

## SELF CALIBRATION OF 3-PRS MANIPULATOR WITHOUT REDUNDANT SENSORS

S. M. O'Brien<sup>1</sup>, J. A. Carretero<sup>2</sup>, P. Last<sup>3</sup>

<sup>1</sup> *University of New Brunswick, Dept. of Mechanical Engineering, Steven.OBrien@unb.ca,*

<sup>2</sup> *University of New Brunswick, Dept. of Mechanical Engineering, Juan.Carretero@unb.ca*

<sup>3</sup> *Institute for Machine Tools and Production Eng., Technical University Braunschweig, Germany, p.last@tu-bs.de*

---

In this paper a new calibration strategy that does not require any sensors beyond those used to control actuators is applied to the 3-PRS parallel manipulator. Parallel manipulators have several advantages over their serial counterparts, but have seen limited use because of low accuracy, among other reasons. Calibration allows the kinematic model that is used to control the manipulator to be adjusted to more closely replicate the physical manipulator. The architecture and kinematics of the 3-PRS are presented, as well an explanation of this new calibration strategy. The strategy makes use of direct kinematic singularities to obtain the redundant information required for calibration. Implementation of the algorithm is accomplished via a nested series of optimization problems, each one accomplishing a simpler stage of the overall procedure. A simulated calibration is performed, and the algorithm successfully returns the exact values used to generate the test data.

**Keywords:** Parallel manipulator, calibration, singularity-based calibration.

---

## AUTO-CALIBRATION DU MANIPULATEUR 3-PRS SANS SENSEURS REDONDANTS

Dans cet article, une nouvelle stratégie de calibrage qui n'a pas besoin d'aucun senseur autre de ceux employés pour commander les actionneurs est appliquée au manipulateur parallèle 3-PRS. Les manipulateurs parallèles ont plusieurs avantages par rapport aux manipulateurs en série, mais leur utilisation est limitée à cause de leur basse exactitude, entre autres raisons. Le calibrage permet ajuster les paramètres cinématiques pour qu'ils reflètent plus exactement le manipulateur réel. L'architecture et la cinématique du 3-PRS sont présentées, aussi bien qu'une explication de cette nouvelle stratégie de calibrage. La stratégie utilise les singularités cinématiques directes pour obtenir l'information redondante nécessaire pour le calibrage. L'exécution de l'algorithme est accomplie grâce à la solution de plusieurs problèmes d'optimisation, chacune accomplissant une étape plus simple du problème global. Un calibrage simulé est utilisé pour démontrer que l'algorithme renvoie avec succès les valeurs exactes employées pour produire des essais.

**Mots clés:** Manipulateurs parallèle, calibration, calibration à base de singularités.

---

## 1 INTRODUCTION

It has been shown that, in general, kinematic manipulators (both serial and parallel) have better repeatability than absolute accuracy [1]. A position recorded in joint space can be reproduced within very small tolerances; whereas seeking arbitrary positions in task space results in significant, and often the case, unacceptable errors.

Errors such as these are caused by discrepancies between the theoretical kinematic model and the real world implementation of the manipulator. The kinematic model is the set of mathematical equations that are used to translate task space coordinates into joint space coordinates and vice versa. Because manipulators can only be controlled by changing the position of their actuators, they can only be controlled in joint space. When a discrepancy exists between the theoretical model and an actual physical model, inaccuracies should naturally be expected.

Minimizing these kind of discrepancies is the goal of all calibration procedures. Much work has been done in the field of manipulator calibration for both serial [2] and parallel [3, 4, 5] manipulators. In both cases, four separate tasks are generally required to calibrate a specific manipulator, as defined by [6]: (1) modelling, (2) measurement, (3) identification and (4) correction.

Provided a suitable kinematic model is available, the parameters of the model must be identified to perform calibration of the physical manipulator. This can be done in a variety of ways [7, 8, 9, 10]; but in all cases the measured coordinates (either in joint or task space) are compared to theoretical ones. Kinematic parameters can then be corrected to force the theoretical predictions to conform to the measured values.

Last *et al.* [11] proposed a new calibration strategy that allows parallel manipulators to be calibrated without any external sensor and without any additional joint sensors. Measured and theoretical joint space coordinates are obtained for direct kinematic singular configurations. Using the disparity between these singular poses yields a residual between coordinates. The residual can be minimized through the use of optimization strategies. The parameters of this problem are the kinematic parameters of the mechanism, and, once optimized, they will reflect the physical geometry of the manipulator.

