GECARO: A system for the GEometric CAlibration of RObots

Wisama Khalil, Philippe Lemoine

GECARO: A System for the GEometric CAlibration of RObots

Wisama Khalil – Philippe Lemoine

Institut de Recherche en Cybernétique de Nantes, UMR 6597
Ecole Centrale de Nantes
1 rue de la Noë, B.P. 92 101, F-44321 Nantes cedex
Wisama.Khalil@ircyn.ec-nantes.fr

ABSTRACT. This paper presents a software package for the simulation and the practical calibration of the geometric parameters of robots. This system which is called GECARO, GEometric CAlibration of RObots, contains a large variety of methods to identify the geometric parameters of robots. GECARO is running on PC computers and developed using MATLAB; any general serial robot can be treated directly. The identifiable parameters are determined using a numerical method based on the QR decomposition, while the identification is carried out using linearized model which is solved iteratively using least squares criterion and by updating the observation matrix after each iteration.

RÉSUMÉ. Cet article présente un logiciel pour la simulation et la réalisation de l'étalonnage de paramètres géométriques des robots. Ce système qui est baptisé GECARO, GEometric CAlibration of RObots, est composé d'un ensemble de méthodes pour l'identification de paramètres géométriques. GECARO est développé sous MATLAB et s'exécute sur PC ; il peut traiter directement n'importe quel robot à chaîne ouverte simple. Les paramètres identifiables pour chaque méthode sont déterminés par une méthode numérique fondée sur la décomposition QR de la matrice d'observation. L'algorithme d'identification utilise une méthode itérative du type moindres carrés linéaire qui met à jours les paramètres géométriques après chaque itération.

KEY WORDS: calibration, geometric parameters, static accuracy, identification, robots.
MOTS-CLÉS : étalonnage, paramètres géométriques, précision statique, identification, robots.

1. Introduction

The absolute accuracy of a robot depends to a large extent on the accuracy of the values of the geometric parameters used in the direct and inverse geometric models. Geometric calibration of robots is the process by which the parameters defining the base frame parameters, link parameters and end-effector parameters are precisely identified.

Classically, the geometric calibration is carried out by solving a system of linear or non-linear equations which is a function of the joint positions and the location (position and orientation) of the end-effector frame [ROT 87] [WHI 86] [CAE 93] [GUY 95] [DAM 96] [HOL 89]. All the robots are provided with joint position sensors, thus the problem of using the classical methods is to find the appropriate external sensor which can measure with sufficient accuracy the position or location of the terminal frame in the world fixed frame. To overcome this problem, autonomous calibration methods which do not need an external sensor have been proposed [BEN 91] [EDW 94] [ZHO 95] [TAN 94] [KHA 95] [NAK 95], one can find also methods using simple sensors giving the distance between the terminal point of two configurations.

The software package GECARO contains a complete set of these methods, such that we can select the appropriate method as function of the robot and the available sensors.

The paper is organized as follows: the parameters defining the robot will be presented in section 2. In section 3 we describe the calibration methods developed in GECARO. Section 4 presents the software package GECARO. Section 5 presents an example. Section 6 is the conclusion.

2. Description of the robot

Two types of parameters are needed:

– The geometric parameters defining the location of the end-effector of the robot with respect to the fixed world frame.

– The joint parameters defining the joint limits, the joint offsets, the joint transmission gains, and the coupling matrices between the motor and joint variables.

2.1. *Definition of the geometric parameters*

We consider serial robots consisting of n joints and n+1 links. Link 0 is the base and link n is the terminal link, frame j is defined fixed on link j. We denote:

frame -1: the fixed reference frame,

frame n+1: the end-effector frame.

The end-effector location can be calculated with respect to the reference frame by the direct geometric model:

$$ {}^{-1}\mathbf{T}_{n+1} = {}^{-1}\mathbf{T}_0 \, {}^{0}\mathbf{T}_n(\mathbf{q}) \, {}^{n}\mathbf{T}_{n+1} \qquad [1] $$

where ${}^{i}\mathbf{T}_j$ is the (4x4) transformation matrix defining frame j with respect to frame i, and $\mathbf{q}$ is the (nx1) joint variables vector.

The definition of the link frames will be carried out by Khalil and Kleinfinger notation [KHA 86] [KHA 99]. Frame j is defined such that $\mathbf{z}_j$ is along the axis of joint j, and $\mathbf{x}_j$ is perpendicular to $\mathbf{z}_j$ and $\mathbf{z}_{j+1}$. Frame j is defined with respect to frame j-1 by the matrix ${}^{j-1}\mathbf{T}_j$, function of the four parameters (α_j, d_j, θ_j, r_j), [Figure 1], such that:

$$ {}^{j-1}\mathbf{T}_j = \mathbf{Rot}(\mathbf{x},\alpha_j)\,\mathbf{Trans}(\mathbf{x},d_j)\,\mathbf{Rot}(\mathbf{z},\theta_j)\,\mathbf{Trans}(\mathbf{z},r_j) \qquad [2] $$

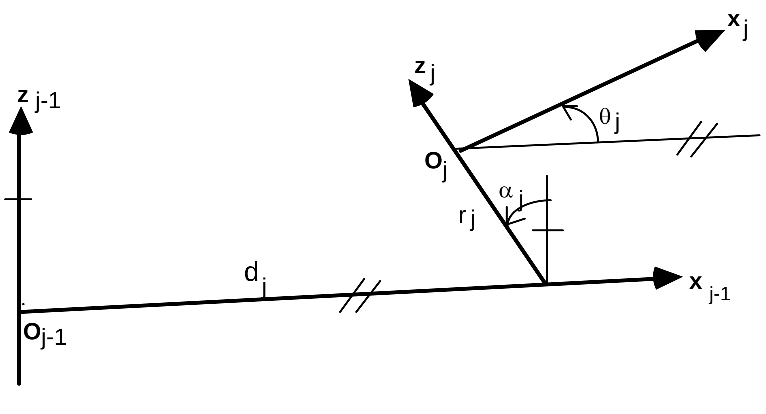

Figure 1. *Definition of frame j with respect to frame j-1*

Although that frames ${}^{-1}\mathbf{T}_0$ and ${}^{n}\mathbf{T}_{n+1}$ can be arbitrarily defined, it have been proved that the calculation of ${}^{-1}\mathbf{T}_{n+1}$ can be obtained as a function of the geometric parameters of n+2 frames represented by four parameters for each frame (α_j, d_j, θ_j, r_j) for j = 0,…,n+1, except for the first frame for which $\alpha_0 = 0$, $d_0 = 0$ [KHA 91a] [KHA 99]. The joint variable q_j is equal to θ_j if joint j is rotational and r_j if joint j is prismatic. We define the type of the joint by the parameter σ_j , with:

$\sigma_j = 0$ for j rotational, $\sigma_j = 1$ for j prismatic, and $\sigma_j = 2$ if frame j contains only constant parameters as in the case of frame 0 and frame n+1.

