Closed-Form Position Analysis of Variable Geometry Trusses

Josep M. Porta[*], Federico Thomas

Institut de Robòtica i Informàtica Industrial, CSIC-UPC
Llorens i Artigas 4-6, 08028, Barcelona, Spain

Abstract

Variable geometry trusses are composed, in general, of unit cells which can be modeled as bars connected by spherical joints. Under mild conditions, it has been shown that the only feasible cells are topologically equivalent to bipyramids. Unfortunately, using standard formulations, the closed-form position analysis of bipyramids is not a trivial task. Actually, it has only been achieved for bipyramids with up to 7 vertices, whose closure polynomial has been shown to be of order 24. In this paper, using a distance-based formulation and a kinematic inversion for fans of tetrahedra, the problem is solved for bipyramids with up to 11 vertices, whose closure polynomial is of degree 896. No other position analysis problem leading to such a high-order closure polynomial has been previously solved.

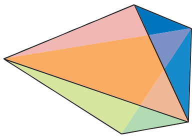
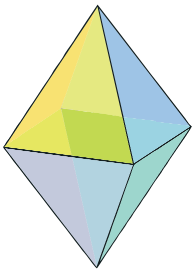
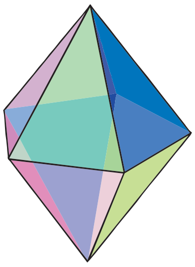
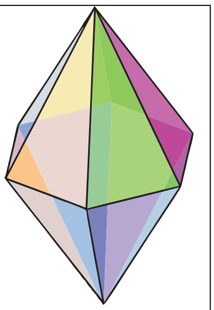
Keywords: Variable Geometry Trusses, Position Analysis, Bipyramids, Fans of tetrahedra, Coordinate-free formulations, Distance-based formulations

1. Introduction

A truss is a structure where each element, typically a bar, only supports tension or compression forces because it is connected to other bars through what are assumed to be multiple spherical joints [1], although in some cases the joints may be separated in the actual construction [2, 3]. While rigid trusses have been widely used in construction, passive mobile trusses are commonly used, for instance, as shock absorbers. The advent of automation opened the possibility to build active trusses, *i.e.*, trusses which can actively vary their geometry as needed [4]. The motion of such devices is commonly achieved by having actuated bars with variable length. Actually, the number of such bars gives the number of degrees of freedom of the device. Arbitrary variable geometry trusses may be defined but, to facilitate their design, analysis, construction, and control, they are typically built with repetitions of a given unit cell. A mechanism composed of several unit cells can have a large workspace, like a serial robot, but at the same time the high stiffness of a parallel one. This is why variable geometry trusses are sometimes considered a generalization of the serial/parallel robots [5]. Due to their exceptional stiffness to weight ratio,

[*]Corresponding author at: Tel: +34 934015751. Fax: +34 934015750.
Email addresses: porta@iri.upc.edu (Josep M. Porta), fthomas@iri.upc.edu (Federico Thomas)

Table 1: The cells traditionally considered as the building blocks of variable geometry trusses.

Truss	 Hexahedron	 Octahedron	 Decahedron	 Dodecahedron
Order (N)	1	2	3	4
Vertices $(N+4)$	5	6	7	8
Faces $(2N+4)$	6	8	10	12
Edges $(3N+6)$	9	12	15	18
Spherical joints $(5N+8)$	13	18	23	28
Quadratic equations $(3N+3)$	6	9	12	15
Polynomial degree $(N2^N)$	2	8	24	64
References	[14]	[15], [3]	[16]	[17]

their structural simplicity, and their shape versatility, variable geometry trusses have a myriad of potential applications including robot arms [6], hyper-redundant manipulators [7], flight simulators [8], payload vibration reduction systems [9], tools to manipulate large payloads [10], morphing wings [11], space devices [12] or civil engineering structures [13].

The design of novel variable geometry trusses rely on having a complete kinematics characterization of their constituent cells. The larger the set of cells whose position analysis is solved, *i.e.*, the set of cells whose assembly modes can be obtained solely from their edge lengths, the larger the options available to the designer. The motion capabilities and the ways to actuate a cell increase with its number of bars, but this also complicates its position analysis.

