \cdots & (\mathbf{n}_7^u \cdot \mathbf{k}_6^2) \\ (\mathbf{o}_2^u \cdot \mathbf{k}_1^2) & \cdots & (\mathbf{o}_7^u \cdot \mathbf{k}_6^2) \\ (\mathbf{a}_2^u \cdot \mathbf{k}_1^2) & \cdots & (\mathbf{a}_7^u \cdot \mathbf{k}_6^2) \end{bmatrix}, \qquad (57)
$$

$$\mathbf{J}_4 = \begin{bmatrix} (\mathbf{n}_2^u \cdot \mathbf{k}_1^3) & \cdots & (\mathbf{n}_7^u \cdot \mathbf{k}_6^3) \\ (\mathbf{o}_2^u \cdot \mathbf{k}_1^3) & \cdots & (\mathbf{o}_7^u \cdot \mathbf{k}_6^3) \\ (\mathbf{a}_2^u \cdot \mathbf{k}_1^3) & \cdots & (\mathbf{a}_7^u \cdot \mathbf{k}_6^3) \end{bmatrix}. \tag{58}$$

Equation (54) is nothing more than the Jacobian formulation given by Equation (3), repeated below

$$\Delta\mathbf{x} = \mathbf{J}\Delta\boldsymbol{\rho}, \tag{59}$$

where $\Delta\mathbf{x}$ is the 6×1 pose error vector of measured pose less computed pose, $\mathbf{J}$ is the 6×24 Jacobian and $\Delta\boldsymbol{\rho}$ is the 24×1 DH parameter error vector of actual less nominal DH parameters.

6 Obtaining the Jacobian by Direct Differentiation

The Jacobian is a mapping between the Cartesian linear and angular velocity of the EE reference frame and the joint rates. If the DH parameters are considered to be constant the Jacobian is a function of only the joint angles. The relationship is represented by:

$$\dot{\mathbf{x}} = \mathbf{J}\dot{\boldsymbol{\theta}}, \tag{60}$$

where $\dot{\mathbf{x}}$ is the 6×1 vector of Cartesian linear and angular velocities, $\mathbf{J}$ is the 6×6 Jacobian and $\dot{\boldsymbol{\theta}}$ is the 6×1 vector of joint rates. The Jacobian can be obtained from the Plücker line coordinates of the six joint axes with respect to the base frame origin. It is well known that the Jacobian can be partitioned into linear and angular components. If only the linear components are considered, as is the case when only linear measurements of the EE reference point are made, the sub-Jacobian relating linear EE reference point velocities to the six joint rates is easily obtained by taking the partial derivatives of the transformation equation for the forward kinematics. However, we are limited to three measurements with this method as the forward kinematics is a point transformation, limiting the dimension to three.

If the DH kinematic model is used then in addition to the partial derivatives with respect to the joint angles, the partial derivatives with respect to the DH parameters are also required. First, we need a functional relationship that we can differentiate. We take:

$$\mathbf{x} = \mathbf{f}(\boldsymbol{\rho}), \tag{61}$$

where $\mathbf{x}$ is the homogeneous vector of Cartesian coordinates of the EE reference point expressed in the base frame of the robot and $\mathbf{f}(\boldsymbol{\rho})$ is the product of the homogeneous coordinate transformation from the EE frame to the base frame and the homogeneous vector of Cartesian coordinates of the EE reference point expressed in the EE reference frame. The homogeneous coordinate transformation is a function of (see Section 3): the joint angles, ϕ_i; the joint twist angles, α_i; the link lengths, a_i; and the link offsets, d_i. For the simplest model of a 6R robot, there are a minimum of 24 DH parameters (i.e., 4×6). This simple model does not calibrate the robot base frame relative to the measurement frame, nor the tool frame (i.e., the camera attached to the robot tool, flange) relative to the EE frame. But, for our measurement concept, which provides distance differences, we shall see that these additional frames need not be accounted for.

