INTERVAL PROPAGATION FOR SOLVING PARALLEL SPHERICAL MECHANISMS

Enric Celaya
Institut de Robòtica i Informàtica Industrial
Llorens i Artigas, 4-6, 08028 Barcelona
celaya@iri.upc.es

Abstract An interval propagation method for spherical kinematic loops is used in a branch and prune algorithm to solve the direct kinematics of parallel spherical mechanisms. The algorithm finds all solutions with a desired resolution. The use of specific properties of the rotation equations involved allows the method to be more efficient than more general algorithms for this problem.

Keywords: Interval methods, interval propagation, parallel spherical mechanisms.

1. Introduction

The direct kinematics of parallel manipulators has proven to be very complex. In the spatial domain, for example, it is well known that the general Stewart-Gough platform can have as many as 40 different solutions for a given position of the actuators (Dietmaier, 1998). If we stay into the domain of spherical mechanisms, we still find that the parallel platform studied in Gosselin et al., 1994 is described by an 8-degree polynomial and has 8 possible solutions. An alternative to the traditional algebraic approaches to find the solutions of the direct kinematics of parallel manipulators consists in using interval methods to progressively reduce the length of the interval containing a solution through a branch and prune process, until the desired precision is reached. Typical problems of branch and prune interval methods are the sensitivity to the size of the initial intervals, the availability of an efficient pruning mechanism, and the decision about how much pruning must be done before branching takes place.

Here we present an interval method for spherical mechanisms that is well suited for this problem, since it uses an efficient algorithm for determining the set of all allowed values for a variable in a spherical kinematic loop. Besides this, a low cost interval propagation algorithm that provides the set of values for one variable compatible with a given interval of another one with no overestimation is used for pruning. Using enough precision in the computations, the method grants to find all the solutions of the mechanism. Interval methods are also well suited for underconstrained problems whose solution presents one or more degrees of freedom, so that the set of allowed values for a variable is not a finite number of isolated points, but one or more intervals of values. In this case, the algorithm will return a set of boxes (of the size of the working resolution) containing the solution.

This paper is structured as follows: first, the procedure to find the allowed values for variables in rotation equations is described in Section 2, while the algorithm for interval propagation is described in Section 3. In Section 4, the description of the branch and prune algorithm for the solution of the direct kinematics of parallel spherical manipulators is presented, and an application example is given in Section 5. The main conclusions of the paper are resumed in Section 6.

2. Solution sets for variables in spherical kinematic loops

A spherical kinematic loop yields a rotation equation of the form:

$$\prod_{i=1}^{n} \mathbf{Rx}(\theta_i)\,\mathbf{Rz}(\alpha_i) = I, \tag{1}$$

where Rx and Rz are rotations around the x and z axis, respectively, θ_i are the rotation variables, and α_i are constant angular parameters that can be assumed to be in the interval $[0, \pi]$. The rotation equation written in this form has an intuitive geometric interpretation, according to which α_i are the sides of a spherical polygon, and θ_i its exterior angles. For $n > 3$, equation (1) has, in general, an infinite number of solutions, that is, each variable θ_i can take an infinite number of values. In Celaya and Torras, 1994, a way to describe the solution of a rotation equation is proposed, that consists in determining the set of allowed values for each variable. Such a description provides a kind of information that is not directly available when the solution is expressed in algebraic form, as for example, the enclosing box (or set of boxes) that contains all the solution vectors $(\theta_1, ..., \theta_n)$. This description permits obtaining any solution vector by sequentially fixing the value of a variable within its solution set, then finding the set of allowed values for the next variable in the resulting equation, fixing it to a value within this set, and so on.