It was shown in [5] that, under mild conditions, the only feasible cells in variable geometry trusses are topologically equivalent to bipyramids, *i.e*, polyhedral resulting from joining two n-gonal pyramids at their bases (see Table 1). Note, thought, that unit cells must not be decomposable in combinations of simpler unit cells. Thus, the bipyramid of order 1 in Table 1, *i.e.*, the hexahedron, cannot be properly considered a unit cell since it can be separated into two tetrahedra. We include it in the discussion because it will be the basic element used to derive the univariate closure polynomial of higher order bipyramids.

The position analysis of bipyramids can be addressed using numerical techniques such as those based on continuation [18, 19], or interval analysis [20, 21]. However, closed-form solutions in the form of univariate closure polynomials are, in general, preferable because they offer more information on the problem, such as an upper bound on the number of assembly modes, or the possible structure of the singularity set. Closed-form solutions are typically obtained using elimination techniques [22] where a univariate closure polynomial is obtained applying algebraic manipulations, sometimes driven by intuition, on a set of equations resulting from an algebraic formulation of the problem. Thus, the first step to solve the problem is to obtain a good formulation.

A straightforward formulation of the position analysis of bipyramids results from assigning coordinates to three vertices defining a face, and leaving all other vertices' coordinates as variables. These variable coordinates are obviously constrained by the edge lengths which translate into quadratic equations relating them. The result is a large sparse system of equations from which it is not easy to obtain a univariate resultant without introducing extraneous factors [23]. As an alternative, ad-hoc approaches for particular cells permit obtaining more compact formulations by introducing position and angular variables. In this case, the former variables are usually put in terms of the latter, and the tangent half-angle substitution is used to obtain a reduced system of algebraic expressions.

Unfortunately, even with compact formulations, the elimination process gets rapidly involved as the order of the bipyramid increases. Therefore, up to now, only closed-form solutions for the octahedral [15, 3] and the decahedral bipyramids [16] are available. For these two cases, it has been shown that the univariate closure polynomials are of degree 8 and 24, respectively. The following bipyramid in complexity, the dodecahedral bipyramid of order 4, has only been analyzed using numerical techniques [17].

In this paper, the position analysis problem is solved for bipyramids up to order 7 using a coordinate-free distance-based formulation and a kinematic inversion for fans of tetrahedra (a sequence of tetrahedra sharing a common edge). The presented method starts by finding the distance between the end-vertices of a fan of k tetrahedra, *i.e.*, its vertices of degree 3. Then, a univariate closure polynomial for a $(k-1)$-order bipyramid is obtained using a kinematic inversion. The procedure is surprisingly simple, and, from our point of view, its ability to obtain a closure polynomial without using any kind of variable elimination is remarkable.

This paper is organized as follows. Section 2 shows how to obtain the distance between the end-vertices of a tetrahedral fan with an arbitrary number of tetrahedra. Then, Section 3 uses this result to derive a univariate closure polynomials for bipyramids using a kinematic inversion. Section 4 solves the position analysis of bipyramids up to order 7, and, finally, Section 5 summarizes and discusses the main contributions of the proposed approach.

