Robotica (1997) volume 15, pp 395–398. Printed in the United Kingdom © 1997 Cambridge University Press

Parallel topology robot calibration

A.B. Lintott and G.R. Dunlop

Department of Mechanical Engineering, University of Canterbury, Christchurch (New Zealand)
Email: R.Dunlop@canterbury.ac.nz

SUMMARY

A calibration method for a Stewart platform has been developed as part of a project aimed at developing a calibration method for a Delta robot. The Delta has 3 degrees of freedom (DOF) but is more complex than the Stewart platform for calibration purposes because an extra link is inserted in each kinematic chain between the base and the Nacelle member.

KEYWORDS: Robot calibration; Stewart platform; Parallel topology.

INTRODUCTION

Parallel topology robots have not yet made a significant impact in industrial applications and this is probably why there has been very little work on the calibration of fully parallel manipulators. The Stewart platform[1] was originally proposed as a flight simulator platform, and is widely used for this today. McCallion and Pham[2] used the mechanism as a 6 DOF platform for robotic assembly, while Dunlop et al.[3] applied the mechanism to satellite tracking. The Stewart platform has also been considered for use as a machine tool.[4,5]

The Delta robot was first proposed by Clavel.[6,7] The robot is a simplification of a 6 DOF mechanism first proposed by Hunt[8] which, in turn, bears a resemblance to the Stewart platform. The Delta robot has 3 translational degrees of freedom and is actuated by 3 kinematic chains consisting of a rotary arm in series with 2 ball-jointed parallel arms. A number of prototypes have been constructed at the Swiss Federal University of Lausanne (EPFL) and a production version now exists.

The work presented in this paper is at an early stage of development. The goal is to develop an effective calibration model for the full Delta robot. Thus far, work has focused on developing a calibration model for the Stewart platform because it is similar in form to a sub-structure of the Delta robot. In this paper, the model for a variant of the Stewart platform with fixed leg lengths and mobile base joints is developed as an intermediate step to the Delta structure.

NOTATION

In this paper, vector quantities are represented in lower case bold type (e.g. $\mathbf{p}_i$), matrices are represented in upper case bold (e.g. $\mathbf{R}$), and scalars are shown in italic (e.g. t_i). Vector valued functions are denoted with an arrow above an upper case italic symbol (e.g. $\vec{T}$). Unit vectors are identified with a "hat" or caret above a vector symbol (e.g. $\hat{\mathbf{r}}$) and error parameters are indicated by the symbol δ before the appropriate symbol. For instance, the vector of error parameters relating to the vector $\mathbf{p}_i$ is the same dimension as $\mathbf{p}_i$ and is written $\delta\mathbf{p}_i$.

THE FIXED-LEG STEWART PLATFORM

This variant of the Stewart platform structure is not proposed as a practical design. It serves as a simplified structure with similar properties to the lower part of the Delta robot.

In this design, the Stewart platform has 6 inextensible legs that are connected by ball joints to the nacelle and the base (Figure 2). The nacelle ball joints are fixed in position relative to the nacelle coordinate frame while the base ball joints are allowed to move along straight rails. The robot is actuated by sliding the base ball joints along these rails a distance b_i $(i = 1 \cdots 6)$ from the base point of the rail. In the Delta robot model, the 6 straight rails will be replaced by 3 paired circular arcs representing the loci of the 3 actuator arms.

The error model allows for deviations in the ball joint positions and in the lengths of the legs, giving 42 error parameters in total. For this simplified model, the rails are assumed to be perfectly straight and accurately located.

1. Modelling

For the fixed-leg Stewart platform, the loop closure condition is given by (1). The meanings of the symbols in (1) may be inferred from Figure 2.

$$\mathbf{x} + \mathbf{R}(\mathbf{q}_i + \delta\mathbf{q}_i) - (\mathbf{p}_i + \delta\mathbf{p}_i + b_i\hat{\mathbf{r}}_i) - (t_i + \delta t_i)\hat{\mathbf{t}}_i = 0 \quad (1)$$

Matrix $\mathbf{R}$ is the 3 by 3 rotation matrix that converts the nacelle basis vectors into the base frame basis fectors. The *inverse geometric solution* expresses the actuator coordinates as a function of the endpoint position. If the nacelle is fixed in space and leg i is allowed to swing from the nacelle, then the locus of the base ball joint traces out a sphere. The sphere intersects the rail in 2 locations and the convention is adopted that the point of intersection furthest from base platform centre is taken to be the attachment point. Given $\mathbf{x}$, the centre of the spherical locus for leg i is located in the base coordinate frame by vector $\mathbf{c}_i$.