The focus of this paper is to adapt Last *et al.*'s calibration method to the 3-PRS parallel manipulator proposed in [12]. A fixtureless calibration strategy using a motion capture system was developed for this manipulator in [13]. Although the method was shown to be effective to calibrate a number of kinematic parameters, the experiments on the available motion capture system had much lower quality than expected. The self calibration method used here removes the need for any external sensors and allows calibration to be performed in situ as often as necessary. Unfortunately, this method does require a forward displacement solution, which has no closed-form solution for the 3-PRS [14]. This is a minor drawback, the result of which is a longer computing time during calibration.

What follows is divided into five sections. First, the kinematics that govern the 3-PRS manipulator are described; for both the inverse and forward displacement solutions. Next, a more detailed explanation of the self calibration algorithm and a brief discussion of its limitations are presented. Third is a thorough treatment of the application of the algorithm to the 3-PRS, along with a validating computer simulation. The final two sections discuss the results and provide some concluding remarks.

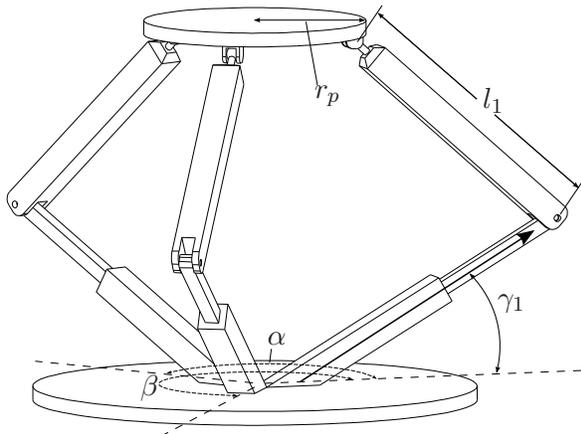


Figure 1: The 3-PRS parallel manipulator.

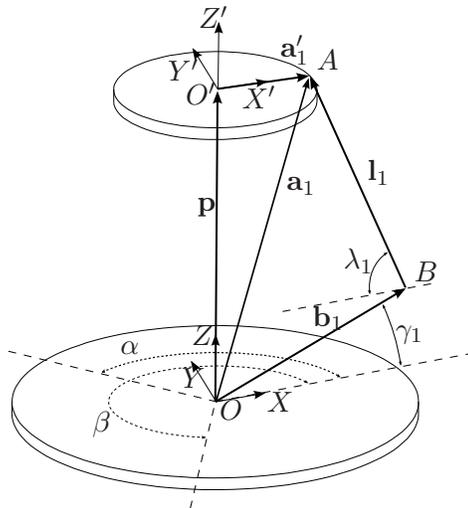


Figure 2: Vector model of the 3-PRS.

## 2 KINEMATICS OF THE 3-PRS

The 3-PRS parallel manipulator is a three degree of freedom (3-DOF) spatial manipulator. The end effector is a mobile platform that is connected to a base by three serial branches with identical kinematic architecture. Each branch, from base to end effector, consists of an active prismatic joint (P), a passive revolute joint (R), and a passive spherical joint (S). An example of the manipulator is shown in Figure 1.

Also shown in Figure 1 are the architectural parameters that affect the kinematics of the manipulator. These are the quantities being sought after by the calibration process. Angles between the prismatic joints are defined by  $\alpha$  and  $\beta$ ; the radius of the end effector platform is  $r_p$  and is defined as the radius of the circle that contains the centre of all three spherical joints. Leg length,  $l_i$ , is defined as the shortest distance from the revolute joint axis to the centre of the spherical joint, and, due to manufacturing inaccuracies, may be unique for each branch. The inclination of the prismatic joints of branch  $i$  is defined by angle  $\gamma_i$  and is measured between the line of action of the joint and the plane defined as the manipulator's base. Not shown in the figure are the prismatic joint offsets,  $d_{0i}$ ; these values represent the displacement of the prismatic joint at the encoder's zero position. Although these offsets do not affect the kinematics of the manipulator, they will be required to operate the physical mechanism. As will be seen later, it is often convenient to express the kinematic parameters as a vector:

$$\mathbf{k} = [\alpha, \beta, r_p, l_1, l_2, l_3, \gamma_1, \gamma_2, \gamma_3, d_{01}, d_{02}, d_{03}]^T \quad (1)$$

It will be the purpose of the calibration process to obtain all these quantities.

### 2.1 Inverse Displacement Solution

The kinematics of the 3-PRS were first analysed by Carretero *et al.* [12], a revised set were developed by Pond and Carretero for their work on [15] which incorporated the inclination of the

prismatic joints (*i.e.*,  $\gamma$ ). In both papers, an inertial frame is defined with the  $XY$  plane parallel to the base platform, and the  $X$  axis aligned with the projection of the prismatic joint of branch 1, onto the  $XY$  plane. It should be noted that the kinematic models in [12] and [15], as well as most other robot kinematic models, are theoretical works that do not take into account inaccuracies such as joint offset or misalignment. For instance, the equations derived in [15] assume that all the prismatic joints intersect at a common point, the origin of the base frame. Furthermore, the revolute joints are assumed to be perfectly perpendicular to the line of action of the prismatic joints, as well as parallel to the base. Under these assumptions, each spherical joint is constrained to lie on a plane perpendicular to the base plane, and parallel to the prismatic joint of the corresponding branch.