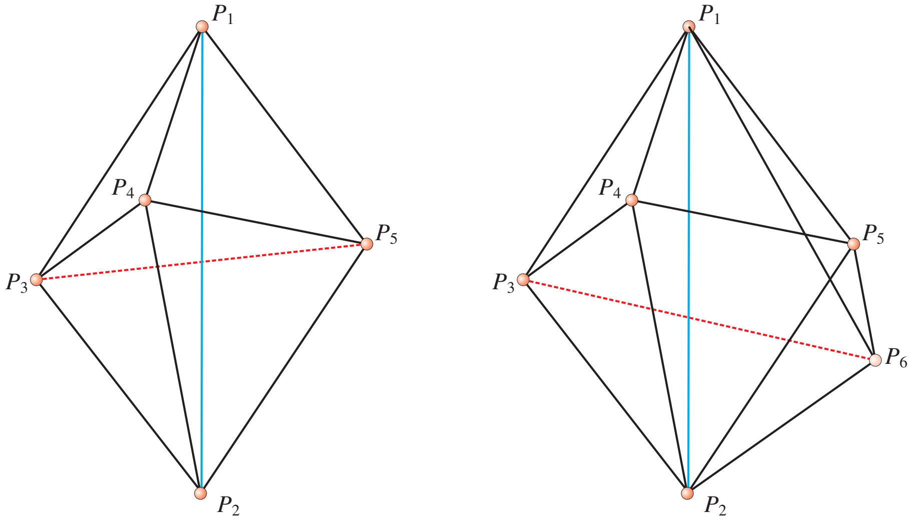


Figure 1: In the fan of two tetrahedra ($P_1P_2P_3P_4$ and $P_1P_2P_4P_5$) depicted on the left, the squared distance between its end-vertices, $s_{3,5}$, can be expressed as a 2-valued function of the edge lengths in the fan. By chaining this operation twice, the squared distance between end-vertices of the fan of three tetrahedra ($P_1P_2P_3P_4$, $P_1P_2P_4P_5$ and $P_1P_2P_5P_6$) on the right can be obtained as a 4-valued function of all involved edge distances. If an edge connecting the end-vertices of the fan (represented in dotted red) is added and the common edge (represented in solid cyan) is removed, the result is a bipyramid.

2. The Distance Geometry of fans of tetrahedra

The valid distances between a set of points can be characterized using the theory of Cayley-Menger determinants [24, 25]. The Cayley-Menger bi-determinant of two sets of points, $P_{i_1}, \ldots, P_{i_n}$ and $P_{j_1}, \ldots, P_{j_n}$, is defined[1] as

$$D(i_1, \ldots, i_n; j_1, \ldots, j_n) = \begin{vmatrix} 0 & 1 & \ldots & 1 \\ 1 & s_{i_1,j_1} & \ldots & s_{i_1,j_n} \\ 1 & \vdots & \ddots & \vdots \\ 1 & s_{i_n,j_1} & \ldots & s_{i_n,j_n} \end{vmatrix}, \tag{1}$$

where $s_{k,l}$ stands for the squared distance between P_k and P_l. If the two sets of points are the same, then $D(i_1, \ldots, i_n) = D(i_1, \ldots, i_n; i_1, \ldots, i_n)$ is called the Cayley-Menger determinant of the involved set of points.

The Cayley-Menger determinant is proportional to the squared volume of the simplex spanned by points $P_{i_1}, \ldots, P_{i_n}$. Thus, in three dimensions, any Cayley-Menger determinant involving more than 4 points necessarily vanishes. In particular, for the fan of two tetrahedra depicted in Fig. 1-left, we have that

$$D(1, 2, 3, 4, 5) = 0, \tag{2}$$

[1] An alternative common definition (see for instance [24]) includes a constant factor, which is dropped here to simplify the formulation.

which yields a quadratic expression in any of the involved distances. Following the procedure detailed in [26], this expression permits expressing one of the squared distances, say $s_{3,5}$, as a function of the other distances as

$$s_{3,5} = -\frac{1}{D(1,2,4)} \left(D(1,2,3,4;1,2,4,5)\Big|_{s_{3,5}=0} \pm \sqrt{D(1,2,3,4)\, D(1,2,4,5)} \right), \tag{3}$$

where $D(1,2,3,4;1,2,4,5)|_{s_{3,5}=0}$ denotes the Cayley-Menger bideterminant $D(1,2,3,4;1,2,4,5)$ with $s_{3,5}$ set to 0.

