Interval Analysis and Robotics

J.-P. Merlet

INRIA Sophia-Antipolis, France
Jean-Pierre.Merlet@sophia.inria.fr

Abstract. Interval analysis is a relatively new mathematical tool that allows one to deal with problems that may have to be solved numerically with a computer. Examples of such problems are system solving and global optimization but numerous other problems may be addressed as well. This approach has the following general advantages:

- it allows to find solutions of a problem only within some finite domain
- numerical computer round-off errors are taken into account so that the solutions are guaranteed

A further inherent property that is of interest for robotics problems is that this approach allows one to deal with the *uncertainties* that are unavoidable in robotics. Although the basic principles of interval analysis are easy to understand and to implement, this approach will be efficient only if additional methods are used and if the problem at hand is formulated appropriately. In this paper we will emphasize various robotics problems that have been solved with interval analysis, many of which are currently beyond the reach of other mathematical approaches.

1 Introduction

Interval analysis [6, 8, 12] is a powerful numerical method that allows one to solve a broad range of problems (going from system solving to global optimization). It has been early used for solving the inverse kinematics problem for serial 6R robot [14] but is now used for solving many robotic problems: clearance effect on robot [18], robot reliability [2], localization and navigation [1, 4, 17], motion planning [13], collision detection [15], calibration [5] to name a few. From the numerical view point the main advantages of this approach are flexibility (many different problems may be considered) and guaranteed results (no solution may be lost) as numerical round-off errors are taken into account. We will also see that this method allows one to deal with uncertainties in the robot modeling and control, that cannot be avoided in practice. The drawback of the method are its worst case exponential complexity and the necessary high level of expertise to implement efficient algorithms. A key element of interval analysis is *interval arithmetic* that will be presented in the next section.

M. Kaneko and Y. Nakamura (Eds.): Robotics Research, STAR 66, pp. 147–156.
springerlink.com

1.1 Interval Arithmetic

The purpose of interval arithmetic is to determine guaranteed bounds for the minimum and maximum of a given function f over ranges for the unknowns. This determination is called an *interval evaluation* of the function and leads to a range $[\underline{F}, \overline{F}]$ that varies according to the ranges for the unknowns. If $\mathbf{X}$ denotes the ranges for the unknowns and $\mathbf{X_0}$ is a particular instance of the values of the unknowns within $\mathbf{X}$, then we have:

$$\underline{F} \leq f(\mathbf{X_0}) \leq \overline{F} \tag{1}$$

An interval evaluation may be calculated in different ways. The simplest is called the *natural evaluation*, which consist in using specific interval versions of all mathematical operators used in the function (interval version exists for all classical operators). For example the addition of two intervals $\mathbf{a} = [\underline{a}, \overline{a}]$, $\mathbf{b} = [\underline{b}, \overline{b}]$ is defined as $\mathbf{a}+\mathbf{b} = [\underline{a}+\underline{b}, \overline{a}+\overline{b}]$. Natural evaluation may be simply illustrated with $f = x^2 - 2x$ when x lie in the range [3,5]. In that case we can safely state that for any instance of x in [3,5], then x^2 lie in [9,25], $2x$ in [6,10] and consequently $-2x$ in [-10,-6]. Summing the interval for x^2 and $-2x$ leads to [9,25]+[-10,-6]= [-1,19] which constitutes the interval evaluation of f over the range [3,5]. Interval evaluation may also be obtained using other methods (Taylor expansion, centered form) and is sensitive to the analytical form used for the function. For example f may also be written as $f = (x-1)^2 - 1$ whose evaluation is [3,15].

This example shows that interval arithmetic requires simple operations but also illustrates one of the drawbacks of the method. Indeed clearly for any x in [3,5] the value of f lie in [3,15]: hence interval arithmetic may overestimate the values of the minimum and maximum of the function. This occurs because we have multiple occurrence of the same variable in f which are considered as independent during the calculation. But this overestimation has the following properties:

- it does not always occur: for example if f was defined as $x^2 + 2x$, then there will be no overestimation of the function for the range [3,5]
- let us define an interval evaluation as $[\underline{a}, \overline{a}]$, f_m, f_M the real minimum and maximum of the function over a given range and the size of the overestimation as $\text{Max}(f_m - \underline{a}, \overline{a} - f_M)$: in our example the size of the overestimation is 4. But this size decreases with the width of the input range.