Restraining the motion of the spherical joints greatly simplifies the constraint equations that relate the dependent and independent task space variables. Because the 3-PRS is a spatial manipulator, the task space is composed of six variables (*e.g.*,  $x, y, z, \psi, \phi, \theta$ ). However, the manipulator has only 3-DOF, requiring that only three variables be chosen as independent. These parameters are all that is required to completely define the end effector pose (position and orientation). Choice of the independent variables is arbitrary, and should be chosen based on the task at hand; Carretero *et al.* [12] used elevation of the platform (displacement along the inertial  $Z$  axis), and two rotations of the end effector about the inertial frame (angles  $\psi$  and  $\theta$  about the inertial  $X$  and  $Y$  axes, respectively). Based on the constraints placed on the spherical joints' motion, it is possible to derive a set of constraint equations which yield the three dependent variables ( $x, y, \phi$ ) as a function of the independent variables ( $z, \psi, \theta$ ).

Once the end effector pose has been completely described, it is straightforward to determine the required actuator positions necessary to attain the described pose [12]. Each actuator will have two solutions but one can be eliminated by inspecting the geometry of the mechanism. Obtaining the actuator positions from the specified task space variables constitutes the complete inverse displacement solution (IDS).

## 2.2 Forward Displacement Solution

The IDS focuses on obtaining actuator positions from task variables. It follows naturally that the forward displacement solution (FDS) provides end effector pose given actuator positions. The FDS is not strictly necessary for position control of the 3-PRS manipulator. This is convenient because the FDS requires significantly more computational time to obtain. On the other hand, the FDS is required for self calibration, and in fact is the cornerstone of the calibration method used here. Unfortunately, a closed-form FDS is not possible because of the many solutions that are possible for any given actuator configuration.

The standard solution when the forward kinematics can not be solved analytically is to apply an iterative approach using the IDS [16]. This solution starts with an approximate end effector pose and computes the required joint coordinates. The solution is improved at each iteration by taking a step defined by the inverse of the Jacobian matrix<sup>4</sup>. This, of course, requires the Jacobian matrix to be invertible which is impossible at singular configurations.

---

<sup>4</sup>In this case, it is assumed that the Jacobian is square which is the case for the 3-PRS whose Jacobian is a  $3 \times 3$  matrix.

As will be discussed in detail in Sections 3 and 4, the calibration methods proposed in [11] requires the manipulator to transit singularities. Therefore, methods such as the one presented in [16] can not be used to obtain the FDS. Fortunately, a fast and efficient solution of the forward displacement problem for the 3-PRS, which does not require the Jacobian matrix to be inverted, was presented by Tsai *et al.* in [14].

Given the displacement of the prismatic joints, the position of the revolute joints  $B_i$  will be known; and can be written as follows:

$$\mathbf{b}_1 = \begin{bmatrix} d_1 c_{\gamma_1} \\ 0 \\ d_1 s_{\gamma_1} \end{bmatrix}; \quad \mathbf{b}_2 = \begin{bmatrix} d_2 c_{\alpha} c_{\gamma_2} \\ d_2 s_{\alpha} c_{\gamma_2} \\ d_2 s_{\gamma_2} \end{bmatrix}; \quad \mathbf{b}_3 = \begin{bmatrix} d_3 c_{\beta} c_{\gamma_3} \\ d_3 s_{\beta} c_{\gamma_3} \\ d_3 s_{\gamma_3} \end{bmatrix} \quad (2)$$

where  $d_i$  is the distance from the origin,  $O$  to the revolute joint (see Figure 2), including the offset  $d_{0i}$ ; and  $c_* = \cos(*)$  and  $s_* = \sin(*)$ .

If the direction  $s_{l_i}$  of the leg links  $\mathbf{l}_i$  can be found, then, the position of the corresponding spherical joint  $A_i$  can also be found. Further, the pose of the end effector is simple to obtain once the position of the three spherical joints is known. Therefore, the FDS can be reduced to determining the angle between leg link and the  $XY$  plane, defined as angle  $\lambda_i$ .