Taking the partial derivatives of $\mathbf{f}(\boldsymbol{\rho})$ with respect to all 24 parameters we obtain after eliminating the homogeneous coordinate (which vanishes upon differentiation):

$$\begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \frac{\boldsymbol{\partial f}}{\boldsymbol{\partial \rho}} \begin{bmatrix} \dot{\boldsymbol{\phi}} \\ \dot{\boldsymbol{\alpha}} \\ \dot{\mathbf{a}} \\ \dot{\mathbf{d}} \end{bmatrix}, \tag{62}$$

where,

$$\frac{\boldsymbol{\partial f}}{\boldsymbol{\partial \rho}} = \begin{bmatrix} \frac{\partial f_1}{\partial \phi_1} & \cdots & \frac{\partial f_1}{\partial \phi_6} & \frac{\partial f_1}{\partial \alpha_1} & \cdots & \frac{\partial f_1}{\partial \alpha_6} & \frac{\partial f_1}{\partial a_1} & \cdots & \frac{\partial f_1}{\partial a_6} & \frac{\partial f_1}{\partial d_1} & \cdots & \frac{\partial f_1}{\partial d_6} \\ \frac{\partial f_2}{\partial \phi_1} & \cdots & \frac{\partial f_2}{\partial \phi_6} & \frac{\partial f_2}{\partial \alpha_1} & \cdots & \frac{\partial f_2}{\partial \alpha_6} & \frac{\partial f_2}{\partial a_1} & \cdots & \frac{\partial f_2}{\partial a_6} & \frac{\partial f_2}{\partial d_1} & \cdots & \frac{\partial f_2}{\partial d_6} \\ \frac{\partial f_3}{\partial \phi_1} & \cdots & \frac{\partial f_3}{\partial \phi_6} & \frac{\partial f_3}{\partial \alpha_1} & \cdots & \frac{\partial f_3}{\partial \alpha_6} & \frac{\partial f_3}{\partial a_1} & \cdots & \frac{\partial f_3}{\partial a_6} & \frac{\partial f_3}{\partial d_1} & \cdots & \frac{\partial f_3}{\partial d_6} \end{bmatrix}. \tag{63}$$

This 3×24 Jacobian relating the linear velocities to changes in the DH parameters is completely general and can be applied to any 6R wrist-partitioned robot architecture, but its terms are quite complicated.

If the displacement errors due to the difference between where the robot *thinks* it is and where it *actually* is are small relative to the link lengths then Equation (62) can be used to represent this difference. That is, the difference equations are approximated by Equation (62), and we can write

$$\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \frac{\boldsymbol{\partial f}}{\boldsymbol{\partial \rho}} \begin{bmatrix} \Delta \boldsymbol{\phi} \\ \Delta \boldsymbol{\alpha} \\ \Delta \mathbf{a} \\ \Delta \mathbf{d} \end{bmatrix}, \tag{64}$$

where the Δ indicates the difference between measured and computed locations for the Cartesian EE coordinates, while for the joint angles and DH parameters it is the difference between actual and nominal parameter values.

For a simulated calibration procedure implemented in MATLAB the symbolic expressions for each element in the Jacobian were evaluated using Maple V and converted to MATLAB pseudo code, with some minor modifications. The Maple V command string `C(J[1,1],optimized)` produces the MATLAB pseudo code for the first row, first column Jacobian element. The additional arguement *optimized* causes common sub-expression optimisation to be performed. The Maple V output is then *cut-and-pasted* into a MATLAB sub-routine used to evaluate the Jacobian.

7 DH Parameter Identification

We are interested in solving either Equation (59), or Equation (64) for $\Delta\boldsymbol{\rho}$. Since there are 24 unknowns and only 6, or 3 equations, either we need to formulate more equations, or measure more than one pose and approximate the solution in a least squares sense. We'll go with measuring more than one pose.