$$\mathbf{c}_i = \{\mathbf{x} + \mathbf{R}(\mathbf{q}_i + \delta\mathbf{q}_i)\} \quad (2)$$

Point $\mathbf{s}$ lies on the sphere and on the line representing the rail.

$$(\mathbf{s} - \mathbf{c})^{\mathrm{T}}(\mathbf{s} - \mathbf{c}) = (t_i + \delta t_i)^2 \quad (3)$$

$$\mathbf{s} = \mathbf{p}_i + \delta\mathbf{p}_i + b_i\hat{\mathbf{r}} \quad (4)$$

https://doi.org/10.1017/S0263574797000489 Published online by Cambridge University Press

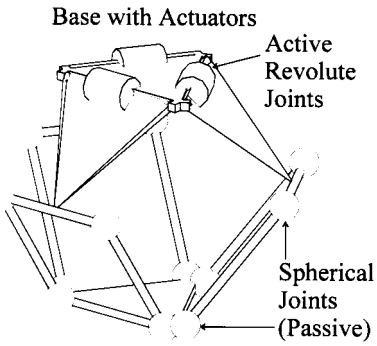

Fig. 1. Schematic diagram of the Delta robot

The sphere is described by (3) and the line is described by (4). By substitution,

$$(b_i\hat{\mathbf{r}} - \mathbf{h}_i)^{\mathbf{T}}(b_i\hat{\mathbf{r}} - \mathbf{h}_i) = (t_i + \delta t_i)^2 \tag{5}$$

where:

$$\mathbf{h}_i = \mathbf{c} - \mathbf{p}_i - \delta\mathbf{p}_i \tag{6}$$

hence:

$$b_i = \hat{\mathbf{r}}_i^{\mathbf{T}}\mathbf{h}_i \pm \sqrt{(\hat{\mathbf{r}}_i^{\mathbf{T}}\mathbf{h}_i)^2 - 2\hat{\mathbf{r}}_i^{\mathbf{T}}\mathbf{h}_i(\mathbf{h}_i^{\mathbf{T}}\mathbf{h}_i - t_i^2 - \delta t_i^2)} \tag{7}$$

Substituting (4) into (3), a quadratic equation (5) is generated in terms of b_i. Provided that $\mathbf{x}$ is within the workspace of the robot, (5) will generally have 2 real roots (7). Following the convention mentioned above, the largest root is the one that is selected.

The *direct geometric solution* expresses the endpoint position as a function of the actuator coordinates, robot geometry, and error parameters. While the direct geometric solution has been solved analytically,[9] it is much more computationally efficient to use a numerical solution algorithm such as the Newton-Raphson algorithm.[10]

$$\mathbf{e} = [\delta t_1 \cdots \delta t_6 (\delta\mathbf{q}_1)^{\mathrm{T}} \cdots (\delta\mathbf{q}_6)^{\mathrm{T}} (\delta\mathbf{p}_1)^{\mathrm{T}} \cdots (\delta\mathbf{p}_6)^{\mathrm{T}}]^{\mathbf{T}} \tag{8}$$

where

$$\begin{aligned}(\delta\mathbf{q}_i)^{\mathrm{T}} &= [\delta q_{1i} \quad \delta q_{2i} \quad \delta q_{3i}],\\ (\delta\mathbf{p}_i)^{\mathrm{T}} &= [\delta p_{1i} \quad \delta p_{2i} \quad \delta p_{3i}]\end{aligned} \tag{9}$$

$$\mathbf{J} \equiv \frac{\partial\mathbf{x}}{\partial\mathbf{e}} = \left[\frac{\partial\mathbf{x}}{\partial(\delta t_i)} \frac{\partial\mathbf{x}}{\partial(\delta\mathbf{q}_i)} \frac{\partial\mathbf{x}}{\partial(\delta\mathbf{p}_i)}\right], \quad 1 \le i \le 6. \tag{10}$$

The *differential solution* expresses rates of change in the endpoint coordinates as a function of changes in the error parameters. If the error parameters are grouped into a single 42 by 1 partitioned vector as in (8) then the direct Jacobian matrix is defined as in (10).