To lighten the notation, and assuming that only $s_{1,2}$ and $s_{3,4}$ are left as variables, (3) will be written as

$$s_{3,5} = \Psi_{3,1,2,4,5}(s_{1,2}, s_{3,4}). \tag{4}$$

Then, if we have the fan of three tetrahedra depicted in Fig. 1-right, we conclude that

$$\begin{aligned} s_{3,6} &= \Psi_{3,1,2,5,6}(s_{1,2}, s_{3,5}) \\ &= \Psi_{3,1,2,5,6}(s_{1,2}, \Psi_{3,1,2,4,5}(s_{1,2}, s_{3,4})). \end{aligned} \tag{5}$$

By iterating this reasoning, if we have a fan of tetrahedra with m vertices, $m > 4$, the distance between its end-vertices can be expressed as the following $2^{(m-4)}$-valued function:

$$\begin{aligned} s_{3,m} &= \Psi_{3,1,2,m-1,m}(s_{1,2}, s_{3,m-1}) \\ &= \Psi_{3,1,2,m-1,m}(s_{1,2}, \Psi_{3,1,2,m-2,m-1}(s_{1,2}, s_{3,m-2})) \\ &\quad \vdots \\ &= \Psi_{3,1,2,m-1,m}(s_{1,2}, \Psi_{3,1,2,m-2,m-1}(s_{1,2}, \cdots, \Psi_{3,1,2,5,6}(s_{1,2}, \Psi_{3,1,2,4,5}(s_{1,2}, s_{3,4}))\cdots)), \end{aligned} \tag{6}$$

Next, using a kinematic inversion, this formula is used to obtain closure polynomials for bipyramids.

3. Deriving closure polynomials for bipyramids

In a bipyramid, $s_{3,4}$ and $s_{3,m}$ in (6) are known and we want to obtain the values of $s_{1,2}$ compatible with these fixed distances. This operation can be seen as a kinematic inversion. Kinematic inversion is the process of fixing different links in a mechanism (or assuming that any link other than the rigid links is fixed). It is a good method of generating some new mechanisms and it is very often used in mechanism analysis [27]. To perform this inversion, the first step is to clear the square roots in (6). As radicals will appear nested, clearing them will consist in an iterative process starting from the outer one. At each step of this process, the expressions involving a radical will have the general form

$$\alpha_0 + \alpha_1 \sqrt{r} + \alpha_2 \left(\sqrt{r}\right)^2 + \alpha_3 \left(\sqrt{r}\right)^3 + \cdots = 0, \tag{7}$$

where r and α_i are symbolic expressions. This equation can be rewritten as

$$(\alpha_0 + \alpha_2 r + \alpha_4 r^2 + \ldots) + \sqrt{r}\,(\alpha_1 + \alpha_3 r + \alpha_5 r^2 + \ldots) = 0, \tag{8}$$

which can be unfolded into two equations, one for each sign of $\sqrt{r}$. Since we are interested in the roots of both equations, we obtain their product, which can be written as

$$(\alpha_0 + \alpha_2 r + \alpha_4 r^2 + \ldots)^2 - r\,(\alpha_1 + \alpha_3 r + \alpha_5 r^2 + \ldots)^2 = 0, \tag{9}$$

which does not include the radical any more.

While clearing radicals as explained above introduces no extraneous roots, one cannot expect to obtain the minimal degree polynomial due to the presence of singularities in the formulation. Indeed, each function $\Psi_{3,1,2,i-1,i}$, $i = 5, \ldots, m$, in (6), introduces a dividing term of the form $D(1, 2, i-1)$, which vanishes when P_1, P_2, and P_{i-1} are aligned. If this happens, the fan of tetrahedra can be divided in two parts which can freely rotate about the axis defined by P_1P_2. These degenerated cases have to be considered separately from the general analysis. Assuming that the fan of tetrahedra does not degenerate, *i.e.*, that $D(1, 2, i-1) \neq 0$ for $i = 5, \ldots, m$, these factors can be removed from the final expression. This can be simply performed by iteratively dividing the expression obtained after clearing radicals by them until the remainder is not null. This finally leads to the sought-after minimum degree univariate closure polynomial of the bipyramid of order $N = m - 4$ in the squared distance between its apices.