For purposes of discussing the parameter identification let us assume that we have only EE reference point coordinates for measurements. If we measure m robot poses and assume the DH parameter errors to be constant then $\Delta\mathbf{x}$ will have dimensions $3m \times 1$ and $\mathbf{J}$ will have $3m \times 24$.

If the world were not cruel and the Jacobian had full rank (for a suitably large over determined matrix full rank means that all there is no dependence between columns, hence in our case full rank means 24) then we could simply employ the Householder reflection algorithm embedded in MATLAB's *matrix division* algorithm. This results in the least squares approximation to the overconstrained system of equations. The MATLAB syntax is simply

$$\Delta\boldsymbol{\rho} = \mathbf{J} \setminus \Delta\mathbf{x}.$$

However, the world is cruel. Because of the DH formulation, there is no guarantee that the Jacobian will have full rank, and hence Householder reflections cannot, in general, be used to solve the system in a least squares sense. An alternate approach is required.

7.1 Singular Value Decomposition

Singular value decomposition (SVD) is a very powerful method that can be used to solve sets of equations that are either singular, or numerically very close to singular [11]. SVD methods are based on a theorem which states that any $m \times n$ matrix $\mathbf{A}$, whose number of rows is greater than, or equal to its number of columns, $m \geq n$, can be written as the product of an $m \times n$ column-orthogonal matrix $\mathbf{U}$ (i.e., the dot product of each column with itself is unity), an $n \times n$ diagonal matrix $\mathbf{S}$ (i.e., the off diagonal elements are all zero) whose elements are greater than, or equal to zero (the *singular values* of matrix $\mathbf{A}$), and the transpose of an $n \times n$ orthormal matrix $\mathbf{V}$ (i.e., $\mathbf{V}\mathbf{V}^T = \mathbf{I}$):

$$\mathbf{A}_{m\times n} = \mathbf{U}_{m\times n}\mathbf{S}_{n\times n}\mathbf{V}^T_{n\times n}. \tag{65}$$

When we wish to determine the least-squares solution to an overconstrained set of linear equations (i.e., more equations than unknowns) $\mathbf{A}\mathbf{x} = \mathbf{b}$ whose coefficient matrix, $\mathbf{A}$, has some column dependencies we can use SVD. The least-squares solution vector $\mathbf{x}$ is given by

$$\mathbf{x} = \{\mathbf{V}[\mathbf{S}^{-1}(\mathbf{U}^T\mathbf{b})]\}. \tag{66}$$

Note, the computation must be carried out from *right to left*, as indicated by the brackets.

Let us examine the inverse of the singular value matrix, $\mathbf{S}^{-1}$, a bit more carefully. Because $\mathbf{S}$ is a square diagonal matrix, its inverse is given by

$$\mathbf{S}^{-1} = \left[\text{diag}\left(\frac{1}{s_j}\right)\right]. \tag{67}$$

If there are column degeneracies in $\mathbf{A}$ then some of the singular values will be zero, or close to zero, depending on the level of degeneracy. For no column degeneracy, the rank of matrix **A** will be equal to the number of its columns. If it is rank deficient by one, then two columns are linearly dependent and one of the singular values will vanish. In general, the rank of the matrix corresponds to the number of non-zero singular values. It may also be that **A** has full rank, but two, or more, columns may be *nearly* linearly dependent. This, in turn, means that some of the singular values will be very small compared to the others. Since Equation (66) depends on $\mathbf{S}^{-1}$ the solution will be numerically unstable.

The remedy to this problem is to simply set the troublesome $1/s_j$ equal to zero! This may seem a bad decision because it makes our rank deficient system even more rank deficient. But, we actually improve the fit of the quantities that we can solve for. By eliminating the small $1/s_j$ we are throwing away linear combinations of equations that are so corrupted by round-off error as to be, at best, useless. That is, the columns in **V** corresponding to the zeroed $1/s_j$ are linear combinations of solutions for **x** which are insensitive to the measured data in **b**.