Furthermore there are ways to decrease the size of the overestimation such as using the interval evaluation of the derivatives to determine if the function is monotonous over the considered ranges.

An interesting property of interval arithmetic is that it can be implemented to take into account round-off errors. For example if a calculation involves the number 1/3, then there is clearly no computer floating point number able to represent this number. There are two successive floating point numbers f_1, f_2 such that $f_1 < 1/3$ and $f_2 > 1/3$ and for any calculation involving this number the computer will round it to the f_i that is the closest to 1/3 (for instance on a Pentium the calculation of `3*(1/3.)-1` in C leads to -0.5510^{-16} instead of 0).

In interval arithmetic the number 1/3 will be represented by the range $[f_1, f_2]$ so that any interval evaluation involving this number will always include the exact value of the calculation. Numerous packages of interval arithmetic are available and our implementations are based on `BIAS/PROFIL`[1].

Notation:

The *width* of a range $[a, b]$ is defined as $b - a$ while the *mid-point* of the range is $(a + b)/2$. A *box* is a set of ranges and the width of the box is defined as the largest width of its ranges.

1.2 Interval Analysis

A key-point for solving a problem with interval analysis is to establish a property $\mathcal{P}(\mathbf{X})$, where $\mathbf{X}$ represents the unknowns of the problem, that will be true at a point $\mathbf{X_0}$ if $\mathbf{X_0}$ is a solution of the problem and will be wrong otherwise. Furthermore we will attach an algorithm $\mathcal{A}$ to the property so that if $\mathbf{X}$ lie in a given box $\mathcal{B}$, then $\mathcal{A}(\mathcal{B})$ will allow to determine if for all points in the box $\mathcal{P}$ is true, wrong or cannot be decided.

Let us give a simple example of such algorithms in the case where the property to satisfy is to find roughly the solutions of a system of equations $\mathbf{F}(\mathbf{X}) = \{F_1(\mathbf{X}) = 0, \ldots, F_n(\mathbf{X}) = 0\}$. The first step of the algorithm will be to compute the interval evaluation of each F_i for the given box. If the resulting interval for a given F_i does not include 0, then the algorithm will return wrong as no point in $\mathbf{X}$ will satisfy equation F_i. Now if the width of $\mathbf{X}$ is small and the interval evaluations of all the F_i is also small and includes 0, then the algorithm will return true: we will then obtain a set of small intervals such that if the system admits solutions, then they will be included into the set (as will be seen later on we are able to design more sophisticated algorithm that returns true only rightly).

For a given problem there are many different properties that have to be satisfied at the solutions. Although equivalent in mathematical terms they may not be so in term of interval analysis. Hence choosing the right property and designing the best associated algorithm is a key point for getting the best efficiency and requires some understanding of interval analysis.

A second step in any interval analysis method is the *filtering* algorithms. A filter takes as input a box $\mathcal{B}$ and returns either $\mathcal{B}$ or a smaller box $\mathcal{B}_f$ included in $\mathcal{B}$, the difference between $\mathcal{B}$ and $\mathcal{B}_f$ being guaranteed not include points for which $\mathcal{C}$ is true. A simple (but often efficient) filter for solving equations is the *2B* method. Assume that we have to find the solution(s) of $f(x) = x^2 + 3x + 1 = 0$ lying in the range [-10,10]. The interval evaluation of f for this range is [-29, 131] and therefore we cannot decide if this range includes a solution. But we may write f as $x^2 = -3x - 1$ and to have a solution of the equation there must be an intersection of the interval evaluation of the left and right hand-side terms. Hence the range for x^2 must be $[0, 100] \cap [-31, 29] = [0, 29]$ from which

[1] http://www.ti3.tu-harburg.de/Software/PROFILEnglisch.html

we deduce that x must be in $[-\sqrt{29}, \sqrt{29}]$. We may also use $x = (-x^2 - 1)/3$ from which we will have deduced that x must lie in $[-33, 1/3]$. Note that a filter may determine that the property is not satisfied for a given box.