If the axis of joint j is parallel to the axis of joint j-1, an additional parameter $\Delta\beta_j$ must be considered [HAY 88], the nominal value of β_j is equal to zero, such that:

$$ {}^{j-1}\mathbf{T}_j = \mathbf{Rot}(\mathbf{y},\beta_j)\ \mathbf{Rot}(\mathbf{x},\alpha_j)\ \mathbf{Trans}(\mathbf{x},d_j)\ \mathbf{Rot}(\mathbf{z},\theta_j)\ \mathbf{Trans}(\mathbf{z},r_j) \qquad [3] $$

The calibration object is to identify the deviation of the real parameters from the nominal values, thus to identify: $\Delta\theta_0$, Δr_0, and $\Delta\alpha_j$, Δd_j, $\Delta\theta_j$, Δr_j, $\Delta\beta_j$ (for j = 1, n+1).

2.2. *Definition of joint parameters*

We mean by joint parameters the following:

– **Qmax** and **Qmin**: the vectors of joint limits.

– **Offset**: a vector, containing the nominal values of the offset of joint positions.

– **K**: the vector of transmission gains [HOL 89]. This parameter can be calibrated in the same manner as the geometric parameters. Thus we can identify also the errors in the transmission gains of the joints ΔK_j (for j = 1, n).

– **Cm**: the (nxn) coupling matrix between the motor variables.

– **Cj**: the (nxn) coupling matrix between the joint variables.

The joint variables are calculated as a function of motor variables using the following relation:

$$ \mathbf{q = Cj * diag(K) * Cm * qm + Offset} \qquad [4] $$

where:
q is the (nx1) joint variables vector,
qm is the (nx1) vector of motor variables,
diag(**K**) is a (nxn) matrix, whose (j,j) element is equal to K_j.

– In addition to the geometric and joint parameters, we need in the calibration process to define a (n+2)x6 matrix which is called the priority matrix. This matrix defines the order of priority of the geometric parameters. Columns 1,...,6 correspond to the parameters α, d, θ, r, β, K respectively. The elements of row j give the priority of the parameters of frame j. When calculating the identifiable parameters, the parameters will be ordered as function of the corresponding priority number, the first parameters will be those having the highest priority number. The importance of this matrix comes from the fact that we construct the base of identifiable parameters as those corresponding to the first independent columns. As a general rule we give a high priority to the parameters which can be updated easily in the control system. If the priority of a parameter is equal to zero, it will not be identified.

3. Calibration methods

The calibration methods of the geometric parameters differ according to the variables used in the calibration model and the measuring devices to carry out them. A unified approach for all of them can be formulated using the framework suggested in [WAM 88] [HOL 96]. According to this formulation the calibration equation can be written in the general form:

$$\mathbf{0} = \mathbf{f}\,(\mathbf{q},\mathbf{X},\eta_r) \qquad [5]$$

where:
X represents the Cartesian variables giving the position and orientation of the terminal frame,
η_r is the vector of the real (unknown) values of the geometric parameters.

The nonlinear calibration equation can be linearized to get the differential equation:

$$\Delta\mathbf{y} = \Psi(\mathbf{q},\mathbf{X},\eta)\,\Delta\eta \qquad [6]$$

where:
$\Delta\eta = \eta_r - \eta$, defines the (Npx1) vector of the errors on the geometric parameters,
η is the vector of the nominal values of the geometric parameters.

To identify η_r the calibration equation will be applied for a sufficient number of configurations and the global system of equations will be solved using non linear techniques for equation [5] or linear iterative techniques for equation [6].

3.1. *Calibration using classical methods*

This method is based on using a set of configurations for which the joint positions and the corresponding Cartesian coordinates of the terminal frame position or location are given. Beside the joint position sensors, we need an external sensor to measure the position or location of the end-effector.

The location of the end-effector with respect to the reference frame can be calculated using the nominal parameters and the joint positions by the use of the direct geometric model as given in equation [1]. The geometric parameters will be calibrated in order to minimize the difference between the measured end-effector location and the calculated location.

Using a first order Taylor development, a linear differential model defining the deviation of the end effector location due to the differential error in the geometric parameters can be obtained as [KHA 91a]:

$$\Delta\mathbf{x} = \mathbf{J}(\mathbf{q},\eta)\ \Delta\eta \qquad [7]$$

where:

$\Delta\mathbf{x}$ represents the (6x1) vector of the position and orientation error (difference between measured and calculated ${}^{-1}\mathbf{T}_{n+1}$), in Appendix I we show how to calculate $\Delta\mathbf{x}$ from the elements of the transformation matrices.

$\mathbf{J}(\mathbf{q},\eta)$ is the (6xNp) Jacobian matrix of frame (n+1) with respect to the geometric parameters, the calculation of its columns are given in Appendix II.

$\Delta\eta$ defines the (Npx1) vector of the errors of the geometric parameters.

Equation [7] gives 6 linear equations in the unknown vector $\Delta\eta$.

To identify $\Delta\eta$, equation [7] will be applied for a sufficient number of configurations $\mathbf{q}^1,\ldots,\mathbf{q}^m$, the corresponding locations will be measured and $\Delta\mathbf{x}^i$ will be calculated. The resulting linear system of equations will be represented by:

$$\Delta\mathbf{X} = \mathbf{W}(\mathbf{Q},\eta)\ \Delta\eta \qquad [8]$$

where:

$$\mathbf{W} = \begin{bmatrix} \mathbf{J}(\mathbf{q}^1) \\ \dots \\ \mathbf{J}(\mathbf{q}^m) \end{bmatrix}, \Delta\mathbf{X} = \begin{bmatrix} \Delta\mathbf{x}^1 \\ \dots \\ \Delta\mathbf{x}^m \end{bmatrix}$$

$$\mathbf{Q} = \mathbf{q}^1,\dots,\mathbf{q}^m$$

$\mathbf{W}$ is the observation matrix of dimension ($6mxN_p$) with $6m >> N_p$.