8 Simulated MATLAB Calibration Experiment

Using the Jacobian formulated as in Section 6, a MATLAB simulation of a calibration experiment was performed. A KUKA KR-15/2 robot was modelled using DH parameters (see Table 1). A home position was selected and arbitrary, but constant increments were sequentially added to the home joint angles to change the robot configuration, these are all listed in Table 2. The increments were selected so the final position would be in the workspace after 100 increment steps, and so that all the joints would move in different ways. The number of measured positions was set to be 100. The tolerance on the smallness of singular values was set to be 10^{-6}, and the convergence criterion was set to be

$$\| \mathbf{J}\boldsymbol{\Delta\rho} - \boldsymbol{\Delta x} \| \leq 10^{-8}.$$

Computed poses were determined with the nominal DH parameters and measured poses were determined with some assigned parameter errors. The calibration procedure was then run. Table 3 lists the results together with the % difference relative to assigned parameter error values (note that degrees are given in *rads*, while lengths are in *m*).

For the simulated calibration experiment data the procedure converged after two iterations. Three of the singular values generated by the SVD algorithm were set to zero. The em effective rank of the 300×24 identification Jacobian was, therefore, reduced to 21. Hence, only 21 of the parameters are observable.

8.1 Which Parameters Were Identified?

The calibration procedure identified 17 of the 24 parameter errors exactly. Three additional parameters, $\Delta\alpha_5$, Δd_2 and Δd_3 were identified with less than 25% difference from the assigned error value, moreover the signs of these three errors agree with the assigned ones. The identified error $\Delta\varphi_5$ was 87.65% greater than assigned, but the sign was correctly determined. The remaining three identified parameter errors were completely wrong. However, the SVD algorithm had functionally reduced the rank of the identification Jacobian by three, which justifies the above results.

Initial position (deg.)		Increments (deg.)	
φ_1	0.0	$\Delta\varphi_1$	-3.0
φ_2	-90.0	$\Delta\varphi_2$	3.0
φ_3	0.0	$\Delta\varphi_3$	-2.0
φ_4	0.0	$\Delta\varphi_4$	-3.5
φ_5	0.0	$\Delta\varphi_5$	3.2
φ_6	0.0	$\Delta\varphi_6$	-2.5

Table 2: Initial joint angles; constant increments.

The question remains, how do we know which of the parameters have been accurately identified in the absence of apriori knowledge?

8.2 On the Completeness of the Kinematic Model

The DH parameters represent the geometric components of the robot's kinematic geometry. There are additionally non-geometric components, due primarily to the characteristics of the joints, which make the encoder outputs different from the true joint angles. This can be modelled by [12]

$$\varphi_i = \mathrm{k}_i\psi_i + \lambda_i, \tag{68}$$

where for the ith joint ψ_i is the joint encoder output, k_i is the gain of joint angle sensor and λ_i is the joint angle offset. However, for our first attempt, we assumed the gains to be all unity and the joint angle offsets to be perfect.

9 Pose Measurement

The most difficult aspect of kinematic calibration is obtaining accurate measurements of the EE pose. If, for instance, the aim is to calibrate the robot to have $\pm 100\mu$ positioning accuracy then position measurements with $\pm 10\mu$ are required. These measurements are costly in both time and money.

The measurement system consists A CCD camera, two MEL displacement sensors and an LED array all mounted to the tool flange of the robot, see Figure 7. The measurement equipment data are listed in table 4.