A third (optional) step is the *existence operator* that takes as input a box $\mathcal{B}$ and may determine that a single point in a box $\mathcal{B}_u$ satisfies the property $\mathcal{C}$ and also provide a method to compute this point. A typical existence operator for a system of equations may be derived from the Kantorovitch theorem. Provided that the theorem is satisfied we may compute a box in which there is a single solution of the system and furthermore provide an initial guess of the solution so that the Newton-Raphson scheme is guaranteed to converge toward this solution. The algorithm $\mathcal{A}$ may thus include filtering methods and existence operators.

A key element in interval analysis is the *bisection process* of a box $\mathcal{B}$. In this process a variable x_i will be chosen, with range $[a, b]$ and two new boxes will be derived from $\mathcal{B}$ with ranges identical for all variables except for x_i whose range will be respectively $[a, (a + b)/2], [(a + b)/2, b]$.

With the above tools we may describe the structure of an interval analysis algorithm that belong to the branch-and-bound family. The algorithm will process a list $\mathcal{L}$ of boxes $\mathcal{B} = \{\mathcal{B}_1, \ldots \mathcal{B}_N\}$. The $i - th$ element of the list will be $\mathcal{B}_i$, while the total number of boxes at each iteration will be denoted r_k. When starting the algorithm there is a single box $\mathcal{B}_1$ in $\mathcal{L}$ but new boxes will be added during the process. The iterative scheme is defined as follows:

1. $i = 1$, $k = 1$, r_k=1
2. **while** $(k \leq r_k)$ **do**
3. **if** ($\mathcal{A}(\mathcal{B}_k)$=`wrong`) **then**
4. $r_{k+1} = r_k$
5. $k = k + 1$
6. **else**
7. **if** ($\mathcal{A}(\mathcal{B}_k)$=`true`) **then**
8. **store $\mathcal{B}$ as solution**
9. **else**
10. bisect $\mathcal{B}_k$ into $\mathcal{B}_k^1$, $\mathcal{B}_k^2$
11. store $\mathcal{B}_k^1$ as $\mathcal{B}_{r_k+1}$, $\mathcal{B}_k^2$ as $\mathcal{B}_{r_k+2}$
12. $r_{k+1} = r_k + 2$
13. $k = k + 1$
14. **end if**
15. **end if**
16. **end do**

This algorithm stops when all the boxes in $\mathcal{L}$ have been processed. However it may happen that even at a point algorithm $\mathcal{A}$ cannot decide if the property is true or wrong (e.g. because of round-off errors). A flag allows to manage such case.

As any branch-and-bound algorithm the worst case complexity is exponential because of the bisection. However the practical complexity may be good as will be shown in the following examples.

2 Robotics Examples

2.1 Kinematics

Solving inverse and direct kinematics problems were among the first to be addressed with interval analysis [14], however with mixed results as the methods were not as advanced as today. Nowadays interval analysis has proved to be valuable even for the most complex kinematics problem. Here the problem amounts to solve a system of equations that is derived either from the inverse or direct kinematics. The first problem to address is to determine which set of equations are the most appropriate. We will consider here the direct kinematics problem of parallel robots, which is among the most challenging (figure 1).