It can be seen that if some columns of **W** are dependent, and **W** is of rank b, then relation [8] can be reduced to [KHA 91b]:

$$\Delta\mathbf{X} = \mathbf{W_b}(\mathbf{Q},\eta)\, \Delta\eta_\mathbf{b} \qquad [9]$$

where $\mathbf{W_b}(\mathbf{q})$ contains arbitrarily b independent columns of **W**. The corresponding parameters are defined by the vector $\Delta\eta_\mathbf{b}$, they are known as identifiable parameters or base parameters.

Equation [9] will be solved to get the least squares errors solution, given by:

$$\Delta\eta_\mathbf{b} = \mathbf{W}_\mathbf{b}^+(\mathbf{Q},\eta)\, \Delta\mathbf{X} \qquad [10]$$

with $\mathbf{W}_\mathbf{b}^+$ the pseudo inverse of $\mathbf{W_b}$.

The geometric parameters will be updated in $\Delta\mathbf{X}$ and $\mathbf{W_b}$. The pseudo inverse solution [10] will be iterated till convergence.

The determination of the identifiable parameters $\Delta\eta_\mathbf{b}$ must be done before the identification process. They can be obtained numerically using the QR decomposition of a matrix **W** similar to that defined in equation [8] but obtained using random configurations [KHA 91b]. The outline of this method is given in Appendix III.

The condition number of $\mathbf{W_b}$ gives an idea about the excitation of the configurations used in the calibration. To get good results with respect to error modelling and measuring noise, the condition number must be close to one [KHA 91a].

If the external sensor gives only the position coordinates of the origin of frame (n+1), then only the first three equations of relation [7] will be used. In this case we must have $3m >> N_p$. The identifiable parameters must be carried out on the corresponding observation matrix.

3.2. *Frame link and position link calibration methods*

In classical calibration methods, we need an accurate, fast and not expensive external sensor to measure the real end-effector position or location. Many sensors have been proposed in the literature but neither fulfil these three conditions. Frame link and position link methods need only the joint position sensors, therefore they are called autonomous methods.

Position or frame link methods construct the observation matrix using the fact that different configurations of the robot can give the same position or location of the terminal effector [KHA 95]. Thus, if $\mathbf{q}^i$ and $\mathbf{q}^j$ represent two different configurations giving the same location of the end-effector frame then:

$$^{-1}\mathbf{T}_{n+1}(\mathbf{q}^i,\eta_r) = {}^{-1}\mathbf{T}_{n+1}(\mathbf{q}^j,\eta_r) \qquad [11]$$

If the values of the geometric parameters in the model η are different than the real values η_r, then using a first order development for relation [11] we obtain:

$$[\mathbf{J}(\mathbf{q}^j,\eta_r) - \mathbf{J}(\mathbf{q}^i,\eta_r)]\, \Delta\eta = \Delta\mathbf{x}(\mathbf{q}^i,\mathbf{q}^j,\eta) \qquad [12]$$

where:

J is the Jacobian matrix of frame (n+1) with respect to the geometric parameters variations, it is defined in equation [7], the expressions of its columns are given in Appendix II**.**

Δx is the (6x1) position and orientation differential vector between configurations i and j, that is to say between $^{-1}\mathbf{T}_{n+1}(\mathbf{q}^i,\eta)$ and $^{-1}\mathbf{T}_{n+1}(\mathbf{q}^j,\eta)$, the calculation of **Δx** is given in Appendix I. Relation [12] gives 6 equations in the geometric parameters errors.

To estimate the vector $\Delta\eta$ we have to use equation [12] for sufficient number of pairs of configurations to construct a system of equations similar to [8].

If $\mathbf{q}^i$ and $\mathbf{q}^j$ give the same position but with different orientations, the observation matrix will be constructed using the first three equations of [12]. This type of equations will be called position link method.

The calculation of the identifiable parameters can be carried out as given in Appendix III, by studying the QR decomposition of an observation matrix **W** calculated from [12] using sufficient random pairs of configurations $\mathbf{q}^i(k)$ and $\mathbf{q}^j(k)$ which give the same end-effector location (or position in the case of position link). Each pair of configurations are obtained by supposing a random configuration $\mathbf{q}^i(k)$ then calculating the terminal frame location ${}^{-1}\mathbf{T}_{n+1}(\mathbf{q}^i)$ (or position ${}^{-1}\mathbf{P}_{n+1}(\mathbf{q}^i)$ in the case of position link) using the direct geometric model, finally $\mathbf{q}^j(k)$ will be the solution of the inverse geometric model for ${}^{-1}\mathbf{T}_{n+1}(\mathbf{q}^i)$ (or ${}^{-1}\mathbf{P}_{n+1}(\mathbf{q}^i)$ in the case of position link) and such that $\mathbf{q}^j \neq \mathbf{q}^i$. The inverse geometric model is calculated using a general numerical method based on the differential model of the robot.

It is to be noted that the identifiable parameters in these methods are different than those of the classical methods. In fact the following cases can be deduced directly:

– If a column is constant in the Jacobian matrix, then the corresponding parameter will not be identified, because the corresponding column in the observation matrix in this method will be zero. The geometric parameters corresponding to frame 0 and frame 1 verify this case.

– If a column of the Jacobian matrix is equal for all $\mathbf{q}^i$ and $\mathbf{q}^j$, then the corresponding column in the observation matrix will be zero, and the corresponding parameter cannot be identified. For example in the case of frame link method since frame n and frame n+1 are the same for both $\mathbf{q}^i$ and $\mathbf{q}^j$, thus from the expressions of the columns of **J**, we can deduce that the following parameters are not identifiable: Δr_n, $\Delta\theta_n$, $\Delta\beta_n$, Δd_{n+1}, $\Delta\alpha_{n+1}$, $\Delta\theta_{n+1}$, Δr_{n+1}.

3.3. *The planar calibration methods*

In this case the calibration will be carried out using a set of configurations of the robot, whose terminal points are lying in the same plane. This idea has been used by different authors [ZHO 95] [TAN 94] [KHA 96] [MAU 96] [IKI 97].

In GECARO two methods are developed, the first makes use of the plane equation, while the second uses the coordinates of the normal to the plane.

