Combining CP and Interval Methods for solving the Direct Kinematic of a Parallel Robot under Uncertainties

Carlos Grandon, David Daney, and Yves Papegay

COPRIN – INRIA Sophia Antipolis

1 Introduction

Computing the direct kinematic model of a robot is a central concern in robotics, as it is a key step for evaluating workspaces, performances and for control.

In the case of the 3-2-1 robot - a simplified version of a general 6-Degrees of freedom parallel manipulator - a formal expression of this model is available involving several geometrical parameters [4, 6].

Uncertainties, which appear in these parameters are namely due to measurements and manufacturing, and have to be taken into account when certified results are required.

Due to the wrapping effect, and to the interval natural extension, using interval arithmetic for evaluating the expressions of the direct kinematic model does not produce good results.

In this presentation, we will show that the use of 2B-consistency and 3B-consistency methods greatly enhances the quality of the results and how several original strategies lead to very good performances.

2 Modelling

The parallel manipulator we are considering in this study is shown on figure 1.

It consists of two solids (the basis and the mobile platform) linked through six linear rods of variable lengths (the legs). These rods are attached to each platform through ball joints, considered as punctual.

The direct kinematic model expresses how the lengths of the legs are related with the position and the orientation of the mobile platform, through 6 distance equations:

$$\|\boldsymbol{A_i B_j}\|^2 = d_{i,j}^2, \quad (i,j) \in \{(1,1),(2,1),(3,1),(4,2),(5,2),(6,3)\} \tag{1}$$

When studying a 3-2-1 robot, to take into account its specificity, it is classical to represent the position and the orientation of the mobile platform through the 9 unknown coordinates of the 3 attachment points B_1, B_2 and B_3. The direct kinematic model is therefore completed by 3 distance equations:

$$\|\boldsymbol{B_i B_j}\|^2 = \delta_{i,j}^2, \quad (i,j) \in \{(1,2),(1,3),(2,3)\} \tag{2}$$

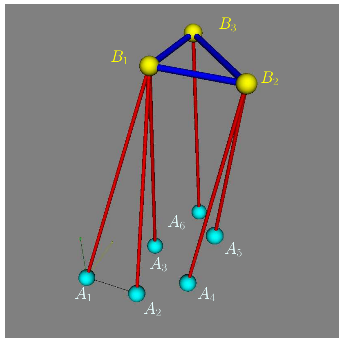


Fig. 1. A typical 3-2-1 parallel robot.

3 Solving

For solving the direct kinematic model, we assume our reference system to be based on A_1, A_2, A_3 - i.e. that $x_{A_1} = y_{A_1} = z_{A_1} = y_{A_2} = z_{A_2} = z_{A_3} = 0$.

3.1 Formal Solution

A closed-form solution is computed in three steps:

1. the 3 first equations of (1) become

$$\left\{ \begin{array}{r} x_{B_1}^2 + y_{B_1}^2 + z_{B_1}^2 = d_{1,1}^2 \\ (x_{B_1} - x_{A_2})^2 + y_{B_1}^2 + z_{B_1}^2 = d_{2,1}^2 \\ (x_{B_1} - x_{A_3})^2 + (y_{B_1} - y_{A_3})^2 + z_{B_1}^2 = d_{3,1}^2 \end{array} \right.$$

   and allows us to easily formally express x_{B_1}, y_{B_1}, and two opposite solutions for z_{B_1},
2. the equations

$$\left\{ \begin{array}{l} (x_{B_2} - x_{A_4})^2 + (y_{B_2} - y_{A_4})^2 + (z_{B_2} - z_{A_4})^2 = d_{4,2}^2 \\ (x_{B_2} - x_{A_5})^2 + (y_{B_2} - y_{A_5})^2 + (z_{B_2} - z_{A_5})^2 = d_{5,2}^2 \\ (x_{B_2} - x_{B_1})^2 + (y_{B_2} - y_{B_1})^2 + (z_{B_2} - z_{B_1})^2 = \delta_{1,2}^2 \end{array} \right.$$

   allow us, as classical equations for intersection of three spheres, to formally express x_{B_2}, y_{B_2}, and z_{B_2}
3. and similarly,

$$\left\{ \begin{array}{l} (x_{B_3} - x_{A_6})^2 + (y_{B_3} - y_{A_6})^2 + (z_{B_3} - z_{A_6})^2 = d_{6,3}^2 \\ (x_{B_3} - x_{B_1})^2 + (y_{B_3} - y_{B_1})^2 + (z_{B_3} - z_{B_1})^2 = \delta_{1,3}^2 \\ (x_{B_3} - x_{B_2})^2 + (y_{B_3} - y_{B_2})^2 + (z_{B_3} - z_{B_2})^2 = \delta_{2,3}^2 \end{array} \right.$$

   gives, the expressions of x_{B_3}, y_{B_3}, and z_{B_3}

Theoretically, successive substitutions of the coordinates of B_1 and of B_2 return 8 different solutions for the 9 expressions of the coordinates of the B_j in terms of the parameters (coordinates of the A_i and distances). However, due to the huge size of the obtained expressions, performing these substitutions is untractable.