Parameter error	Assigned	Identified	% Difference
$\Delta\varphi_1$	0.000870	0.00087	0
$\Delta\varphi_2$	0.000940	0.00094	0
$\Delta\varphi_3$	-0.001000	-0.00100	0
$\Delta\varphi_4$	0.000620	0.00062	0
$\Delta\varphi_5$	-0.000810	-0.00010	87.65
$\Delta\varphi_6$	0.000260	-5×10^{-9}	-100.00
$\Delta\alpha_1$	0.000157	0.000157	0
$\Delta\alpha_2$	0.000130	0.000130	0
$\Delta\alpha_3$	-0.000160	-0.000160	0
$\Delta\alpha_4$	-0.000253	-0.000253	0
$\Delta\alpha_5$	0.000462	0.000524	13.52
$\Delta\alpha_6$	-0.000320	-0.000320	0
Δa_1	0.000031	0.000031	0
Δa_2	0.000051	0.000051	0
Δa_3	0.000012	0.000012	0
Δa_4	-0.000045	-0.000045	0
Δa_5	0.000064	-0.000176	-375.00
Δa_6	0.000058	0.000058	0
Δd_1	-0.000075	-0.000075	0
Δd_2	0.000031	0.000026	-16.13
Δd_3	0.000022	0.000027	22.73
Δd_4	0.000048	0.000048	0
Δd_5	-0.000020	0.000001	105.00
Δd_6	0.000078	0.000078	0

Table 3: Assigned, identified parameter errors, % difference for simulated calibration experiment.

CCD-Camera	Pulnix TM-6CN	Resolution: 752(H)x582(V) Cell size: 8.6(H)x8.3(V) μm
Electronic shutter		1/10000 s
Camera lens	Rodenstock	macro 2x CCD Lens
Displacement sensor 1	MEL M5L/10 Laser	1V/0.2mm
Displacement sensor 2	MEL M52L/2 Stereo Laser	1V/0.5mm
Multimeter	Hewlett Packard	34401A multimeter
Framegrabber	National Instruments	PCI-1408 monochrome
GPIB card	National Instruments	AT-GPIB card
Flat standard	PZA 1000x50x10mm	DIN 874/0 $\pm 0.8 \mu m$
Length standard	PZA 1000x20x20mm, ruled	DIN 865 $\pm 10 \mu m$
Lighting		Red LED array

Table 4: Measurement equipment.

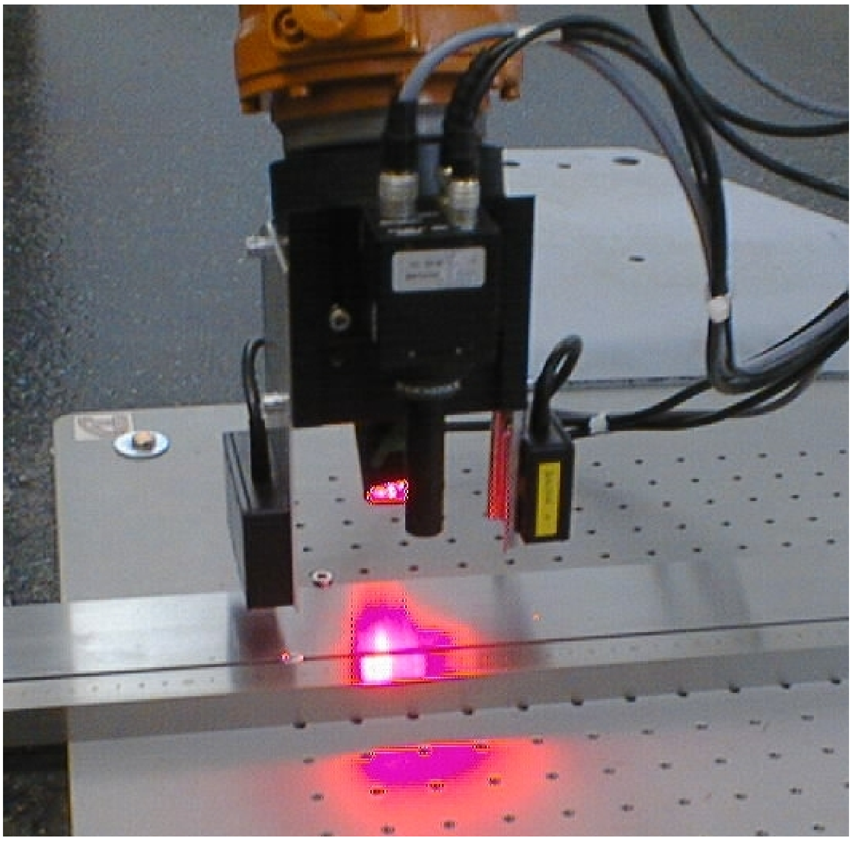

Figure 7: The measurement head.