3.3.1. The first method: calibration using the plane equation

The general equation of a plane is supposed as:

$$a\,x + b\,y + c\,z + 1 = 0 \qquad [13]$$

where a, b, c represent the plane coefficients.

As the terminal point (origin of frame n+1) of the robot is in the plane, then:

$$a\ Px(\mathbf{q},\eta_r) + b\ Py(\mathbf{q},\eta_r) + c\ Pz(\mathbf{q},\eta_r) + 1 = 0 \qquad [14]$$

where Px, Py, Pz represent the Cartesian coordinates of the terminal point in the world frame.

In the case of error modelling, the use of a first order development for η and the plane coefficients in equation [14], we obtain:

$$[Px(\mathbf{q},\eta) \quad Py(\mathbf{q},\eta) \quad Pz(\mathbf{q},\eta) \quad a\ \mathbf{J}x(\mathbf{q},\eta) + b\ \mathbf{J}y(\mathbf{q},\eta) + c\ \mathbf{J}z(\mathbf{q},\eta)] \begin{bmatrix} \Delta a \\ \Delta b \\ \Delta c \\ \Delta\eta \end{bmatrix}$$

$$= -\ a\ Px(\mathbf{q},\eta) - b\ Py(\mathbf{q},\eta) - c\ Pz(\mathbf{q},\eta) - 1 \qquad [15]$$

where:
Jx, **J**y, and **J**z are the first three rows of the Jacobian matrix **J** defined in [7].

Equation [15] gives a linear equation in $\Delta\eta$, and Δa, Δb, Δc for each configuration**.**

Sufficient number of configurations must be used to obtain a system of equations similar to [8] which will be solved iteratively to identify $\Delta\eta$, and the plane coefficients.

The coefficients of the plane are initialized by calculating the equation of the nearest plane to the terminal points of the given configurations. The coordinates of these points are calculated using the robot geometric model and the nominal values of the geometric parameters.

The calculation of the identifiable parameters can be carried out numerically, by studying the QR decomposition of a matrix **W** calculated from [15] using sufficient random points in a given plane.

If the plane coefficients a, b, and c are precisely known, equation [15] can be reduced to contain only $\Delta\eta$ as unknown.

3.3.2. Second method: calibration using normal coordinates

In this method we make use of the fact that the scalar product of the coordinates of the vector normal to the plane and of any vector between two points in the plane is equal to zero. The main advantage of this procedure is that the coordinates of the normal can be obtained easily using inclinometer [ZHO 95] [MAU 96].

The system general equation is thus:

$$a\,[Px(\mathbf{q}^i,\eta_r)-Px(\mathbf{q}^j,\eta_r)] + b\,[Py(\mathbf{q}^i,\eta_r)-Py(\mathbf{q}^j,\eta_r)] + c\,[Pz(\mathbf{q}^i,\eta_r)-Pz(\mathbf{q}^j,\eta_r)] = 0 \qquad [16]$$

where $Pu(\mathbf{q}^i,\eta_r)$, for u = x, y, z represent the coordinates of the terminal point ${}^{-1}\mathbf{P}_{n+1}(\mathbf{q}^i)$ at configuration $\mathbf{q}^i$.

In the case of error modelling, the use of a first order development for η and assuming that the normal coordinates are known, to simplify the writing, we obtain:

$$\left\{a\,[\mathbf{J}x(\mathbf{q}^i)-\mathbf{J}x(\mathbf{q}^j)] + b\,[\mathbf{J}y(\mathbf{q}^i)-\mathbf{J}y(\mathbf{q}^j)] + c\,[\mathbf{J}z(\mathbf{q}^i)-\mathbf{J}z(\mathbf{q}^j)]\right\}\Delta\eta$$

$$= - a\,[Px(\mathbf{q}^i)-Px(\mathbf{q}^j)] - b\,[Py(\mathbf{q}^i)-Py(\mathbf{q}^j)] - c\,[Pz(\mathbf{q}^i)-Pz(\mathbf{q}^j)] \qquad [17]$$

where:
$\mathbf{J}u$, Pu are function of $\mathbf{q}$ and η.

Equation [17] gives a linear equation, for each pair of configurations $\mathbf{q}^i$ and $\mathbf{q}^j$.

Sufficient number of configurations must be used to identify $\Delta\eta$. The global system of equations will be solved iteratively and the geometric parameters will be updated after each iteration.

The calculation of the identifiable parameters can be carried out as given in Appendix III, by studying the QR decomposition of a matrix **W** calculated from [17] using sufficient random configurations where the terminal point of the robot is in a given plane.

3.4. *Calibration using distance measure*

This method needs a set of pairs of configurations where the Cartesian distance between the terminal points of each pair is also given. Thus an external sensor measuring the distance is required.

Suppose two configurations $\mathbf{q}^i$ and $\mathbf{q}^j$, where the measured Cartesian distance between the position of their terminal points is D_r^j, thus we can write:

$$[Px(\mathbf{q}^j,\eta_r) - Px(\mathbf{q}^i,\eta_r)]^2 + [Py(\mathbf{q}^j,\eta_r) - Py(\mathbf{q}^i,\eta_r)]^2 + [Pz(\mathbf{q}^j,\eta_r) - Pz(\mathbf{q}^i,\eta_r)]^2 = (D_r^j)^2 \qquad [18]$$

If the values of the geometric parameters in the model $\boldsymbol{\eta}$ are different than the real values $\boldsymbol{\eta}_r$, using a first order development, we obtain:

$$\{2[Px(\mathbf{q}^j)-Px(\mathbf{q}^i)][\mathbf{J}x(\mathbf{q}^j)-\mathbf{J}x(\mathbf{q}^i)] + 2[Py(\mathbf{q}^j)-Py(\mathbf{q}^i)][\mathbf{J}y(\mathbf{q}^j)-\mathbf{J}y(\mathbf{q}^i)] + 2[Pz(\mathbf{q}^j)-Pz(\mathbf{q}^i)][\mathbf{J}z(\mathbf{q}^j)-\mathbf{J}z(\mathbf{q}^i)]\}\, \Delta\eta = (D_r^j)^2 - (D^j)^2 \qquad [19]$$

where:

D^j is the distance between point j and point i using the nominal model,

$$[Px(\mathbf{q}^j,\eta) - Px(\mathbf{q}^i,\eta)]^2 + [Py(\mathbf{q}^j,\eta) - Py(\mathbf{q}^i,\eta)]^2 + [Pz(\mathbf{q}^j,\eta) - Pz(\mathbf{q}^i,\eta)]^2 = (D^j)^2$$

Relation [19] can be written under the standard form:

$$\mathbf{W}(\mathbf{q}^i,\mathbf{q}^j,\eta)\, \Delta\eta = (D_r^j)^2 - (D^j)^2 \qquad [20]$$

Equation [20] gives a linear equation in $\Delta\eta$. The global system of equations will be solved to get $\Delta\eta$ by using a sufficient number of pairs of configurations.