$$\delta t_i = \|\mathbf{x} + \mathbf{R}(\mathbf{q}_i + \delta\mathbf{q}_i) - (\mathbf{p}_i + \delta\mathbf{p}_i + b_i\hat{\mathbf{r}}_i) - t_i\hat{\mathbf{t}}_i\| \tag{11}$$

$$\vec{T}(\mathbf{x}, \mathbf{e}) = [\delta t_1 \quad \delta t_2 \quad \delta t_3 \quad \delta t_4 \quad \delta t_5 \quad \delta t_6]^{\mathrm{T}} \tag{12}$$

The terms in (10) are not simple to express or evaluate, but the loop closure equation (1) may be rearranged as in (11) and the δt_i collected into a vector function (12). It may be shown that

$$\frac{\partial\mathbf{x}}{\partial(\delta t_i)} = \left\{\frac{\partial}{\partial\mathbf{x}}\vec{T}(\mathbf{x}, \mathbf{e})\right\}^{-1} = \Delta \tag{13}$$

hence

$$\mathbf{J} = \left[\Delta, \Delta \cdot \frac{\partial}{\partial(\delta\mathbf{q}_i)}\vec{T}(\mathbf{x}, \mathbf{e}), \quad \Delta \cdot \frac{\partial}{\partial(\delta\mathbf{p}_i)}\vec{T}(\mathbf{x}, \mathbf{e})\right] \tag{14}$$

which is easier to evaluate.

In the following expression, the subscript represents the ordinal number of the measurement points. Each nominal measurement position, $\mathbf{x}_j^{\mathrm{nom}}$, is associated with a Jacobian matrix $\mathbf{J}_j$ evaluated at $\mathbf{x}_j^{\mathrm{nom}}$. The *identification Jacobian* $\mathbf{M}$ is defined as

$$\mathbf{M} = [\mathbf{J}_1^{\mathrm{T}} \quad \mathbf{J}_2^{\mathrm{T}} \cdots \mathbf{J}_n^{\mathrm{T}}]^{\mathbf{T}} \tag{15}$$

2. *Measurement and Identification*

For simplicity, the measurement device in the example is assumed to be capable of measuring all 6 components of the endpoint position expressed in the world frame. A set of n measurement points in the workspace is chosen and readings are taken from the measurement device when the robot is moved to these points. It is appropriate to consider which set of measurement points in the

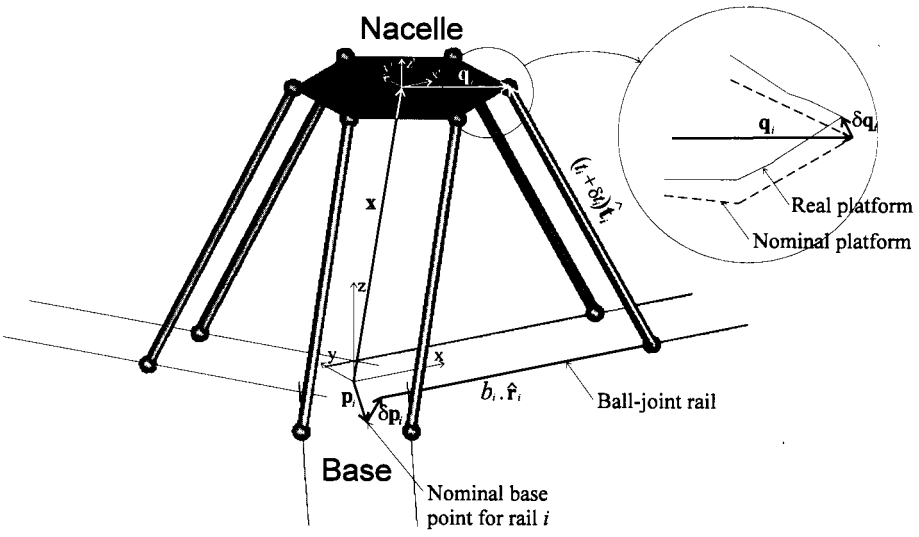

Fig. 2. The fixed-leg Stewart Platform.

https://doi.org/10.1017/S0263574797000489 Published online by Cambridge University Press

robot's workspace guarantees a complete solution for the error parameters $\underline{\delta}$. This is a question of *observability*.

$$O(\mathbf{M}) = \frac{\sqrt[L]{\sigma_1 \cdot \sigma_2 \cdots \sigma_L}}{\sqrt{n}} \tag{16}$$

where $\sigma_1, \sigma_2, \ldots, \sigma_L$ are the singular values of the identification Jacobian.