10 Measurement Concept

Because the six error equations are independent, we should be able to use any subset to identify the parameters. We have opted to try using only position measurements. As long as all six robot axes are used, we should have enough to identify all parameters.

We determined the x and y robot base frame axes directions by having the robot draw lines on the floor in these two directions. A plate with a 21×21 grid of holes precisely drilled with 2.5cm between centres was rigidly mounted to the floor so the grid was aligned with the axis directions of the base frame. Then two straight edges, one flat and one ruled, were set on the grid so they were parallel and their length were in either the x or y directions. The robot was then moved to a *zero* position over so that the plane of the CCD camera chip was parallel to the xy plane, and so that the ruled markings of the ruled straight edge were in its field of view and so the MEL distance sensors were positioned above the flat straight edge.

We next measured linearly over 80cm in 1cm increments along two lines in the y robot base frame basis directions and over 50cm in 1cm increments along three in the x. The different distances was because of an oversight in positioning the plate: portions of the metre long straight edges were always out of the work envelope of the robot.

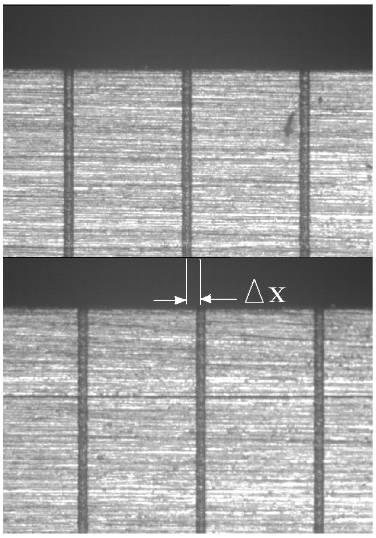

Figure 8: The x positioning error between two images.

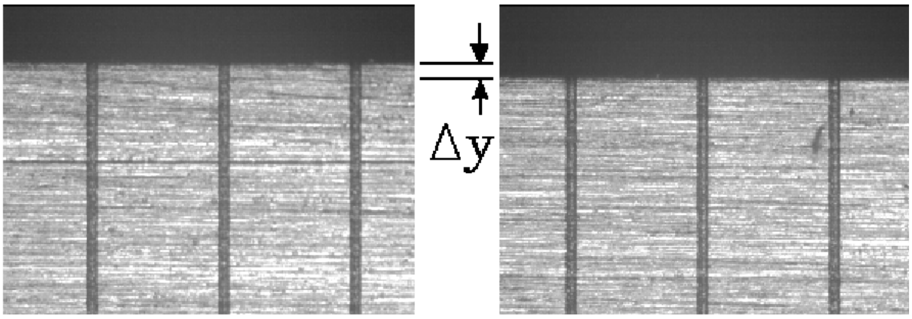

Figure 9: The y positioning error between two images.

11 Discussion

Since we are only concerned in differences in position we only need relative position measurements. That is, by using the *LIN_REL* KRC command and reading the computed position data from the robot controller we could compute the difference between computed and measured position in the direction along the ruler. For a measurement in the x direction, Figures 8, 9, and 10 show how measured differences were calculated. Image processing gave us the linear regression in the centre ruler marking in each image. The edge of the ruler was similarly computed and the y coordinate was the intersection of the two lines. The z coordinate was given by the MEL distance sensors. Additional information on the orientation changes about the axis perpendicular to the ruler length and z directions.

However, at this point, only simulated measurement data has been used in the first parameter identification attempts. So far we have not met with success. We always encounter rank deficiency of $\mathbf{J}$.