Numerical solutions of the direct kinematic model are computed by evaluating at each of the above steps the symbolic expressions after substitution of the numerical values of the parameters, and of the results of the previous steps.. Numerical methods (namely based on determinants [5]) are also available which directly produce the solutions but are much more difficult to extend to intervals.

4 Interval Evaluation

Due to measurement uncertainties and to the unaccuracy of manufacturing and assembly, we only have approximate values and bounded errors for the parameters. In this context, we use an interval representation, and we are interested in getting certified enclosing 3D-boxes approximating the positions of the points B_1, B_2 and B_3 for each of the solutions.

The classical way to compute the corresponding intervals is to perform the evaluation of the symbolic expressions using the interval values of the parameters and interval arithmetics. For avoiding numerical errors when evaluating huge expressions, this numerical evaluation is done at each of the three steps described above.

Actually, each of the system of three spherical equations are solved in the same way, vanishing square terms by substraction of equations to linearly get expressions of two variables, and then solving a polynomial of degree 2, and the expressions are manipulated to minimize the number of occurences of the variables.

However, results are strongly overestimated, in term of width of the obtained intervals as illustrated by the numerical example of table 1. This is a well known drawback of interval methods of accumulating and propagating uncertainties and of not properly considering multiple occurrences of the same variables when evaluating an expression.

4.1 Methods of Constraint Programming

To overcome this overestimation problem, several classical methods of Constraints Programming [3, 1] have been implemented and tested:

- 2B-consistency which is based on arc-consistency for filtering the domain of the variables, using a projection function for computing the pre-image of the function over each variable appearing in the equation,
- 3B-consistency, a stronger consistency, checking whether 2B-consistency can be enforced when the domain of a variable is reduced to the value of one of its bounds in the whole system,
- splitting (or bisection) which divides a problem into two sub-problems through the splitting of the interval domain of a variable into two or more intervals.

$$
\begin{cases} x_{A_1} = 0 \\ x_{A_2} = [1000, 1001] \\ x_{A_3} = [799, 800] \\ x_{A_4} = [1800, 1801] \\ x_{A_5} = [2099, 2100] \\ x_{A_6} = [1300, 1301] \end{cases}
\quad
\begin{cases} y_{A_1} = 0 \\ y_{A_2} = 0 \\ y_{A_3} = [1199, 1200] \\ y_{A_4} = [400, 401] \\ y_{A_5} = [900, 901] \\ y_{A_6} = [2199, 2200]\ [200, 201] \end{cases}
\quad
\begin{cases} z_{A_1} = 0 \\ z_{A_2} = 0 \\ z_{A_3} = 0 \\ z_{A_4} = [199, 200] \\ z_{A_5} = [99, 100] \\ z_{A_6} = [200, 201] \end{cases}
$$

$$
\begin{cases} d_{1,1} = [1100, 1110] \\ d_{2,1} = [900, 910] \\ d_{3,1} = [1203, 1213] \\ d_{4,2} = [855, 865] \\ d_{5,2} = [801, 811] \\ d_{6,3} = [872, 882] \end{cases}
\quad
\begin{cases} \delta_{1,2} = [1489, 1490] \\ \delta_{1,3} = [1256, 1257] \\ \delta_{2,3} = [1799, 1800] \end{cases}
$$

Table 1. Numerical example (lengths in mm)

Splitting is mainly used for separation of the solutions: after a filtering phase, a bisection of the resultant box in the middle point of a variable is performed, followed by a new filtering step in each generated sub-domain. This process is repeated for each of the variables, and either two separated boxes or the convex hull of the connected boxes are returned – see [2] for details.

4.2 Implemented Algorithms

Experiments have been carried out with four different algorithms and several randomly generated set of numerical data.

1. The basic one, tested for comparison purpose, consists in an interval evaluation of symbolic expressions, at each of the three steps.
2. Second algorithm applies a 2B-consistency to the solutions after each step and a 2B-consistency to the whole systems after the third step.
3. Third algorithm applies the same strategy as the previous one but using 3B-consistency instead of 2B-consistency.
4. The last one is an enhancement of the previous, computing a closer approximation of the solutions using a branch and prune algorithm that combines consistency filtering and splitting phases.