The calculation of the identifiable parameters can be carried out as given in Appendix III, by studying the QR decomposition of a matrix **W** calculated using m random points in a given plane.

4. Description of GECARO

GECARO runs on PC computers under Windows 95/98 or Windows NT. A friendly user interface has been developed using C^{++} language, while the calculation algorithms have been developed using Matlab 5.2 software. The main page of GECARO is given in [Figure 2], on which we recognize the parameters defining the robot and the following menus:

Robot – Position measure – Frame measure – Point link – Frame link – Plane link1 – Plane link2 – Distance measure

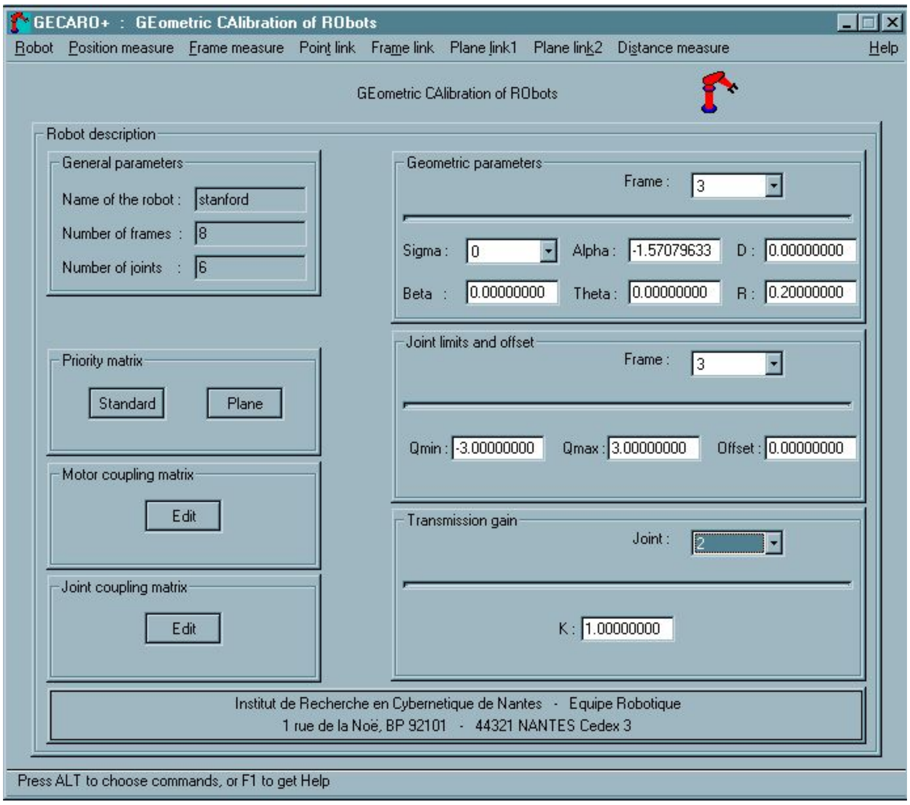

Figure 2. *Main page of GECARO*

Robot menu is used to edit the parameters of a new robot or to select a previously defined robot. The following functions are available under this menu:

New – Open – Save – Save as – Quit – Help

Each of the other menus treats a calibration method. Each of them has the following principal functions:

Identifiable parameters – Points generation – Identification

In the following we detail these functions:

– **Identifiable parameters**: This function calculates the identifiable geometric parameters (base parameters) for the selected robot and the selected method.

The identifiable parameters can be obtained from the classical parameters by eliminating those which have no effect on the observation matrix or those whose corresponding elements in the priority matrix are equal to zero, and by regrouping the parameters whose corresponding columns in the observation matrix are not independent. These parameters are calculated using QR decomposition of an observation matrix which can be calculated using either random values, or by using real experimental data. This function can also be used to test if some experimental data are sufficiently exciting (giving a good condition number for the observation matrix, and giving the same identifiable parameters as those obtained using random values).

It is to be noted that the identifiable parameters are not uniquely defined, the QR method will give as base parameters those corresponding to the first b independent columns of the matrix **W**. This function will order the columns of the matrix **W** and the geometric parameters vector as function of the priority number associated to each parameter. It is more practical to identify, if possible, the parameters which can be updated easily in the control system without changing the symbolic direct and inverse geometric models of the robot. Therefore we use the following rules to define default priorities as input to this function (the user can define arbitrarily his input priority matrix):

- the highest priority is equal to 5 and corresponds to joint offsets and motor transmission gains.

- the parameters r_j and d_j which are not equal to zero will take the priority 4,

- the parameters α_j and θ_j which are not equal to $\pm k\pi/2$, with k integer, will take the priority 3,

- the parameters of the base location or the end-effector frames, will get the priority 2,

- the rest of parameters will get the priority 1.

The user can put equal to zero the elements of the priority matrix corresponding to the parameters which he does not like to identify, either because their values are known accurately, or because he cannot make use of their identified values.

The output of this function is the regrouping relations of the identifiable parameters and a new priority matrix which is similar to the input one, but the priorities of the non identifiable parameters are put equal to zero.

– **Points generation**: this function generates a file which can be used to test and simulate the functions of the corresponding method. In general the generated file contains two matrices:

Qm: contains the motor variables of the generated configurations, they satisfy the constraint of the selected method.

X: contains the Cartesian coordinates of position or location corresponding to **Qm**.

The number of configurations is given by the user, this number must be chosen such that sufficient number of equations can be generated.

– **Identification**: This function gives the solution of the identified geometric parameters and the precision of the obtained solution. Two input files are needed:

- the file containing the priority matrix,

- the file containing the input data required to carry out the identification. For instance in the classical method it must contain the matrix **Qm** of motor variables and the matrix **X** of corresponding Cartesian coordinates, while in the planar method only a matrix **Qm** is needed. In the case of position or frame link methods the number of configurations of each set, giving the same Cartesian position and location, will be given in a vector (whose dimension is equal to the number of sets).