Observe that, given a valid configuration, its mirror projection defines another valid configuration because they both are defined by the same inter-point distances. Thus, each real solution of the obtained univariate polynomial defines at least two valid spatial configurations for the corresponding bipyramid. All valid configurations can be computed by assigning a valid set of coordinates to a reference triangle, say P_1, P_3 and P_4, and then, using trilateration, we can obtain 2 sets of possible coordinates for P_2, 4 for P_5, 8 for P_6, etc. Therefore, we will obtain 2^N sets of possible coordinates, either real or imaginary. Only the real solutions yield feasible configurations.

4. Results

Next, we apply the presented method to solve the position analysis of bipyramids. Irrespective of the actually actuated edges in each application, for a given instance of the position analysis problem, segments $\overline{P_1P_i}$, $\overline{P_2P_i}$, $\overline{P_jP_{j+1}}$, and $\overline{P_3P_m}$, with $2 < i \leq m$, and $2 < j < m$, have fixed length. In the examples analyzed next, such lengths are randomly generated. To this end, random coordinates are assigned to the vertices of the bipyramids and the required distances between them are then used as input to the position analysis. This procedure as well as the method presented in this paper has been implemented in Maple 15 and run on an Intel Core 7 at 3Ghz, with 12 Gb of RAM. This implementation can be found as multimedia material attached to this paper.

If we start with the position analysis of a bipyramid of order one, then (6) reduces to (4), that is,

$$s_{3,5} = \Psi_{3,1,2,4,5}(s_{1,2}, s_{3,4}). \tag{10}$$

Then, since in this case $s_{3,4}$ is known, the above equation, using (3), can be rewritten as

$$s_{3,5} = \frac{\Phi_1(s_{1,2}) \pm \sqrt{\Phi_2(s_{1,2})}}{\Phi_3(s_{1,2})}, \tag{11}$$

with $\Phi_1(s_{1,2})$, $\Phi_2(s_{1,2})$, and $\Phi_3(s_{1,2})$ symbolic expressions in $s_{1,2}$. This equation can be arranged as

$$\Phi_1(s_{1,2}) - s_{3,5}\,\Phi_3(s_{1,2}) \pm \sqrt{\Phi_2(s_{1,2})} = 0, \tag{12}$$

which, after clearing radicals, yields

$$\Phi_2(s_{1,2}) - (\Phi_1(s_{1,2}) - s_{3,5}\,\Phi_3(s_{1,2}))^2 = 0. \tag{13}$$

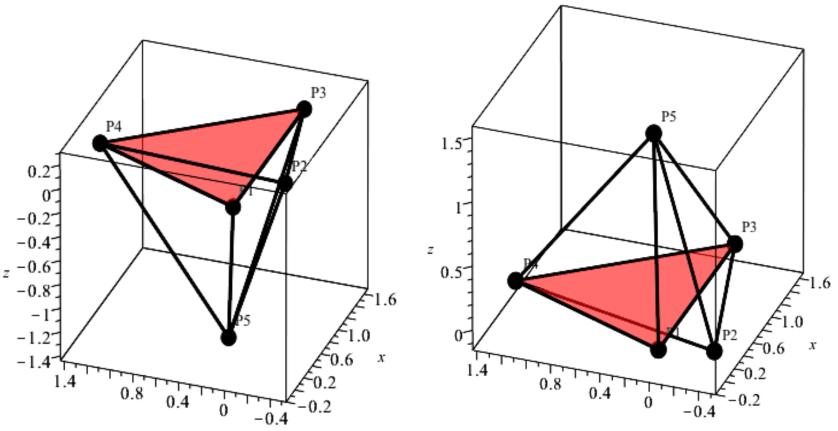


Figure 2: Two valid configurations resulting from the position analysis of the analyzed bipyramid of order 1. The other two are mirror configurations with respect to the plane defined by the reference triangle in red.