The table 2 below shows the results (1 selected solution) obtained on the numerical example described in table 1. The precision vector is the vector of the half-widths of the intervals computed for the respective coordinates of B_1, B_2 and B_3, volumes are the respective volumes of the correspondant boxes.

Algorithms 1, 2, and 3 compute a solution in the form $\boldsymbol{B} \pm \boldsymbol{\epsilon}$, where $\boldsymbol{B}$ represents the coordinates of the points B_1, B_2 and B_3, and $\boldsymbol{\epsilon}$ is the precision vector.

Last algorithm computes a representation of the solution space through a set of disjoint 3D-boxes with a given precision of 10 mm.

Algo.	Precision vector	Volumes	Time
1	(10.4, 17.2, 22.5, 125, 86, 155, 1350, 1350, 1400)	$(32244, 1.3283 \times 10^7, 2.0331 \times 10^{10})$	0.001
2	(10.4, 17.2, 16.2, 45.6, 73.2, 15.5, 637, 395, 539)	$(23222, 415187, 1.0835 \times 10^9)$	0.002
3	(10.3, 9.57, 6.74, 23.5, 32.8, 7.22, 56.6, 24.3, 25.4)	$(5321, 44438, 278634)$	0.127
4	(10.3, 9.57, 6.74, 23.5, 32.8, 7.22, 56.6, 23.1, 25.4)	–	4.041

Table 2. Numerical results

Figure 2 shows the results computed for B_1, B_2 and B_3 with the first three algorithms. The figure 3 is a zoom on each of the point showing also the boxes solutions computed with the last algorithm.

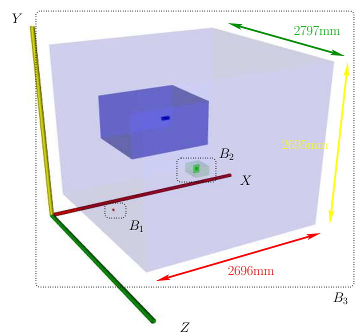
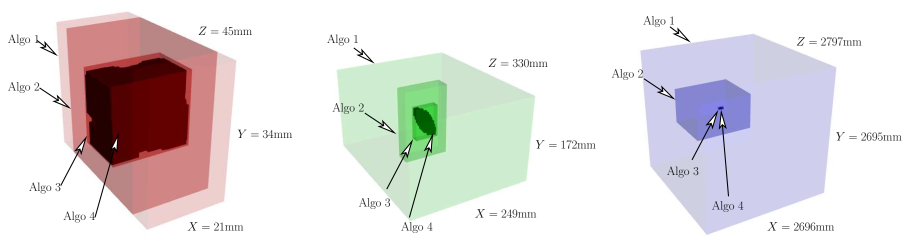


Fig. 2. The three points B_1, B_2, B_3



(a) on B_1 (b) on B_1 (c) on B_1

Fig. 3. Zoom on points $B1, B_2, B_3$

5 Conclusion

The paper discusses interval extension of the direct kinematic of a 3-2-1 parallel robot.

Motivation of this study is the development of a 3D measurement device based on this architecture where legs of the robot are replaced by cables.

This application emphasizes the problem of interval evaluation of a model and shows that specific filtering algorithms are interesting to provide realistic sharp results. It is important to note that the splitting is only useful for separating solutions but does not enhance the sharpness of the bounding.

References

1. F. Benhamou, F. Goualard, and L. Granvilliers. Revising hull and box consistency. In *International Conference on Logic Programming*, pages 230–244, 1999.
2. C. Grandon, N. Morales, and G. Chabert. A strategy for solving systems of equations with uncertainties. In *Accepted for publication in 12th GAMM - IMACS International Symposion on Scientific Computing, Computer Arithmetic and Validated Numerics*, 2006.
3. O. Lhomme. Consistency techniques for numeric csps. In *IJCAI*, pages 232–238, 1993.
4. D.E. Manolakis. Efficient solution and performance analysis of 3-d position estimation by trilateration. *IEEE Transactions on Aerospace and Electronic Systems*, 32:1239–1248, 1996.
5. F. Thomas, E. Ottaviano, L. Ros, and M. Ceccarelli. Uncertainty model and singularities of 3-2-1 wire-based tracking systems. In *ARK*, pages 107–117, 2002.
6. F. Thomas, E. Ottaviano, L. Ros, and M. Ceccarelli. Performance analysis of a 3-2-1 pose estimation device. *IEEE Transactions on Robotics*, 21(3), 2005.