The unit vectors of the leg links can be expressed in terms of  $\lambda_i$  as:

$$\mathbf{l}_1 = l_1 \begin{bmatrix} -c_{\lambda_1} \\ 0 \\ s_{\lambda_1} \end{bmatrix}; \quad \mathbf{l}_2 = l_2 \begin{bmatrix} -c_{\lambda_2} c_{\alpha} \\ -c_{\lambda_2} s_{\alpha} \\ s_{\lambda_2} \end{bmatrix}; \quad \mathbf{l}_3 = l_3 \begin{bmatrix} -c_{\lambda_3} c_{\beta} \\ -c_{\lambda_3} s_{\beta} \\ s_{\lambda_3} \end{bmatrix} \quad (3)$$

where  $\mathbf{l}_i = l_i \mathbf{s}_{l_i}$ .

Adding equation (3) to equation (2) defines the positions of the spherical joints with respect to the inertial frame  $\{O\}$ . That is,  $\mathbf{a}_i = \mathbf{b}_i + l_i \mathbf{s}_{l_i}$ . The spherical joints' positions can also be described with respect to the end effector frame  $\{O'\}$ , in which case their position remains constant, and is:

$$\mathbf{a}'_1 = \begin{bmatrix} r_p \\ 0 \\ 0 \end{bmatrix}; \quad \mathbf{a}'_2 = \begin{bmatrix} r_p c_{\alpha} \\ r_p s_{\alpha} \\ 0 \end{bmatrix}; \quad \mathbf{a}'_3 = \begin{bmatrix} r_p c_{\beta} \\ r_p s_{\beta} \\ 0 \end{bmatrix} \quad (4)$$

As demonstrated by Tsai *et al.* [14] the distance between any two spherical joints is constant, regardless of which frame of reference they are seen from. Therefore, the distance between spherical joints 1 and 2 can be written with respect to frame  $\{O\}$  and frame  $\{O'\}$ , and the expressions equated. The procedure is repeated for each joint combination and yields the following three equations:

$$|\mathbf{a}_1 - \mathbf{a}_2|^2 = |\mathbf{a}'_1 - \mathbf{a}'_2|^2 \quad (5)$$

$$|\mathbf{a}_2 - \mathbf{a}_3|^2 = |\mathbf{a}'_2 - \mathbf{a}'_3|^2 \quad (6)$$

$$|\mathbf{a}_3 - \mathbf{a}_1|^2 = |\mathbf{a}'_3 - \mathbf{a}'_1|^2 \quad (7)$$

Equations (5) to (7) are a system of 3 simultaneous equations in 3 unknowns. In [14], Tsai *et al.* showed that these equations could be solved using Bezout's elimination method; it was also

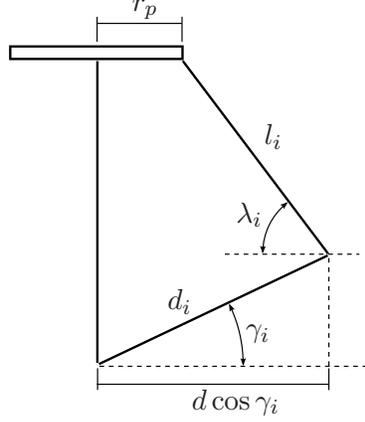


Figure 3: Approximation of  $\lambda_i$  by assuming end effector horizontal.

demonstrated that it yields 64 solutions. Elimination of infeasible, or mechanically impossible solutions is time consuming. Tsai *et al.* suggested an optimization technique to minimize an objective function subject to boundary and behaviour constraints to compel the desired solution.

The objective function to be minimized is formed by summing equations (5) to (7). Gathering all terms to one side yields:

$$\begin{aligned}
& (d_1 c_{\gamma_1} - l_1 c_{\lambda_1} - d_2 c_{\alpha} c_{\gamma_2} + l_2 c_{\lambda_2} c_{\alpha})^2 + (-d_2 s_{\alpha} c_{\gamma_2} + l_2 c_{\lambda_2} s_{\alpha})^2 \\
& + (d_1 s_{\gamma_1} + l_1 s_{\lambda_1} - d_2 s_{\gamma_2} - l_2 s_{\lambda_2})^2 - (r_p - r_p c_{\alpha})^2 - (r_p s_{\alpha})^2 \\
& + (d_2 c_{\alpha} c_{\gamma_2} - l_2 c_{\lambda_2} c_{\alpha} - d_3 c_{\beta} c_{\gamma_3} + l_3 c_{\lambda_3} c_{\beta})^2 + (d_2 s_{\alpha} c_{\gamma_2} - l_2 c_{\lambda_2} s_{\alpha} - d_3 s_{\beta} c_{\gamma_3} \\
& + l_3 c_{\lambda_3} s_{\beta})^2 + (d_2 s_{\gamma_2} + l_2 s_{\lambda_2} - d_3 s_{\gamma_3} - l_3 s_{\lambda_3})^2 - (r_p c_{\alpha} - r_p c_{\beta})^2 - (r_p s_{\alpha} - r_p s_{\beta})^2 \\
& + (d_3 c_{\beta} c_{\gamma_3} - l_3 c_{\lambda_3} c_{\beta} - d_1 c_{\gamma_1} + l_1 c_{\lambda_1})^2 + (d_3 s_{\beta} c_{\gamma_3} - l_3 c_{\lambda_3} s_{\beta})^2 \\
& + (d_3 s_{\gamma_3} + l_3 s_{\lambda_3} - d_1 s_{\gamma_1} - l_1 s_{\lambda_1})^2 - (r_p c_{\beta} - r_p)^2 - (r_p s_{\beta})^2 = 0
\end{aligned} \tag{8}$$