Menq and Borm[11] present an observability index (16) based on the singular value decomposition of the identification Jacobian **M** (15). It is normally advantageous to choose measurement points that maximise the observability index. Borm and Menq[12] use a numerical optimisation procedure to search for optimum measurement positions. For the example presented here, a similar numerical optimisation was performed using a steepest ascent method. It was observed that the measurement points generally migrate towards the edges of the workspace and toward singular configurations as the optimisation progresses. This is because the best increase in the Menq and Borm observability index is achieved by increasing the greatest singular value of the identification Jacobian. This is not a great problem with serial manipulators since there are no singularities of the type where mobility is gained however parallel robots often possess singularities of this type. When the robot is moved close to a singular configuration some terms of the direct Jacobian matrix become very large. This appears in the singular value decomposition as a very large greatest singular value. The other singular values are less prone to inflation. The near-singular points caused numerical range difficulties during the identification phase. Similar difficulties were encountered when using either the least singular value or the condition number as the observability index. A possible solution to this problem would be to set an arbitrary limit on the 2-norm of the direct Jacobian matrix of the manipulator at each measurement point. This is equivalent to creating an imaginary sphere around each singular point which the measurement points cannot move into during optimisation.

The data for this example were generated numerically. A set of values was chosen for the $\underline{\delta}$ error parameters. The object was to recover these values from the data using the identification technique.

Once the measurement data has been obtained, the values of the error parameters $\underline{\delta}$ must be determined. This is a model fitting exercise. The simplest approach to solving for $\underline{\delta}$ is to use a linear least squares solution. Linear least squares estimation works satisfactorily if the identification Jacobian is well conditioned, the observability is relatively high, and the calibration model is sufficiently linear. For situations where the condition number is poor, singular value decomposition[13] is often used. For improved performance,[14] it is possible to use a weighted least squares solution which weights the contribution of each data point to the result according to the inverse of its uncertainty.

Non-linear least squares solutions are generally more computationally demanding but they often give a more accurate result. A practical technique is the Levenberg-Marquardt algorithm.[13] The technique involves minimising a cost function (17)

$$\chi^2 = \sum (\mathbf{x}_i^{\text{act}} - \bar{G}(\mathbf{x}_i^{\text{nom}}, \tilde{\mathbf{e}})) \cdot \text{diag}\,(\mathbf{C})^{-1} \tag{17}$$

where $G(\mathbf{x}_i^{\text{nom}}, \tilde{\mathbf{e}})$ is the direct geometric solution and **C** is the covariance matrix of the experimental data. Matrix **C** may be known a-priori or it may be estimated from the experimental data.

For this example, the simulated data were processed using MatLab.[15] The number of measurement points used was 50. The results for noisy and noise free data heave been analysed by both linear and non-linear methods and presented in Table I.

RESULTS

To evaluate the improvement in accuracy due to calibration, the RMS error in the endpoint position may be evaluated before and after calibration. The tabulated results were obtained using a set of randomly distributed points within the workspace, not equal to the points used for measurement.

The iterative linear least squares results were generally poor because the calibration model is significantly non-linear. The Levenberg-Marquardt result for noise free data converges to a result that is very close indeed to the actual error parameter vector, while the same method applied to noised data reduces the RMS position error in world space to about 2% of the original uncalibrated RMS error.

When measurements that have been optimised for Menq-and-Borm observability are used, a problem is encountered in that some of the measurement points are close to physical singularities of the robot. Near singularities, the iterative direct geometric solution methods are less likely to converge. Apart from that, the repeatability of the physical robot would suffer near singularities, adding noise to the data. It was necessary to move these points away from the singularities in order to achieve convergence. The optimised measurements do not provide better parameter estimates than the non-optimised measurements.

CONCLUSION

A method for robot calibration has been applied successfully to the simulated calibration of a parallel manipulator. The high degree of non-linearity in the calibration model strongly favours the use of non-linear solutions in the identification phase.

Problems were encountered when the observability criterion of Menq and Borm was applied to optimise the measurement positions. Methods for ensuring convergence of the direct geometric solution by constraining the optimised points to avoid areas of poor conditioning need to be applied.