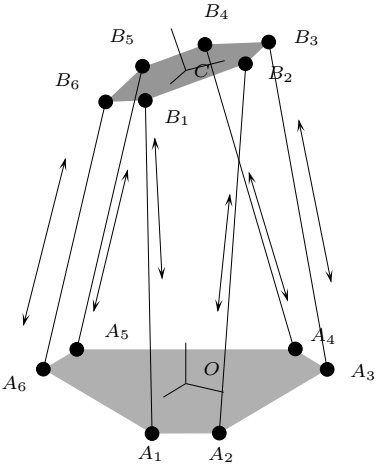

Fig. 1. A parallel robot

Here being given the leg lengths we have to determine what can be the poses of the platform. The most simple inverse kinematic relations give the square of the 6 leg lengths ρ_i as function of pose parameters which are the coordinates of the center C of the platform and three angles that define its orientation. The drawback of such equations is that there are multiple occurrences of the variables in the equations, which is a major disadvantage for interval analysis. Assume now that the platform pose is defined by the three coordinates of the anchor points $B_1, B_2, B3$. Being given these coordinates and the geometry of the platform we may derive the coordinates of B_4, B_5, B_6 as $\mathbf{OB_i} = \lambda_i \mathbf{OB_1} + \alpha_i \mathbf{OB_2} + \beta_i \mathbf{OB_3}$ in which $\lambda_i, \alpha_i, \beta_i$ are known constants. We then derive 6 constraint equations as $||\mathbf{A_iB_i}||^2 = \rho_i^2$ for i in [1,6] and 3 additional constraints $||\mathbf{B_iB_j}||^2 = d_{ij}^2$ for i, j in [1,3], $i \neq j$ in which d_{ij} are known distances. This set of 9 equations in 9 unknowns has the advantage that the variables appear only once in each equation and consequently there will be no overestimation during the interval evaluation. Furthermore the geometry of the problem imposes natural bounds for

the variables The algorithm to solve these equations uses filtering methods such as the `2B, 3B`, interval Newton and existence operators based on Kantorovitch theorem and Neumaier exclusion test and is fully described in [9]. This algorithm provides *exact* solutions in the sense that no solution can be lost and a solution can be calculated with an arbitrary accuracy. This feature is shared only with the solving method based on Groebner basis [16] as other approaches such as elimination or continuation cannot guarantee both properties.

In terms of computation time the algorithm will find all solutions in a time that ranges from a few seconds to one minute which makes it the second most efficient solving approach (the best Groebner basis implementation requires between one second to 30 seconds). But, opposite to the Groebner basis approach, the computation time will decrease with the size of the search space. Especially it will compete with the Newton-Raphson scheme for real-time use, while still guaranteeing to find the correct solution (or detecting that the robot is close to a singularity) which is not the case of the usual iterative scheme.

2.2 Workspace Analysis and Motion Planning

A classical robotics problem is to determine various types of workspace of a given robot. Usually the workspace is defined as the set of poses that satisfy inequality constraints $\mathbf{C}(\mathbf{X}) \leq 0$ and finding a description as a set of boxes of the workspace up to a pre-defined accuracy ϵ is based on the interval evaluation of the constraints. If for a ball $\mathcal{B}$ all upper bounds of the evaluations are negative, then $\mathcal{B}$ belongs to the workspace. If one of the evaluation has a positive lower bound, then $\mathcal{B}$ does not belong to the workspace. Otherwise the box is bisected until either one of the above two properties is verified or the width of the box is lower than ϵ, in which case the box is discarded [10].

By using the same algorithm it is easy to check if a given volume is included in the workspace if the volume can be described by explicit functions involving bounded parameters (for example points of a sphere will be defined by the distance between the center and the point and two angles). Here uncertainties in the modeling parameters $\mathbf{P}$ (for example the coordinates of the anchor points A_i, B_i) may be managed if they are restricted to lie within ranges that are defined by the manufacturing tolerances. These intervals may be directly introduced in the constraints inequalities, still allowing their interval evaluations. Hence even at a given pose the constraint interval evaluation will be an interval, that may be overestimated because we may have multiple occurrence of $\mathbf{P}$. It may thus occur that at a pose we cannot verify if one constraint is satisfied in which case we will have to add $\mathbf{P}$ to the variable used in the interval analysis algorithm.