Although the objective function has many terms, it is not overly complicated, and an algebraic expression for the gradient as well as the Hessian can be derived. Analytical solutions for both of these gradient matrices greatly increases the computational efficiency [17] of the optimization algorithms used to determine the values of  $\lambda_i$  for  $i = 1, 2, 3$ .

Unlike Tsai *et al.*, behaviour constraints are not used. They were considered, but implementing a constrained optimization algorithm would have required significantly more computational time. Instead, it was found that by using a good approximation of the solution as a starting point, the basic Newton Search method performed very well.

It is possible to approximate  $\lambda_i$  by examining the geometry of the manipulator (see Figure 3). All dimensions shown are known except the angle  $\lambda_i$  and the inclination of the end effector platform. If the end effector is assumed to be exactly horizontal, then  $\lambda_i$  can be approximated by:

$$\lambda_i = \arccos \left( \frac{d_i c_{\gamma_i} - r_p}{l_i} \right) \tag{9}$$

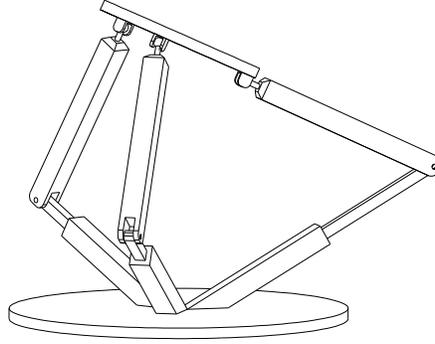


Figure 4: Example of a direct kinematic singularity of the 3-PRS.

Using this approximation as a starting point, the Newton Search method converged very quickly (almost always in less than 20 iterations). Once each angle  $\lambda_i$  is known, the position of the spherical joints can be found using equations (2) and (3). The end effector pose can be readily derived from the position of the spherical joints. The results of this procedure were checked by using the IDS to verify that the end effector pose returned corresponded to the original actuator positions. The results were found to be accurate to within  $1 \times 10^{-9}$  units.

### 3 SELF CALIBRATION WITHOUT ADDITIONAL SENSORS

As mentioned in the introduction, the self calibration strategy used here was developed by Last *et al.* [11], and allows calibration without the use of any external or additional sensors. This is accomplished by specific knowledge about the manipulator in question when it enters a direct kinematic singularity. When this occurs, all actuators can remain locked and the end effector is still capable of small movements. The positions of the actuators for a given singular configuration are very precise. By comparing their displacement as predicted by the kinematic model to measured values from the physical model, redundant information is obtained.

Because the self calibration strategy uses singular configurations to obtain the redundancy required for calibration, this strategy can not be applied to all manipulators. That is, not all parallel manipulators have direct kinematic singularities within the reachable workspace, and serial manipulators do not exhibit this type of singularity at all. Fortunately the 3-PRS has three distinct singularities of this type within its workspace (one for each branch). Specifically, the type of singularity that is being exploited here occurs when the line in the direction  $s_{l_i}$  and passing through  $B_i$  intersects the line passing through the centre of the spherical joints  $A_i$  on the other two branches (see Figure 4).

To pass through the singularity, two of the actuators are locked in the same position while the third one is released. Application of an external force (gravity if oriented as depicted in Figures 1 and 4) causes the end effector to pivot downwards. As this happens the released joint will extend until the singularity is achieved, at which time, it will start to retract. By monitoring the actuator position for this joint, the time at which the velocity is zero corresponds exactly to the exact moment that the manipulator passes the singular pose.

A reading of the position of the extended prismatic joint,  $q_{measured}$ , is taken from the singular configuration. Using the positions of the fixed actuators as input the singular configuration is imitated using the kinematic model. From this theoretical singular pose, a corresponding reading,  $q_{theoretical}$ , is taken, where  $\mathbf{k}$  represents a vector of kinematic parameters for the model used to generate the singular pose. The theoretical reading is compared with the measured value to form a residual,

$$r(\mathbf{k}) = q_{measured} - q_{theoretical}(\mathbf{k}) \quad (10)$$

The process is repeated at various singular configurations. Each residual is added to a vector and an optimization algorithm is used to find the kinematic parameters  $\mathbf{k}$  that result in the sum of least squares.