5. Example

In this example we present the identifiable parameters of the Stanford manipulator shown in [Figure 3] and whose geometric parameters are given in [Table 1]. The robot parameters are defined using the menu "Robot" and the different boxes of the main page of GECARO [Figure 2]. The parameters of [Table 1] correspond to 6 joints (frames 1,...,6) and two fixed frames (frame 0 and frame 7). The identifiable parameters of the different methods are given in [Table 2], where:

– The parameters denoted by "0" means non identifiable parameters which have no effect on the model,

– The parameters denoted by "n" means non identifiable parameters whose effect are regrouped on the other parameters,

– The other parameters are identifiable.

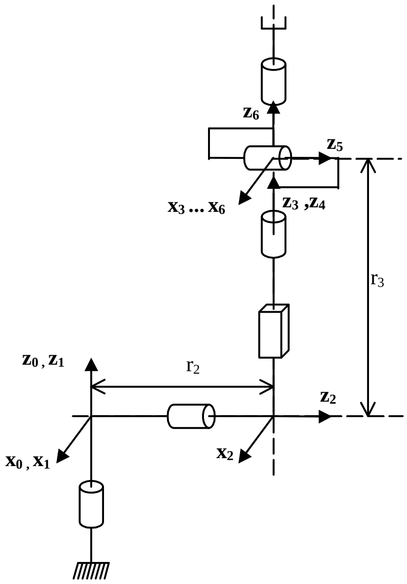

Figure 3. *Stanford manipulator*

j	σ_j	α_j	d_j	θ_j	r_j	β_j	K_j
0	2	0	0	$\pi/2$	0.5	0	0
1	0	0.1	0	θ_1	0	0	1
2	0	$-\pi/2$	0	θ_2	0.2	0	1
3	1	$\pi/2$	0	0	r_3	0	1
4	0	0	0	θ_4	0	0	1
5	0	$-\pi/2$	0	θ_5	0	0	1
6	0	$\pi/2$	0	θ_6	0	0	1
7	2	1.3	0.2	$\pi/2$	0.1	0	0

Table 1. *The geometric parameters of the Stanford (RRPRRR) manipulator (units are given in meters for the distances and radians for the angles).*

Parameter	Position Measure	Location Measure	Point Link	Frame Link	Plane Link 1 (Known Plane)	Plane Link 1 (Unknown Plane)	Plane Link 2 (Known Plane)	Plane Link 2 (Unknown Plane)	Distance Measure
$\theta 0$			0	0	n	n	n	n	0
r0			0	0		n	0	0	0
$\alpha 1$			0	0		n		n	0
d1			0	0	n	n	0	0	0
$\theta 1$			0	0		n		n	0
r1			0	0	n	n	0	0	0
$\beta 1$	n	n	0	0	n	n	n	n	0
$\alpha 2$									
d2									
$\theta 2$									
r2				n					
$\beta 2$	n	n	n	n	n	n	n	n	n
$\alpha 3$									
d3									
$\theta 3$	n	n	n	n	n	n	n	n	n
r3									
$\beta 3$	n	n	n	n	n	n	n	n	n
$\alpha 4$									
d4	n	n	n	n	n	n	n	n	n
$\theta 4$									
r4	n	n	n	n	n	n	n	n	n
$\beta 4$									
$\alpha 5$									
d5									
$\theta 5$									
r5									
$\beta 5$	n	n	n	n	n	n	n	n	n
$\alpha 6$									
d6									
$\theta 6$				0					
r6			n	0	n	n	n	n	
$\beta 6$	n	n	n	0	n	n	n	n	n
$\alpha 7$	n		n	0	n	n	n	n	n
d7				0					
$\theta 7$	0		0	0	0	0	0	0	0
r7	n		n	0	n	n	n	n	n
$\beta 7$	n	n	n	0	n	n	n	n	n
k1									
k2									
k3									
k4									
k5									
k6									
Total	31	34	24	21	27	24	26	24	25

Table 2. *Identifiable Parameters of the Stanford manipulator*

6. Conclusion

This paper presents the software package GECARO which is devoted for the simulation and the practical calibration of the geometric parameters of robots. The parameters defining the robot, the calibration methods, and the main menus of this package are described in the paper. GECARO is running on PC computers and developed using MATLAB; any general serial robot can be treated directly. This system contains classical calibration methods which require external sensors to measure the terminal link location, as well as autonomous calibration methods which need only joint position sensors. For all the presented methods the identifiable parameters are determined using a numerical method based on the QR decomposition. GECARO can provide for all of developed methods appropriate robot configurations to simulate the capacity and the converge of each method. It has been seen that the identification using a linearized model and by updating the observation matrix and the error vector after each iteration converges in about 5 iterations even if the errors in the geometric parameters are too big (giving up to 60 cm error on the terminal point position).

The future development concerns the calibration of serial robots having non-negligible elasticity on the joints or on the links and also concerns the identification of the geometric parameters of parallel robots.

Acknowledgements

This work has been realized thanks to the financial support of C.R.I.T.T. "Centre Régional d'Innovation et de Transfert de Technologie des Pays de la Loire Productique".

REFERENCES

[BEN 91] BENNET, HOLLERBACH J., "Autonomous-calibration of single-loop, closed kinematic chains formed by manipulators with passive end points constraints". IEEE Trans. on *Robotics and Automation*, vol. RA-7(5), p. 597-606, 1991.

[CAE 93] CAENEN J.L., "Contribution à l'identification de paramètres géométriques et non géométriques d'un modèle de robot. Application à l'amélioration de la précision de positionnement statique". Doctorat Thesis, Valenciennes, France, 1993.

[DAM 96] DAMAK M., "Théorie et instrumentation pour l'étalonnage statique des robots: vers une programmation hors-ligne industriellement plus efficace". Doctorat Thesis, ENSAM, Lille, France, 1996.

[DON 79] DONGARRA J.J., MOLER C.B., BUNCH J.R., STEWART G.W., "LINPACK User's Guide". Philadelphia, PA: SIAM, 1979.

[EDW 94] EDWARDS CH., GALLOWAY L., "A Single-Point Calibration Technique for a Six Degree-Of-Freedom Articulated Arm", Int. J. of *Robotic Research* 13, n° 3, p. 189-199, June 1994.