Using the random data generated as explained above, we have that

$$\begin{aligned}
\Phi_1(s_{1,2}) &= s_{1,2}^2 - 6.9\, s_{1,2} + 0.34,\\
\Phi_2(s_{1,2}) &= 0.99\, s_{1,2}^4 - 13.92\, s_{1,2}^3 + 50.35\, s_{1,2}^2 - 12.17\, s_{1,2} + 0.59,\\
\Phi_3(s_{1,2}) &= 0.15\, s_{1,2}^2 - 1.69\, s_{1,2} - 0.10,\\
s_{3,5} &= 3.26.
\end{aligned}$$

Then, substituting these expressions in (13) yields

$$0.76\, s_{1,2}^4 - 12.57\, s_{1,2}^3 + 48.48\, s_{1,2}^2 - 12.69\, s_{1,2} + 0.59 = 0.$$

Finally, after dividing the above right hand side polynomial by the singularity factor

$$D(1,2,4) = s_{1,2}^2 - 10.64\, s_{1,2} + 0.64, \tag{14}$$

we obtain the minimum degree closure polynomial

$$s_{1,2}^2 - 5.75\, s_{1,2} + 1.20. \tag{15}$$

As explained above, each root of this polynomial leads to two possible configurations of the bipyramid. In this case both roots yield real configurations. Figure 2 shows two of them. The other two are symmetric with respect to plane defined by P_1, P_3, and P_4, used as a reference triangle.

The coefficients of the closure polynomial are computed in exact rational arithmetics and, thus, the accuracy on the solutions depends uniquely on the method used to determine the roots of this polynomial.

To verify the results, let us define the error function

$$E(s_{1,2}) = \frac{\Phi_1(s_{1,2}) \pm \sqrt{\Phi_2(s_{1,2})}}{\Phi_3(s_{1,2})} - s_{3,5}. \tag{16}$$

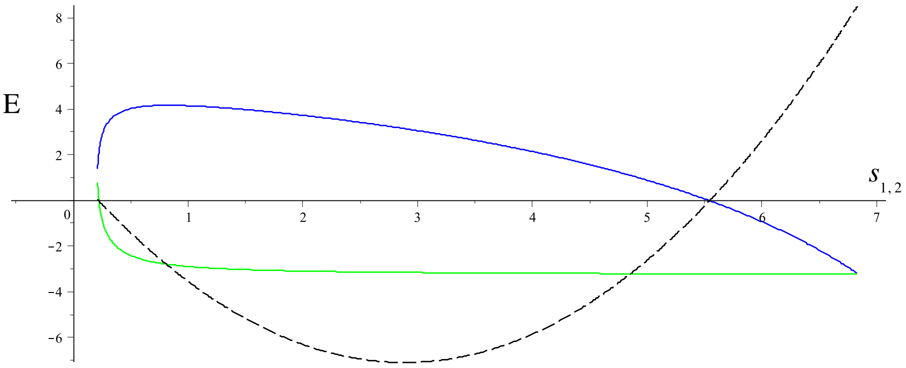


Figure 3: Plot of the closure polynomial (in dashed black), and the two values of the error function (in blue and green). Observe how their roots coincide.

N	t_e	t_r	t_f	d_c	t_s	n_r	t_c	n_c	t_b	t_k
1	0.12	0.00	0.00	2	0.02	2	0.02	4	98	0.05
2	0.13	0.00	0.00	8	0.05	6	0.02	4	1835	1.16
3	0.14	0.00	0.00	24	0.01	16	0.16	16	-	7.51
4	0.38	0.03	0.04	64	0.07	36	0.47	24	-	38.55
5	0.61	1.32	0.83	160	0.85	102	2.73	68	-	152.34
6	0.92	36.11	17.14	384	15.49	224	9.05	112	-	421.16
7	1.57	953.25	327.05	896	371.19	476	23.39	140	-	961.51

Table 2: Results obtained for the position analysis of bipyramids up to order 7. All times are given in seconds. Execution times over 2000 seconds are not given. See text for details.

The roots of this 2-valued function must coincide with those of (15). Figure 3 shows the plot of (15) (in dashed black) and that of (16) (in blue and green). Roots indeed coincide.