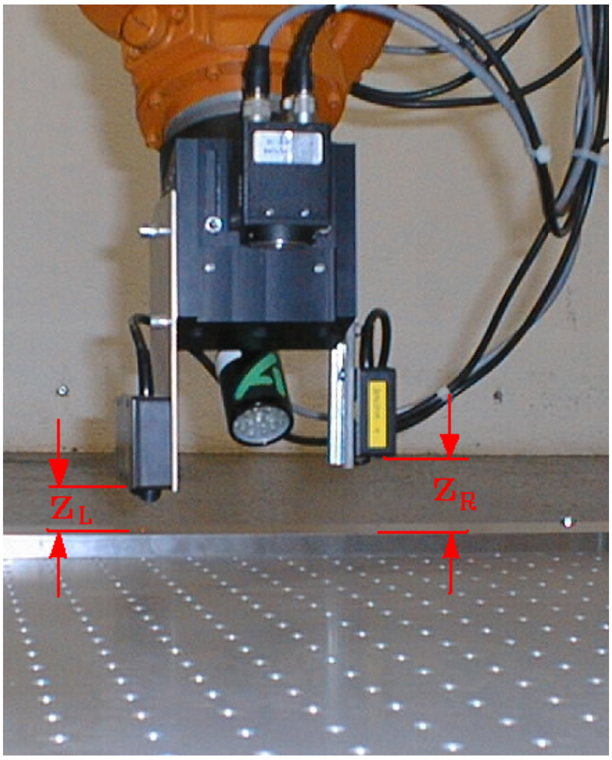

Figure 10: The z position measurement.

References

[1] M.J.D. Hayes, M. Leitner, P. O'Leary, R. Ofner, and C. Sallinger. Cable Connector Profile Measurement Using Laser Light Plane Sectioning. Technical report, Institute for Automation, University of Leoben, July 21 2000.

[2] C.H. An, C.G. Atkeson, and J.M. Hollerbach. *Model-Based Control of a Robot Manipulator.* The MIT Press, Cambridge, Massachusetts, U.S.A., 1988.

[3] J.J. Craig. *Introduction to Robotics, Mechanics and Control,* second edition. Addison-Wesley Publishing Co., Reading, Mass., U.S.A., 1989.

[4] B.W. Mooring, Z.S. Roth, and M.R. Driels. *Fundamentals of Manipulator Calibration.* John Wiley and Sons, Inc., New York, N.Y., U.S.A., 1991.

[5] J. Denavit and R.S. Hartenberg. "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices". *J. of Applied Mechanics*, pages 215–221, 1955.

[6] P.I. Corke. "A Robotics Toolbox for MATLAB". *IEEE Robotics and Automation Magazine*, vol. 3, no.1: pages 24–32, March 1996.

[7] J. Angeles. *Fundamentals of Robotic Mechanical Systems: Theory, Methods, and Algorithms.* Springer-Verlag, New York, N.Y., U.S.A., 1997.

[8] S.A. Hayati. “Robot Arm Geometric Link Parameter Estimation ”. In *Proceedings of the 22nd IEEE Conference on Decision and Control*, pages 1477–1483, December 1983.

[9] Chunche Gong, Jingxia Yuan, and Jun Ni. “A Self-Calibration Method for Robotic Measurement System”. *ASME Manufacturing Science and Engineering*, MED-vol. 10: pages 879–888, 1999.

[10] H. Zhuang and Z.S. Roth. *Camera-Aided Robot Calibration.* CRC Press, Boca Raton, Fla., U.S.A., 1996.

[11] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C, 2nd Edition.* Cambridge University Press, Cambridge, England, 1992.

[12] J.M. Hollerbach and C.W. Wampler. “The Calibration Index and Taxonomy for Robot Kinematic Calibration Methods”. *International Journal of Robotics Research*, vol. 15: pages 705–712, 1996.