## 4 IMPLEMENTATION

Self calibration without redundant sensors is an exercise in optimization several layers deep, specially as applied to the 3-PRS manipulator. At each iteration, a new set of kinematic parameters is tested, which requires that the theoretical singular configurations be recalculated, to obtain a new residual vector. Every singular pose that must be identified requires a bisection search; every iteration of which requires a solution of the forward displacement problem. This process is outlined in Figure 5, and each stage of the process is described in the sections that follow.

### 4.1 Optimization of Kinematic Parameters

The primary goal of the self calibration algorithm is to determine a set of kinematic parameters that best represents the physical mechanism being calibrated. Section 3 described a method by which to measure how closely the values of  $\mathbf{k}$  mimic that mechanism, allowing a numerical search to be performed.

The Levenberg-Marquardt (LM) algorithm is a local search technique for obtaining the sum of least squares of non-linear functions [18]. This is exactly the type of problem presented here. Lourakis [18] wrote a self-contained implementation of the LM algorithm in C which he titled `levmar`; the code for this package is freely available online.

To use the `levmar` function, requires a measurement vector, which corresponds to the prismatic joint positions for the measured singular configurations. At each iteration of the algorithm, the function tests a new  $\mathbf{k}$ , which requires an estimated measurement vector. This estimation is a recalculation of the theoretical prismatic joint positions for the predicted singular configurations, and represents the objective function that the algorithm uses to guide its search. Algorithm termination occurs when either: the gradient of the sum of least squares, the relative change in step size or the sum of least squares drop below their respective thresholds; or, if a maximum number of iterations has been reached.

### 4.2 Calculation of Actuator Positions

The residual vector, as defined in Section 3, is calculated by the LM algorithm internally. All that is required for this calculation is an update of  $q_{theoretical}$  values for each singular configuration that was tested, *i.e.*, at least one new  $q_{theoretical}$  for each parameter being calibrated. The current