We may also design algorithms that calculate more complex workspace. For example we may have to compute the region of the workspace of the robot in which the eigenvalues of $\mathbf{K} = \mathbf{J^T J}$, where $\mathbf{J}$ is the Jacobian matrix, are all included in a given range [a,b]. Let us assume that the characteristic polynomial $P_c = \sum_{i=0}^{i=n} a_i(\mathbf{X})\lambda^i$ of $\mathbf{K}$ is available (but this is not a necessary condition). We have thus to find the poses in which $\mathbf{C}(\mathbf{X}) \leq 0$ (C-1) and such that all the roots in λ of P_c are in the range $[a, b]$ (C-2). We will first design an algorithm $\mathcal{A}_1$

that checks if (C-2) is satisfied for a set of poses defined by a box. In this case the coefficients of P_c are intervals (i.e. P_c is is fact a family of polynomials) but we may still apply classical algebraic geometry theorems that will provide an interval $[u_1, u_2]$ including all the bounds for all polynomials in P_c. If $[u_1, u_2]$ is included in $[a, b]$, then all poses in the box satisfy (C-2) while if the intersection of $[u_1, u_2]$, $[a, b]$ is empty all poses in the box violate (C-2). Assume now that the relative complement $\mathcal{I}$ of $[a, b]$ in $[u_1, u_2]$ (i.e. the numbers that are in $[u_1, u_2]$ but not in $[a, b]$) is not empty: $\mathcal{I}$ is constituted either of one interval (if u_1 or u_2 belongs to $[a, b]$) or of two intervals. We design the interval analysis algorithm $\mathcal{A}_1$ whose unknown is λ and input is $\mathcal{I}$ and is based on the interval evaluation of P_c for discarding boxes included in $\mathcal{I}$ that cannot cancel P_c. If $\mathcal{A}_1$ completes, then all the roots of P_c lie in $[a, b]$ and (C-2) is satisfied for the box. Clearly as P_c has interval coefficients it may happen that $\mathcal{A}_1$ cannot decide if a box satisfies (C-2).

Hence we will embed $\mathcal{A}_1$ in another interval analysis algorithm $\mathcal{A}$ whose variables are $\mathbf{X}$. This algorithm is basically the same than the one described for computing the workspace except that if a box satisfies the workspace constraints $\mathbf{C}(\mathbf{X}) \leq 0$, then it is sent to the $\mathcal{A}_1$ algorithm that is allowed to perform only a limited number of bisections. Indeed for large $\mathbf{X}$ boxes $\mathcal{A}_1$ will probably not be able to complete and therefore the allowed maximum number of bisection g should be adjusted according to the width of the box (typically as $1/g$). If $\mathcal{A}_1$ finishes (detect that C-2 is not satisfied), then the box is stored as a solution (discarded), otherwise the box is bisected if its width is sufficient. This algorithm has been able to manage rather complicated problems [3].

2.3 Singularity Analysis

Singularity analysis is an important problem especially for closed-loop robots that cannot be controlled at a singular pose. If $\mathbf{J^{-1}}$ denotes the inverse kinematic Jacobian of the robot, singular poses are defined as the poses for which $|\mathbf{J^{-1}}| = 0$.

Major practical problems to consider are to be able to detect if a singularity occurs within a given single-component workspace or trajectory. Solving those problems is critical for the safety of the robot as being close to a singularity may lead to very large forces in the legs of the robot. But finding the singularities is complex as if usually we have a closed-form for $\mathbf{J^{-1}}$, it may however be extremely difficult to obtain a closed-form of the determinant. Furthermore as safety is a critical issue we have to consider uncertainties in the location of the A_i, B_i. For solving this problem we will use the continuity of the determinant with respect to $\mathbf{X}$. Assume that at some arbitrary pose $\mathbf{X_1}$ of the workspace we are able to show that $|\mathbf{J^{-1}}|$ as a constant sign (say positive). Then if we are able to demonstrate that at some other pose $\mathbf{X_2}$ the determinant is negative any path connecting $\mathbf{X_1}$ to $\mathbf{X_2}$ must cross a singularity and consequently the workspace is singular. On the other hand if no pose $\mathbf{X_2}$ can be found, then the workspace is singularity-free.