[GUY 95] GUYOT G., "Contribution à l'étalonnage géométrique des robots manipulateurs". Doctorat Thesis, Nice, France, 1995.

[HAY 88] HAYATI S.A., "Robot arm geometric link calibration". Proc. IEEE, *Decision and Control Conference*, p. 798-800, 1988.

[HOL 89] HOLLERBACH J., "A Survey of Kinematic Calibration". *Robotics Review* n° 1, edited by Khatib, Craig and Lozano-Perez, Cambridge, MA, MIT Press, p. 207-242, 1989.

[HOL 96] HOLLERBACH J.M., WAMPLER C.W., "The calibration index and taxonomy of kinematic calibration methods". Int. J. of *Robotics Research*, vol. 14, p. 573-591, 1996.

[IKI 97] IKITS M., HOLLERBACH J.M., "Kinematic Calibration Using a plane Constraint". IEEE *Robotics and Automation Conference*, p. 3191-3196, 1997.

[KHA 86] KHALIL W., KLEINFINGER J.F., "A new geometric notation for open and closed loop robots". Proc. IEEE *Robotics and Automation Conference*, San francisco, p. 1174–1180, 1986.

[KHA 91a] KHALIL W., GAUTIER M., ENGUEHARD CH., "Identifiable parameters and optimum configurations for robots calibrations". *Robotica*, vol. 9, p. 63-70, 1991.

[KHA 91b] KHALIL W., GAUTIER M., "Calculation of the identifiable parameters for robots calibration". 9th IFAC/IFORS Symposium on *Identification and System Parameter Estimation*, p. 888-892, Budapest, 1991.

[KHA 95] KHALIL W., GARCIA G., DELAGARDE J.F., "Calibration of the geometric parameters of robots without external sensors". IEEE *Robotics and Automation Conference*, p. 3039-3044, 1995.

[KHA 96] KHALIL W., LEMOINE PH., GAUTIER M., "Autonomous calibration of robots using planar points". Proc. 6th Int. Symp. on *Robotics and Manufacturing*, WAC'96, vol. 3, Montpellier, p. 383-388, mai 1996.

[KHA 99] KHALIL W., DOMBRE E., "Modélisation, identification et commande des robots". Hermes, Paris, 1999.

[LAW 74] LAWSON C.L., HANSON R.J., "Solving Least Squares Problems". NJ: Englewood Cliffs, Prentice-Hall, 1974.

[MAU 96] MAURINE P., "Développement et mise en œuvre de méthodologies d'étalonnage de robots manipulateurs industriels". Doctorat Thesis, Montpellier, France, december 1996.

[NAK 95] NAKAMURA H., ITAYA T., YAMAMOTO K., KOYAMA T., "Robot autonomous error calibration method for off-line programming system". IEEE *Robotics and Automation Conference*, p. 1775-1783, 1995.

[ROT 87] ROTH S., MOORING B.W., RAVANI B., "An overview of robot calibration". IEEE J. *Robotics and Automation*, vol. RA-3, n° 5, p. 377-385, 1987.

[TAN 94] TANG G.R., LIEU L.S., "A study of three robot calibration methods based on flat surfaces". *Mechanism and Machine Theory*, vol. 29, n° 2, p. 195-206, 1994.

[WAM 88] WAMPLER C.W., HOLLERBACH J.M., ARAI T., "An implicit loop method for kinematic calibration and its application to closed-chain mechanisms". IEEE Trans. on *Robotics and Automation*, vol. 11, n° 5, p. 710-724, 1988.

[WHI 86] WHITNEY D.E., LOZINSKI C.A., ROURKE J.M., "Industrial robot forward calibration method and result". ASME J. of *Dynamics Systems Measurements, and Control*, vol. 108, p. 1-8, 1986.

[ZHO 95] ZHONG X.-L., LEWIS J.M., "A new method for autonomous robot calibration". IEEE *Robotics and Automation Conference*, p. 1790-1795, 1995.

Wisama Khalil *received the "Doctorat d'Etat" degree from Montpellier (France) in 1978. Since 1983 he is professor of robotics in the "Ecole Centrale de Nantes". He is responsible of the robotics team of the "IRCyN UMR-CNRS n° 6597". His research interests include kinematics, dynamics, identification and control of robots.*

Philippe Lemoine *obtained the "Ingénieur CNAM" degree in Nantes (France) in 1991. Since 1993 he is a research engineer in the "Ecole Centrale de Nantes". He essentially works with the robotics team of the "IRCyN UMR-CNRS n° 6597".*

Appendix I

Calculation of the difference vector between transformation matrices

Let us assume two transformation matrices $\mathbf{T}^i$ and $\mathbf{T}^j$ defining frames i and j with respect to the same reference frame such that:

$$\mathbf{T}^i = \begin{bmatrix} \mathbf{s}^i & \mathbf{n}^i & \mathbf{a}^i & \mathbf{P}^i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{T}^j = \begin{bmatrix} \mathbf{s}^j & \mathbf{n}^j & \mathbf{a}^j & \mathbf{P}^j \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

where **s, n, a** represent the unit vectors along the axes x, y and z respectively.

The vector $\Delta\mathbf{x}$ representing the position and orientation difference between frames i and j is equal to [KHA 99]:

$$\Delta\mathbf{x} = \begin{bmatrix} \mathbf{P}^i - \mathbf{P}^j \\ \delta_{i,j} \end{bmatrix} \qquad [21]$$

where:

$\mathbf{P}^i$ - $\mathbf{P}^j$ is the differential position vector between frames i and j,

$\delta_{i,j}$ is the differential rotation vector between the orientation of frames i and j. It can be calculated by one of the following methods:

– First method:

$$\delta_{i,j} = \frac{1}{2} [\mathbf{s}^j \times \mathbf{s}^i + \mathbf{n}^j \times \mathbf{n}^i + \mathbf{a}^j \times \mathbf{a}^i] \qquad [22]$$

where x denotes the vector product.

– Second method:

Let the orientation matrix in configuration j be defined as: $\mathbf{A}^j = \begin{bmatrix} \mathbf{s}^j & \mathbf{n}^j & \mathbf{a}^j \end{bmatrix}$, thus:

$$\mathbf{A}_d = \mathbf{A}^i (\mathbf{A}^j)^T = \mathbf{Rot}(\mathbf{u},\alpha) \qquad [23]$$

Where $\mathbf{Rot}(\mathbf{u},\alpha)$ is the (3x3) orientation matrix representing a rotation α about the vector $\mathbf{u}$.