Future work is aimed at expanding the fixed-leg Stewart platform model so that it may be applied to calibration of the Delta robot.

https://doi.org/10.1017/S0263574797000489 Published online by Cambridge University Press

Table I. A set of typical results from simulated calibration procedures

Solution method	$\underline{\delta}$ Not calibrated	$\tilde{\delta}$ Least Sqr. No noise	$\tilde{\delta}$ Least Sqr. Noise	$\tilde{\delta}$ Lev.-Marq. No noise	$\tilde{\delta}$ Lev.-Marq. Noise	$\tilde{\delta}$ Lev.-Mar. No noise Optimised	$\tilde{\delta}$ Lev.-Mar. Noise Optimised
RMS Error							
Position (mm)	4·33	0·0774	0·175	$1{\cdot}38\times10^{-5}$	0·082	$3{\cdot}65\times10^{-5}$	0·086
Orientation (rad)	0·0546	$2{\cdot}80\times10^{-4}$	$1{\cdot}13\times10^{-3}$	$5{\cdot}87\times10^{-8}$	$1{\cdot}36\times10^{-4}$	$1{\cdot}92\times10^{-7}$	$1{\cdot}31\times10^{-4}$

REFERENCES

1. D. Stewart, "A Platform with Six Degrees of Freedom" *Proc. IMechE* (*London*) **180,** Part 1, No. 15 371–386 (1965).
2. H. McCallion, and D.T. Pham, "The analysis of a six degree of freedom work station for mechanised assembly" *5th World Congress on Theory of Machines and Mechanisms* (1979). pp. 611–616.
3. G.R. Dunlop, P.J. Ellis and N.V. Afzulpurkar, "The satellite tracking keyhole problem: a parallel mechanism mount solution" *IPENZ Trans.* **20** No 1 EMCh, 1–7 (1993).
4. G. Stix, "Nice Legs" *Scientific American* p. 24 (December 24, 1995).
5. G. Rathbun, "A Stewart platform six-axis milling machine development" *ME thesis* (University of Canterbury, Christchurch, New Zealand, 1985).
6. R. Clavel, "DELTA, a Fast Robot with Parallel Geometry" *Proc. Int. Symposium on Industrial Robots* (April, 1988) pp. 91–100.
7. R. Clavel, "Conception d'un Robot Parallèle Rapide à 4 Degrés de Liberté" *Thèse No.* 925 (*Institut de Microtechnique, Ecole Polytechnique Fédérale de Lausanne,* 1991).
8. K.H. Hunt, "Structural Kinematics of In-Parallel-Actuated Robot-Arms" *ASME J. of Mechanisms, Transmissions, and Automation in Design* **105,** 705–712 (December, 1983).
9. B. Dasgupta, and T.S. Mruthyunjaya, "A Canonical Formulation of the Direct Position Kinematics Problem for a General 6-6 Stewart Platform" *Mech. Mach. Theory* **29,** No. 6, 819–827 (1994).
10. J.P. Merlet, "Direct Kinematics of Parallel Manipulators" *IEEE Trans. on Robotics & Automation* **9,** No. 6, 842–846 (December, 1993).
11. C.H. Menq, and J.H. Borm & J.Z. Lai, "Estimation and Observability Measure of Parameter Errors in a Robot Kinematic Model" *Proc. 2nd USA–Japan Symp. on Flexible Autom.* (1988) pp. 73–79.
12. J.H. Borm, and C.H. Menq, "Experimental Study of Observability of Parameter Error in Robot Calibration" *Proc. IEEE Int. Conf. on Robotics & Automation* (1989) **Vol. 1,** pp. 587–592.
13. W.H. Press et al., *Numerical Recipes in C*: *The Art of Scientific Computing* 2nd ed. (Cambridge University Press, Cambridge & New York, 1992).
14. B.W. Mooring, Z.S. Roth & M.R. Driels, *Fundamentals of Manipulator Calibration* (John Wiley & Sons, New York, 1991).
15. MathWorks Inc., "MatLab Version 4.2c.1" *Numerical analysis software package* (1994).

LIST OF SYMBOLS

Notation

In this paper, vector quantities are represented in lower case bold type (e.g. $\mathbf{p}_i$), matrices are represented in upper case bold (e.g. $\mathbf{R}$), and scalars are shown in italic (e.g. t_i). Vector valued functions are denoted with an arrow above an upper case italic symbol (e.g. $\vec{T}$). Unit vectors are identified with a "hat" or caret above a vector symbol (e.g. $\hat{\mathbf{r}}$) and error parameters are indicated by the symbol δ before the appropriate symbol. For instance, the vector of error parameters relating to the vector $\mathbf{p}_i$ is the same dimension as $\mathbf{p}_i$ and is written $\delta\mathbf{p}_i$.

https://doi.org/10.1017/S0263574797000489 Published online by Cambridge University Press