The computation of the set of allowed values for a variable θ_s is simple provided all parameters α_j, $(j{\neq}s)$ are $\leq\pi/2$, a condition that can always be fulfilled, since the equation can be rewritten by appropriately substituting sides of length $\alpha_i{>}\pi/2$ by sides of length, $(\pi{-}\alpha_i)$, what corresponds to consider a supplementary polygon whose relation with

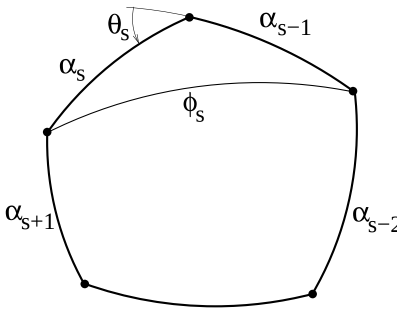

Figure 1. Solving for θ_s in a spherical polygon.

the original one is straightforward (Chiang, 1988). Denoting by ϕ_s the arc delimited by the extreme points of sides α_s and α_{s-1} (Fig. 1), we have, according to the cosine law of spherical trigonometry:

$$\cos\phi_s = \cos\alpha_{s-1}\cos\alpha_s - \sin\alpha_{s-1}\sin\alpha_s\cos\theta_s, \tag{2}$$

On the other hand, ϕ_s is also the arc delimited by the extreme points of the chain of sides α_j, ($j{\neq}s$,s-1), which is a function of θ_j, ($j{\neq}s$-1,s, s+1). Assuming that $\alpha_j{\leq}\pi/2$, $\forall j{\neq}s$, it can be shown that $\cos\phi_s$ can only take values in the interval $[\cos\sigma, \cos\delta]$, where $\sigma{=}\min(\pi, \sum_{i\neq s,s-1}\alpha_i)$, and $\delta{=}\max(0, \alpha_M{-}\sum_{i\neq M,s,s-1}\alpha_i)$, with $\alpha_M{=}\max_{i\neq s,s-1}(\alpha_i)$ (see Celaya, 1992 or Celaya and Torras, 1994 for details). Then, using (2), we obtain that the allowed values for cos θ_s are those that satisfy:

$$\frac{\cos\alpha_{s-1}\cos\alpha_s - \cos\delta}{\sin\alpha_{s-1}\sin\alpha_s} \leq \cos\theta_s \leq \frac{\cos\alpha_{s-1}\cos\alpha_s - \cos\sigma}{\sin\alpha_{s-1}\sin\alpha_s} \tag{3}$$

Taking into account that the arccos function is double-valued, the set of allowed values for θ_s, according to (3), will consist of two symmetric intervals, except when the left hand side of (3) is ≤ -1 or the right-hand side is ≥ 1 (or both), in which cases the two intervals join in a single one.

A function denoted by $\mathit{SolveSpherical}(eq, \theta)$ will be used in the remaining of this paper to provide the set of solution values of variable θ in the rotation equation eq as explained above.

3. Interval propagation between variables in spherical kinematic loops

To obtain the set of values $S_i[\theta_j{=}x_j]$ of variable θ_i that are compatible with the value x_j of θ_j, we can rewrite the rotation equation (1) substituting θ_j for this value and find the set of allowed values for θ_i in the resulting equation using the $SolveSpherical()$ function. But, if what we are interested in is the set of values of θ_i that are compatible with an *interval of values* I_j of θ_j, then we will use the propagation algorithm described next.

Clearly, the set $S_i[\theta_j{\in}I_j]$ of values of θ_i that are compatible with $\theta_j{\in}I_j$, is a subset of $S_i{=}SolveSpherical(eq, \theta_i)$. Since $S_i[\theta_j{\in}I_j]$ is defined as the set of all those values of θ_i that are compatible with any value of $\theta_j{\in}I_j$, it must include, in particular, all the values compatible with the extrema of the interval. Thus, if $I_j{=}[a, b]$, then $(S_i[\theta_j{=}a]{\cup}S_i[\theta_j{=}b]){\subset}S_i[\theta_j{\in}I_j]$. Therefore, all we need is to check the compatibility in the difference set