Hence detecting a $\mathbf{X_2}$ pose is the property we will investigate with interval analysis. To begin with we will assume that there are no uncertainties in the

robot modeling. Assume that $\mathbf{X}$ belongs to a box: using interval arithmetic we are able to calculate an interval evaluation of each element of $\mathbf{J^{-1}}$, i.e. this matrix is now an *interval matrix*. Classical algorithm for calculating the determinant of a matrix such as row expansion or Gaussian elimination may be extended to interval matrix, thereby allowing to get an interval evaluation of $|\mathbf{J^{-1}}|$ for the box. We will calculate this interval evaluation for a pose $\mathbf{X_t}$ chosen arbitrary in the workspace. If at this pose the interval evaluation does not allow to show that the determinant is of constant sign (i.e. the lower and upper bound of the evaluation have not same sign), then we will not be able to show that the workspace is singularity-free (but the following algorithm will still be able to detect that the workspace includes a singularity). We will now assume that a pose $\mathbf{X_1}$ has been found.

Having the possibility of computing an interval evaluation of the determinant is already sufficient to design an algorithm $\mathcal{A}$. But additional filters will allow to drastically reduce the computation time. For a given box $\mathcal{B}$ we will first calculate the interval evaluation of the determinant U of the pre-conditioned matrix $\mathbf{KJ^{-1}}(\mathcal{B})$ where $\mathbf{K}$ is a scalar matrix chosen as the inverse of $\mathbf{J^{-1}}$ computed for the mid-point of $\mathcal{B}$ (if it exists). As $U = |\mathbf{KJ^{-1}}| = |\mathbf{K}||\mathbf{J^{-1}}|$ if $U, |\mathbf{K}|$ have a constant sign, then we may deduce that the sign of $|\mathbf{J^{-1}}(\mathcal{B})|$ is constant. If this sign is negative, then the workspace includes a singularity (all poses in $\mathcal{B}$ are $\mathbf{X_2}$ poses), otherwise $\mathcal{B}$ does not include $\mathbf{X_2}$ pose and may discarded. Note that to reduce the overestimation of U it is necessary to compute a closed-form of $\mathbf{KJ^{-1}}$ and to re-arrange the elements of this matrix to reduce the number of multiple occurrence of the elements of $\mathbf{X}$. A second filter involves a theorem provided by Rohn [7]: consider the set $\mathcal{E}$ of extremal matrices derived from the interval matrix $\mathbf{J^{-1}}(\mathcal{B})$, i.e. all the scalar matrices whose elements are either the lower or upper bound of the corresponding interval element in $\mathbf{J^{-1}}(\mathcal{B})$. If all matrices in $\mathcal{E}$ have a determinant of the same sign, then there is no singular matrix in $\mathbf{J^{-1}}(\mathcal{B})$. If the determinant of $\mathbf{J^{-1}}$ at an arbitrary pose included in $\mathcal{B}$ is positive, then $\mathcal{B}$ does not include $\mathbf{X_2}$ pose and can be discarded. Note that for a $n \times n$ $\mathbf{J^{-1}}$ matrix $\mathcal{E}$ has 2^{n^2} matrices but Rohn proves that it is sufficient to consider only a subset of $\mathcal{E}$ that has 2^{2n-1} elements. Using both filters algorithm $\mathcal{A}$ is very efficient: checking the regularity of $\mathbf{J^{-1}}$ over a large workspace requires less than one second [11].

Having uncertainties in the robot modeling will not change the structure of the $\mathcal{A}$ algorithm. If the intervals describing these uncertainties have a small width we will just take them into account for computing the interval evaluation of the elements of $\mathbf{J^{-1}}$. However it may occur that a given pose we will not be able to determine the sign of $|\mathbf{J^{-1}}|$ because of these uncertainties, in which case we will add them as unknowns in the $\mathcal{A}$ algorithm: this significantly increases the computation time (a few hours may be required) but allows one to obtain the largest level of safety: if the algorithm proves that the workspace is singularity-free, then so it is for the **real** robot.