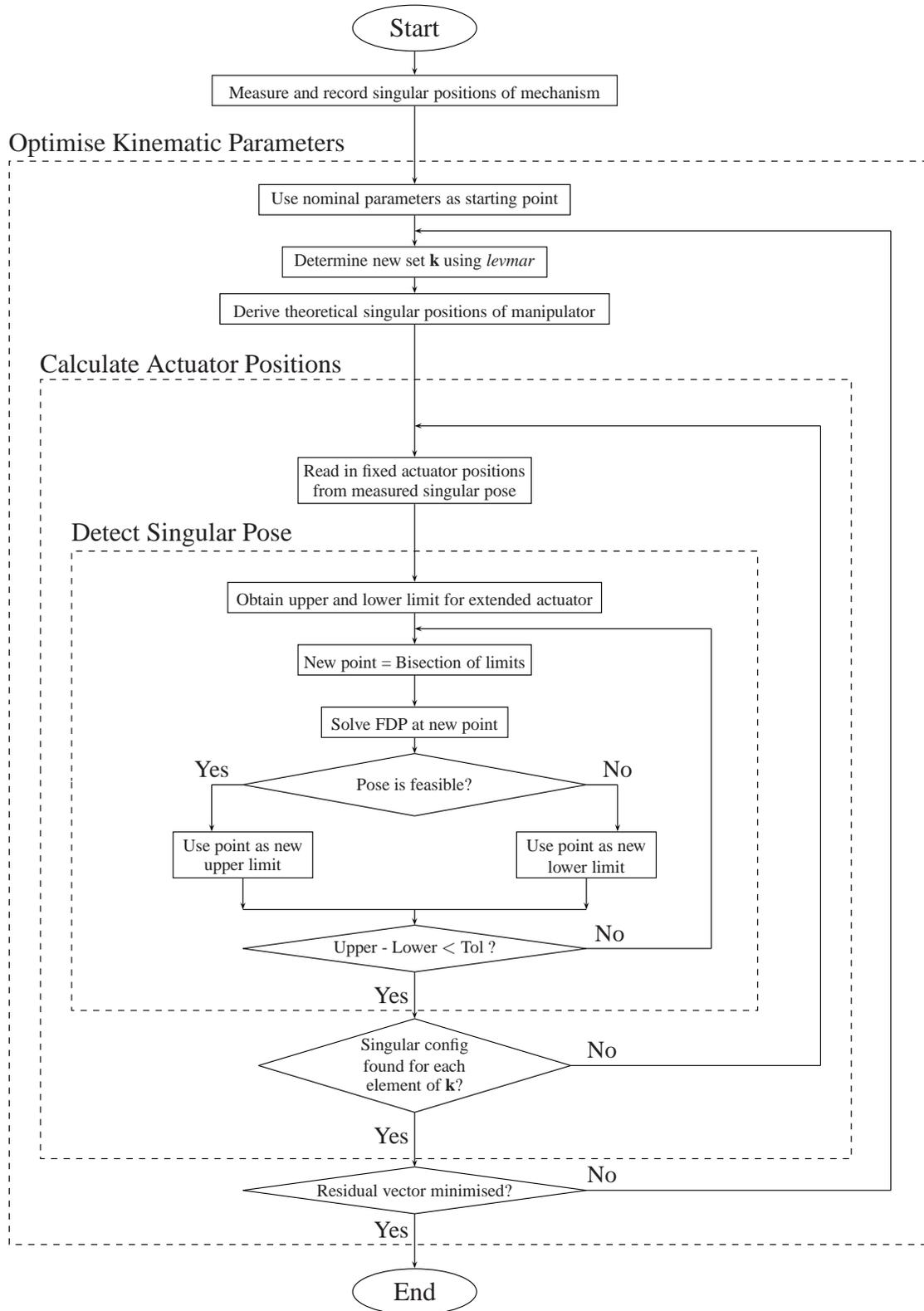


Figure 5: Flow chart of the self calibration algorithm.

model has 12 distinct parameters being calibrated, with 12 measured singular configurations, each requiring a new  $q_{theoretical}$ .

It may seem trivial that these calculations need to be performed for each measurement. Consider however, that the model may be extended to include as many as 22 parameters, the `levmar` function only requires approximately 30 iterations, and the forward kinematics can require up to 20 iterations. Clearly, repetition for each parameter adds a significant level of iteration in the nested structure of the algorithm.

### 4.3 Detection of Singular Configurations

Each  $q_{theoretical}$  that must be calculated requires a new theoretical singular configuration. Identifying a singular configuration is not trivial and indeed requires several iterations of the FDS. However, there are limits to possible values of  $q_{theoretical}$  as it is impossible for the prismatic joint to extend beyond the reach of the leg link. Thus, the maximum value for  $q_{theoretical}$  is:

$$q_{theoretical}^{max} = \frac{r_p + l_i}{\cos(\gamma_i)} \quad (11)$$

where  $i$  is the branch index. The lower limit,  $q_{theoretical}^{min}$ , is taken as the position of the two fixed prismatic joints, the average is used if they are different. Because only one prismatic joint is driven while the other two are held fixed in the same position, the singularity can only occur once the driven joint is extend beyond the position of the fixed ones.

With clear limits for  $q_{theoretical}$ , a bisection search is performed to obtain its precise value. Recall equation (8) from the forward kinematics, the FDS seeks to minimize the left hand side of this equation. If the solution is not feasible, then the result will be greater than zero. Such is the case when the driven prismatic joint is extended beyond the point of the singular configuration. If the prismatic joint is within the singular point then the left hand side of equation (8) will equal zero (or as near as machine precision will allow). The bisection search continues until the upper and lower limit are within a set tolerance of one another. Once this condition is met, the average value between upper and lower limits is used as  $q_{theoretical}$ . As the search begins to converge, the solution of the previous iteration makes a better starting point for the FDS than the approximation made in equation 9. To gauge when this occurs, a second (much larger) tolerance level is set, and once the distance between upper and lower limits is within this second tolerance the previous solution to the FDS is used as the starting point for the next step. The result is a significant decrease in the number of Newton Search steps required to solve the FDS.

## 5 RESULTS

A computer simulation was run to verify that both the method and the implementation worked correctly. A list of feasible singular configurations was generated using the method described in Section 4.3, using non-nominal kinematic parameters, and recorded in joint space. From this list, a vector of  $q_{measured}$  values was formed, and used as the measured vector for the LM algorithm. The program was then run with nominal values for the starting parameters  $k$ .

The program ran successfully and only required approximately 30 iterations of the `levmar` algorithm to complete. The results always matched the kinematic parameters used to generate

the test data exactly. Although this is only a simulation, and no consideration has been made for any type of noise, the results are extremely encouraging. Successful extrapolation of the model parameters validates the work that has been completed thus far.

## 6 CONCLUSIONS

A calibration strategy that requires no redundant sensors was implemented to calibrate the 3-PRS parallel manipulator. The mechanism was described, and a brief explanation of the inverse kinematics provided. A derivation of the forward displacement solution was presented which constitutes the crux of the calibration strategy. The self calibration strategy was explained and a description of implementation procedure given. Because all measurements are taken in joint space, the FDS is vital to obtain corresponding theoretical singular configurations.