$$D_i = S_i - (S_i[\theta_j = a] \cup S_i[\theta_j = b]).$$

Note that the set D_i is composed of at most six non-intersecting intervals, since each of the S_i consists in, at most, two separate intervals. Due to the symmetry properties of the functions involved, it is possible to show (see Celaya and Torras, 1994) that each of the intervals composing D_i is either completely included in, or completely excluded from $S_i[\theta_j \in I_j]$. As a consequence, it is sufficient to check the compatibility of a single value into each interval forming D_i. This is done by choosing the central point x_k of each interval (six in the worst case), substituting it in the equation, and computing the intersection $S_j[\theta_i = x_k] \cap I_j$. If this intersection is $\neq \emptyset$, then the whole interval must be included in $S_i[\theta_j \in I_j]$, otherwise it must be excluded. A function denoted by $Prop(eq, \theta_j, I_j, \theta_i)$ implementing this propagation algorithm will be used in the remaining of this paper to compute the set $S_i[\theta_j \in I_j]$.

4. Direct kinematics of parallel spherical mechanisms

Solving the direct kinematics of a spherical mechanism consists in computing the orientation of the output link from the values of those variables of the mechanism's equations that are taken as input variables. For sure, the orientation of the output link can be determined once the values of all variables in the mechanism's equations are known, though, in general, this orientation can be directly expressed in terms of just a subset of the variables that can be considered as the *output variables* of the mechanism. So, we can formulate the problem of solving the direct kinematics of a mechanism as that of finding the values that the output variables can take when the input variables are fixed to some values.

Consider a parallel spherical mechanism with two independent kinematic loops, i.e., two rotation equations of the form (1). Let k be the number of variables shared between the two equations, excluding the input variables (since they remain fixed during the solution process).

Obviously, if k=0, each equation can be solved independently, and the problem does not really correspond to a parallel mechanism. In this paper, we will restrict our discussion to the cases of k=1 and k=2.

4.1 Two equations sharing one variable

If k=1, the set of solution values for the shared variable θ_s can be obtained by computing the set of values allowed by each equation for this variable and intersecting them. Then, for each solution value of θ_s, solution values for all the output variables can be found by solving the corresponding equation with the fixed value of θ_s.

4.2 Two equations sharing two variables

In the case of k=2, to obtain the solution values for the shared variables θ_s and θ_t, it is not sufficient to find the sets of values allowed for them by both equations: it is also necessary that the values of the two variables are simultaneously compatible with each other in the two equations. To grant this, the propagation algorithm can be used in an iterative process: Once the set of values for θ_s allowed by the two equations is obtained, it is propagated to θ_t through both equations, and the resulting sets are intersected. The so obtained set of values for θ_t is then propagated back to θ_s through both equations and intersected with the initial set for θ_s to get a smaller set for it. This process is iterated until the convergence to a stable set for θ_s is reached.

If the process converged to an isolated value, this is a solution for θ_s, and the corresponding values of θ_t and of all output variables can be obtained as in the case of k=1, after fixing θ_s to the solution value. If the process converged to a set that is not an isolated value, it must be split in two subsets, and the process must be recursively repeated in each subset until all of them either vanish or converge to isolated values (or to subsets of a size smaller than the working resolution ϵ). The resulting algorithm, called $SolveParallelSpherical()$, is shown in Fig. 2. It receives 6 arguments, the two rotation equations, the shared variables θ_s and θ_t, and their initial sets of values I_s and I_t into which solution values are looked for. The algorithm returns the set of all solution values for the shared variable θ_s contained in I_s.