The differential orientation vector is given by:

$$\delta_{i,j} = \mathbf{u}.\alpha \qquad [24]$$

Where the values of $\mathbf{u}$ and α are obtained from [23] as function of the elements of the matrix $\mathbf{A}_d$.

If α is small we can write:

$$\delta_{i,j} = \mathbf{u}.\sin\alpha = \frac{1}{2} \begin{bmatrix} A(2,3) - A(3,2) \\ A(1,3) - A(3,1) \\ A(2,1) - A(1,2) \end{bmatrix} \qquad [25]$$

where A(i,j) is the (i,j) element of the matrix $\mathbf{A}_d$.

It has been shown that the second method converges better than the first one.

Appendix II

Calculation of the jacobian matrix

Assuming the (4x4) transformation matrix defining frame j with respect to the fixed frame as [KHA 91a] [KHA 99]:

$$^{-1}\mathbf{T}_j = \begin{bmatrix} \mathbf{s}_j & \mathbf{n}_j & \mathbf{a}_j & \mathbf{P}_j \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad [26]$$

The columns of the matrix **J** corresponding to the parameters α_j, d_j, θ_j, r_j, β_j, k_j can be calculated as follows [KHA 99]:

$$\mathbf{j}\alpha_j = \begin{bmatrix} \mathbf{s}_{j-1} \mathrm{x} \mathbf{L}_{j-1,n+1} \\ \mathbf{s}_{j-1} \end{bmatrix}, \qquad \mathbf{j}d_j = \begin{bmatrix} \mathbf{s}_{j-1} \\ \mathbf{0}_{(3x1)} \end{bmatrix},$$

$$\mathbf{j}\theta_j = \begin{bmatrix} \mathbf{a}_j \mathrm{x} \mathbf{L}_{j,n+1} \\ \mathbf{a}_j \end{bmatrix}, \qquad \mathbf{j}r_j = \begin{bmatrix} \mathbf{a}_j \\ \mathbf{0}_{(3x1)} \end{bmatrix},$$

$$\mathbf{j}\beta_j = \begin{bmatrix} \mathbf{n}_{j-1} \mathrm{x} \mathbf{L}_{j-1,n+1} \\ \mathbf{n}_{j-1} \end{bmatrix}, \qquad \mathbf{j}k_j = \mathbf{j}q_j \frac{\partial \mathbf{q}}{\partial K_j} \qquad [27]$$

Where:

x denotes the vector product,

$\mathbf{L}_{j,n+1}$ is the (3x1) position vector between the origin of frame j and the origin of frame n+1 equal to $\mathbf{P}_{n+1}$ - $\mathbf{P}_j$,

$\mathbf{0}_{(3x1)}$ is the (3x1) zero vector,

$\mathbf{j}q_j$ is equal to $\mathbf{j}\theta_j$ if joint j rotational or $\mathbf{j}r_j$ if joint j prismatic.

All the vectors of equations [27] are referred to the measuring fixed frame.

Appendix III

Calculation of the identifiable parameters

Let us consider the following system of overconstrained linear equations:

$$\Delta\mathbf{X} = \mathbf{W}\,\Delta\eta \qquad [28]$$

where **W** is (rxc) matrix with r>>c. If b is the rank of **W**, we can write:

$$\Delta \mathbf{X} = [\mathbf{W}_1 \ \mathbf{W}_2] \begin{bmatrix} \Delta\eta_1 \\ \Delta\eta_2 \end{bmatrix} \qquad [29]$$

where:
$\mathbf{W}_1$ represents b independent columns of **W**,
$\mathbf{W}_2$ represents the other (c-b) columns of **W**.

We can write:

$$\mathbf{W}_2 = \mathbf{W}_1 \ \beta \qquad [30]$$

with β is (bx(c-b)) matrix with constant elements.

Using relation [30] , equation [28] can be written as:

$$\Delta \mathbf{X} = \mathbf{W}_1 \ \Delta\eta_b \qquad [31]$$

with:

$$\Delta\eta_b = \Delta\eta_1 + \beta \ \Delta\eta_2 \qquad [32]$$

The solution of equation [31] will give $\Delta\eta_b$ which is called the identifiable (or base) parameters vector. The matrix β is not needed in the identification process.

Numerically, the study of the base parameters is equivalent to study the space spanned by the columns of the matrix **W**.

Using QR decomposition, **W** can be decomposed as [DON 79] [LAW 74]:

$$\mathbf{Q}^T \ \mathbf{W} = \begin{bmatrix} \mathbf{R} \\ \mathbf{0}_{(r-c)xc} \end{bmatrix} \qquad [33]$$

Q is a (rxr) orthogonal matrix,
R is a (cxc) upper triangular matrix,
$\mathbf{0}_{ixj}$ is the (ixj) matrix of zeros.

Theoretically the non identifiable parameters are those whose corresponding elements on the diagonal of the matrix **R** are equal to zero. Thus assuming τ as the numerical zero, if the element $|R_{ii}| \leq \tau$, the corresponding parameter $\Delta\eta_i$ is not identifiable. The numerical zero τ can be taken as [LAW 74]:

$$\tau = c.\varepsilon.\max |R_{ii}| \qquad [34]$$

Where ε is the machine precision.

To calculate the identifiable (or base) parameters as function of the standard parameters let us permute the columns of **W** such that the first b columns are independent:

$$[\mathbf{W.P}] = \begin{bmatrix} \mathbf{W}_1 & \mathbf{W}_2 \end{bmatrix} \qquad [35]$$

where:
P is a permutation matrix,
$\mathbf{W}_1$ represents b independent columns of **W**,
$\mathbf{W}_2$ represents the (c-b) dependent columns of **W**.

The QR decomposition of (**W.P**) gives:

$$\begin{bmatrix} \mathbf{W}_1 & \mathbf{W}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 \end{bmatrix} \begin{bmatrix} \mathbf{R}_1 & \mathbf{R}_2 \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$$

Where $\mathbf{R}_1$ is a (bxb) regular matrix. Then it comes:

$$\mathbf{W}_2 = \mathbf{W}_1 \mathbf{R}_1^{-1} \mathbf{R}_2 \qquad [36]$$

Thus from [30] and [36] we get:

$$\beta = \mathbf{R}_1^{-1} \mathbf{R}_2 \qquad [37]$$