3 Conclusion

As shown by the examples (which are not exhaustive: for example we may have also presented results on optimal design or performance analysis of robots) interval analysis is an approach that may solve efficiently very difficult robotics problems and provide a guaranteed result even, as it is the case in practice, if the robot modeling is uncertain. An important point is that it allows one to consider round-off errors that are not so seldom and of which only few roboticists are aware. But there is no such thing as a free lunch and interval analysis requires an in-depth analysis of the problem at hand and a large expertise in this field for designing efficient algorithms. Fortunately there are many teams involved in interval analysis that are eager to help solving difficult problems !

References

[1] Ashokaraj, I., et al.: Sensor based robot localisation and navigation: Using interval analysis and extended kalman filter. In: 5th Asian Control Conference, Melbourne, July 20-23 (2004)
[2] Carreras, C., Walker, I.D.: Interval methods for fault-tree analysis in robotics. IEEE Trans. on Reliability 50(1), 3–11 (2001)
[3] Chablat, D., Wenger, P., Majou, F., Merlet, J.-P.: An interval analysis based study for the design and the comparison of three-degrees-of-freedom parallel kinematic machine. Int. J. of Robotics Research 23(6), 615–624 (2004)
[4] Clerenti, A., et al.: Imprecision and uncertainty quantification for the problem of mobile robot localization. In: Performance Metrics for Intelligent Systems Workshop, Gaithersburg, September 16-18 (2003)
[5] Daney, D., Andreff, N., Chabert, G., Papegay, Y.: Interval method for calibration of parallel robots: a vision-based experimentation. Mechanism and Machine Theory 41(8), 929–944 (2006)
[6] Hansen, E.: Global optimization using interval analysis. Marcel Dekker, New York (2004)
[7] Jansson, C., Rohn, J.: An algorithm for checking regularity of interval matrices. SIAM Journal on Matrix Analysis and Applications 20(3), 756–776 (1999)
[8] Jaulin, L., Kieffer, M., Didrit, O., Walter, E.: Applied Interval Analysis. Springer, Heidelberg (2001)
[9] Merlet, J.-P.: Solving the forward kinematics of a Gough-type parallel manipulator with interval analysis. Int. J. of Robotics Research 23(3), 221–236 (2004)
[10] Merlet, J.-P.: Determination of 6D workspaces of Gough-type parallel manipulator and comparison between different geometries. Int. J. of Robotics Research 18(9), 902–916 (1999)
[11] Merlet, J.-P., Donelan, P.: On the regularity of the inverse jacobian of parallel robot. In: ARK, Ljubljana, June 26-29, pp. 41–48 (2006)
[12] Moore, R.E.: Methods and Applications of Interval Analysis. SIAM Studies in Applied Mathematics (1979)
[13] Piazzi, A., Visioli, A.: Global minimum-jerk trajectory planning of robot manipulators. Trans. on Industrial Electronics 47(1), 140–149 (2000)

[14] Rao, R., Asaithambi, A., Agrawal, S.K.: Inverse kinematic solution of robot manipulators using interval analysis. J. of Mechanical Design 120(1), 147–150 (1998)
[15] Redon, S., et al.: Fast continuous collision detection for articulated models. In: 9th ACM symposium on Solid modeling and applications, Genoa, pp. 145–156 (2004)
[16] Rouillier, F.: Real roots counting for some robotics problems. In: Merlet, J.-P., Ravani, B. (eds.) Computational Kinematics, pp. 73–82. Kluwer, Dordrecht (1995)
[17] Seignez, E., et al.: Experimental vehicle localization by bounded-error state estimation using interval analysis. In: IEEE/RJS IROS, Edmonton, August 2-6 (2005)
[18] Wu, W., Rao, S.S.: Interval approach for the modeling of tolerances and clearances in mechanism analysis. J. of Mechanical Design 126(4), 581–592 (2004)