In order to improve efficiency, the loop for the convergence to a stable set of values for θ_s is stopped as soon as the reduction obtained in a single iteration is less than 50%, in which case, the set is immediately split in two halves. Since, when one half of the set does not contain a solution, it usually vanishes after the first iterations, this strategy makes that most iterations result in a reduction of the set in, at less, one half,

```
SolveParallelSpherical(eqA, eqB, $\theta_s, I_s, \theta_t, I_t$)

$A_s$ = SolveSpherical(eqA, $\theta_s$)
$B_s$ = SolveSpherical(eqB, $\theta_s$)
$I_s = I_s \cap A_s \cap B_s$
L0 = $\infty$
Ls = length($I_s$)

DO WHILE (Ls > $\epsilon$) AND (Ls < L0/2)
    $A_t$ = Prop(eqA, $\theta_s, I_s, \theta_t$)
    $B_t$ = Prop(eqB, $\theta_s, I_s, \theta_t$)
    $I_t = I_t \cap A_t \cap B_t$
    $A_s$ = Prop(eqA, $\theta_t, I_t, \theta_s$)
    $B_s$ = Prop(eqB, $\theta_t, I_t, \theta_s$)
    $I_s = I_s \cap A_s \cap B_s$
    L0 = Ls
    Ls = length($I_s$)
END DO

IF (Ls < $\epsilon$) THEN RETURN($I_s$)
ELSE
    Split $I_s$ into $I1_s$, $I2_s$
    $S1$ = SolveParallelSpherical(eqA, eqB, $\theta_s, I1_s, \theta_t, I_t$)
    $S2$ = SolveParallelSpherical(eqA, eqB, $\theta_s, I2_s, \theta_t, I_t$)
    RETURN($S1 \cup S2$)
END
```

Figure 2. Recursive branch and prune algorithm to find the solution values of variable θ_s for two rotation equations sharing variables θ_s and θ_t.

which implies exponential convergence. Other improvements have been implemented to skip the execution of propagation steps through one of the equations when the resulting interval reduction is not large enough.

5. Example

We consider the spherical parallel manipulator analized in Gosselin et al., 1994. It consists of two platforms connected by three identical legs, each formed by two links joined by a revolute pair. The schematics of the general mechanism, assuming some symmetries in the parameters, is shown in Fig. 3.

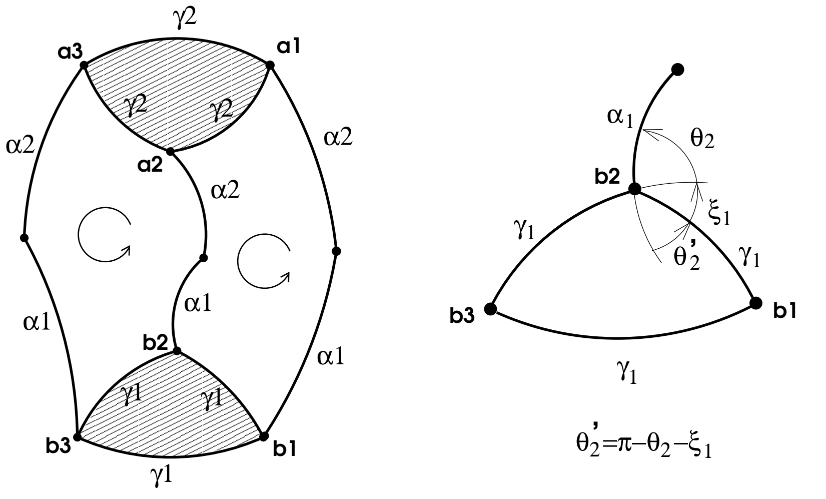

Figure 3. Parallel spherical manipulator and relation between the shared variables.

Two independent rotation equations are obtained from cycles involving legs 1,2 and 2,3, respectively:

$$\begin{aligned}&\mathbf{Rx}(\theta_1)\,\mathbf{Rz}(\alpha_1)\,\mathbf{Rx}(\mu_1)\,\mathbf{Rz}(\alpha_2)\,\mathbf{Rx}(\rho_1)\,\mathbf{Rz}(\gamma_2)\\ &\qquad\qquad\mathbf{Rx}(\rho'_2)\,\mathbf{Rz}(\alpha_2)\,\mathbf{Rx}(-\mu_2)\,\mathbf{Rz}(\alpha_1)\,\mathbf{Rx}(\theta'_2)\,\mathbf{Rz}(\gamma_1) = I,\end{aligned}$$

$$\begin{aligned}&\mathbf{Rx}(\theta_2)\,\mathbf{Rz}(\alpha_1)\,\mathbf{Rx}(\mu_2)\,\mathbf{Rz}(\alpha_2)\,\mathbf{Rx}(\rho_2)\,\mathbf{Rz}(\gamma_2)\\ &\qquad\qquad\mathbf{Rx}(\rho'_3)\,\mathbf{Rz}(\alpha_2)\,\mathbf{Rx}(-\mu_3)\,\mathbf{Rz}(\alpha_1)\,\mathbf{Rx}(\theta'_3)\,\mathbf{Rz}(\gamma_1) = I,\end{aligned}$$