A simulation of the calibration procedure was conducted to validate the method and test the realisation. Kinematic parameters can be obtained that exactly match those used to generate the test data. Although the work is only preliminary, it was successful and is an excellent first step for an accurate yet simple calibration procedure that can be applied to the 3-PRS and integrated into the controller. It is interesting to observe, that the IDS was not required at any stage during the calibration process. The whole process is completed using only the FDS.

Remaining work includes development of an extended kinematic model of the 3-PRS to better model the geometrical discrepancies resulting from manufacturing. Also, physical experiments need to be performed to measure the effectiveness of the calibration strategy. To this end, a prototype of the 3-PRS is currently being constructed at the University of New Brunswick.

## ACKNOWLEDGEMENTS

The first two authors acknowledge the financial support from the National Science and Engineering Research Council of Canada (NSERC) through an NSERC Discovery grant.

## REFERENCES

- [1] J. Ziegert and P. Datsoris, "Basic considerations for robot calibration," 1988, pp. 932-938.
- [2] M. R. Driels, "Generalized joint model for robot manipulator kinematic calibration," *Journal of Robotic Systems*, vol. 4, no. 1, pp. 77-114, 1987.
- [3] L. Notash and R. P. Podhorodeski, "Kinematic calibration of parallel manipulators," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, 1995, pp. 3310-3315.
- [4] P. Last, J. Hesselbach, and N. Plitea, "An extended inverse kinematic model of the hexa-parallel-robot for calibration purposes," in *IEEE International Conference on Mechatronics and Automation, ICMA 2005*, 2005.
- [5] P. Vischer and R. Clavel, "Kinematic calibration of the parallel delta robot," *Robotica*, vol. 16, no. 2, pp. 207-218, 1998.
- [6] B. W. Mooring, Z. S. Roth, and M. R. Driels, *Fundamentals of Manipulator Calibration*. John Wiley and Sons Inc., 1991.

- [7] Y. Koseki, T. Arai, K. Sugimoto, T. Takatuji, and M. Goto, "Design and accuracy evaluation of high-speed and high precision parallel mechanism," in *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, May 1998, pp. 1340–1345.
- [8] F. Liangzhi, A. Elatta, and L. Xiaoping, "Kinematic calibration for a hybrid 5DOF manipulator based on 3-RPS in-actuated parallel manipulator," *International Journal of Advanced Manufacturing Technology*, vol. 25, no. 7-8, pp. 730–734, April 2005.
- [9] H. Zhuang, "Self-calibration of parallel mechanisms with a case study on Stewart platforms," in *IEEE Transactions on Robotics and Automation*, vol. 13, no. 3, June 1997, pp. 387–397.
- [10] P. Renaud, A. Vivas, N. Andreff, P. Pognet, P. Martinet, F. Pierrot, and O. Company, "Kinematic and dynamic identification of parallel mechanisms," *Control Engineering Practice*, vol. 14, no. 9, pp. 1099–1109, September 2006.
- [11] P. Last, C. Budde, M. Krefft, and J. Hesselbach, "Parallel robot self-calibration without additional sensors or constraint devices," in *Proceedings of the Joint Conference on Robotics: ISR*, Munich, Germany, May 2006.
- [12] J. A. Carretero, M. Nahon, and R. Podhorodeski, "Workspace analysis and optimization of a novel 3-dof parallel manipulator," *International Journal of Robotics and Automation*, vol. 15, no. 4, pp. 178–188, 2000.
- [13] C. Van Driel and J. Carretero, "Calibration of the 3-PRS parallel manipulator using a motion capture system," *Transactions of the Canadian Society for Mechanical Engineering*, vol. 29, no. 4, pp. 645–654, 2005.
- [14] M.-S. Tsai, T.-N. Shiau, Y.-J. Tsai, and T.-H. Chang, "Direct kinematic analysis of a 3-PRS parallel mechanism," *Mechanism and Machine Theory*, vol. 38, no. 1, pp. 71–83, 2003.
- [15] G. Pond and J. A. Carretero, "Kinematic analysis and workspace determination of the inclined PRS parallel manipulator," in *Proceedings of RoManSy 2004: 15th CISM-IFTOMM Symposium on Robot Design*, Montreal, Canada, June 2004.
- [16] O. Masory, W. J., and H. Zhuang, "Kinematic modeling and calibration of a Stewart platform," *Journal of Advanced Robotics*, vol. 11, no. 5, pp. 519–539, 1997.
- [17] S. S. Rao, *Engineering Optimization: Theory and Practice*, 3rd ed, New Age International Ltd., 2002.
- [18] M. I. A. Lourakis, "A brief description of the Levenberg-Marquardt algorithm implemented by levmar," <http://www.ics.forth.gr/lourakis/levmar/>, February 11 2005.