Alternatively to the process described above, one can use state-of-the-art numerical solvers such as Bertini [19] or the CuikSuite [28] to directly obtain the sets of valid coordinates. The former is a continuation-based solver and the latter is based on linear relaxations. Both solvers are applied on the standard formulation of the problem, based on point coordinates, as detailed in the Introduction. As expected, both solvers obtain the same four sets of coordinates as those obtained using the presented method.

Now, we can proceed using the same sequence of operations as above to solve the position analysis of bipyramids up to order 7. Table 2 summarizes the results. For each order of the bipyramid, N, the following data are given:

t_e: time taken to obtain the symbolic closure expression;

t_r: time taken to clear the square roots;

t_f: time taken to remove the singular factors;

d_c: the degree of the closure polynomial;

t_s: time taken to determine the roots of the closure polynomial;

n_r: number of real solutions;

t_c: time taken to generate coordinates from the roots;

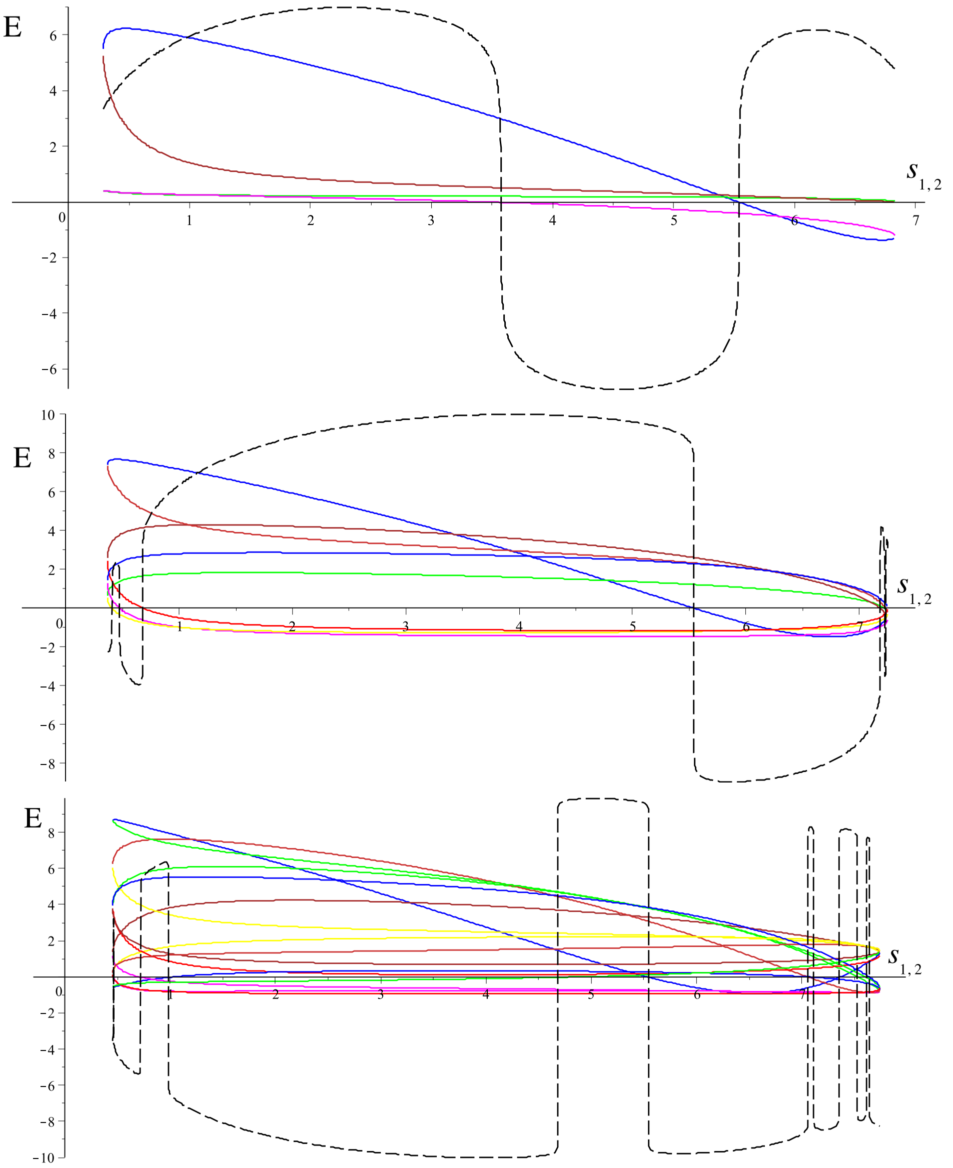


Figure 4: Plots of the 2^N-valued error functions (in different colors) and the univariate closure polynomial (in black dashed line) for bipyramids of order N=2 (top), 3 (middle), and 4 (bottom). The plots of the closure polynomial have been non-linearly scaled to fit them in the ranges of the error functions (as only their roots are relevant here). Similar plots have been obtained for bipyramids with order up to 7, but the details are obviously harder to visualize.

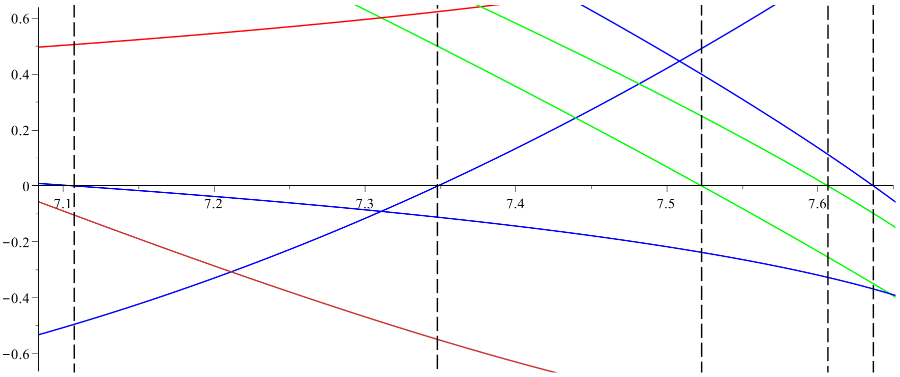


Figure 5: Zoom-in on the right part of the plot in Fig. 4-bottom showing the coincidence between the roots of the 2^4-valued error function (represented by lines in different colors) and those computed using the obtained closure polynomial (represented by the black dashed line).

n_c: number of valid coordinate sets;
t_b: time taken by Bertini to obtain the valid coordinate sets;
t_k: time taken by CuikSuite to obtain the valid coordinate sets.