where the inputs are the angles $\theta_1, \theta_2, \theta_3$ of the joints b_i of the base link, that we define as shown in Fig. 3. Angles θ'_i are related to the input angles θ_i in the form: $\theta'_i = \pi$-θ_i-ξ_1, where ξ_1 is the exterior angle of the spherical equilateral triangle of sides γ_i, which, according to the cosine law, is given by $\cos\xi_1 = (\cos^2\gamma_1 - \cos\gamma_1)/\sin^2\gamma_1$. Similarly, for the angles ρ_i of the end effector joints a_i, we have $\rho'_i = \pi$-ρ_i-ξ_2, with $\cos\xi_2 = (\cos^2\gamma_2 - \cos\gamma_2)/\sin^2\gamma_2$.

In summary, once the input variables θ_i are given fixed values, we have two rotation equations sharing two variables, μ_2 and ρ_2, that can be solved with the *SolveParallelSpherical()* algorithm.

Taking the same parameter values used in Gosselin et al., 1994, γ_1=$\pi/7$, γ_2=$\pi/2$, α_1=1.12, α_2=1.32, and solving the mechanism for the input values θ_i=$-\pi/3$, the algorithm returns the eight solutions shown in Tb. 2, whose correctness can be checked by substituting these values into the rotation equations. In this case, to find the 8 solutions the algorithm performs the minimal number of interval partitions: 7, and the mean reduction rate at each iteration of the pruning process is above 75%.

Table 1. Solution values for $\theta_1 = \theta_2 = \theta_3 = -\pi/3$.

	1	2	3	4	5	6	7	8
ρ_2	3.71014	4.14384	.01532	5.45702	6.03943	1.55548	1.81455	2.39696
μ_2	3.74975	3.10884	.96645	5.27620	4.75242	5.89215	2.10617	1.58239
ρ_1	2.39695	1.81454	.01532	4.14384	3.71013	1.55548	5.45703	6.03944
μ_1	1.58239	2.10617	.96645	3.10884	3.74975	5.89215	5.27620	4.75242
ρ_3	6.03943	5.45702	.01532	1.81455	2.39696	1.55548	4.14384	3.71013
μ_3	4.75242	5.27620	.96645	2.10618	1.58239	5.89215	3.10884	3.74975

6. Conclusions

Interval propagation methods are a good alternative to numerical approaches when closed form algebraic solutions are not available. The method presented here is specific for spherical mechanisms, thus it may take advantage of the availability of the interval propagation algorithm to reduce the solution interval, what makes the algorithm better suited for this problem than more general interval methods. The algorithm is rather insensitive to the selection of the initial intervals, since it starts by reducing the sets of values for the variables to those allowed by the two equations, which in combination with the propagation method, tends to quickly isolate the different solutions.

7. Acknowledgements

This research was financed by the CICYT, project TIC2000-0696.

References

Celaya, E. (1992), “Geometric Reasoning for the Determination of the Position of Objects Linked by Spatial Relationships”, Ph.D. dissertation, Dept. LSI, Universitat Politècnica de Catalunya.

Celaya, E. and Torras, C. (1994), “Solving Multiloop Linkages with Limited-Range Joints”, *Mechanism and Machine Theory*, vol. 29, No. 3, pp. 373-391.

Chiang, C.H. (1988), *Kinematics of spherical mechanisms.* Cambridge University Press. Cambridge.

Dietmaier, P. (1998), “The Stewart-Gough platform of general geometry can have 40 real postures” in ARK, Strobl, pp. 7-16.

Gosselin, C.M., Sefrioui, J., and Richard, M.J. (1994), “On the Direct Kinematics of Spherical Three-Degree-of-Freedom Parallel Manipulators of General Architecture” *ASME Journal of Mechanical Design*, Vol. 116, No. 2, pp. 594–598.