Figure 4 shows the plots of the error functions for bipyramids of order 2, 3, and 4. Figure 5 shows a zoom-in on the right extreme of the last plot, where it can be verified the coincidence between the roots of the closure polynomials and those of the corresponding error functions. Finally, Figs. 6 and 7 show some valid configurations for the solved position analyses.

Despite Bertini and Cuik are compiled and highly optimized solvers written in C, the symbolic approach introduced in this paper is significantly more efficient. In the case of Bertini, the number of paths to track using continuation grows with 2^{m+3} thus becoming impractical for this problem. Bertini is actually well-suited for problems involving a moderate number of variables and equations, even if they are of high degree. The CuikSuite, on the contrary, is specially tailored for large problems with equations of low degree. While results show that the presented symbolic approach becomes impractical for the position analysis of bipyramids of order higher than 7, Cuik remains as the only numerical alternative for solving such problems. The limiting factor for the presented approach is memory usage due to the symbolic manipulation of high-order polynomials in rational arithmetics. This is specially relevant when clearing radicals.

5. Conclusions

Variable geometry trusses can be seen as composed of bar-and-joint cells that have the topology of bipyramids. In the previous literature on this kind of trusses, the available closed-form solutions to the position analysis of these constituent cells were obtained on a case-by-case basis and limited to those having up to 7 vertices. In this paper, we have presented a unifying procedure that generates closed-form solutions in general, independently of the number of vertices. Most surprisingly, this result is obtained without relying on variable eliminations. As an example, we have solved a case that leads to a closure polynomial of degree 896. Up to our knowledge, no other position analysis problem leading to such a high-order closure polynomial has been previously solved in closed-form.

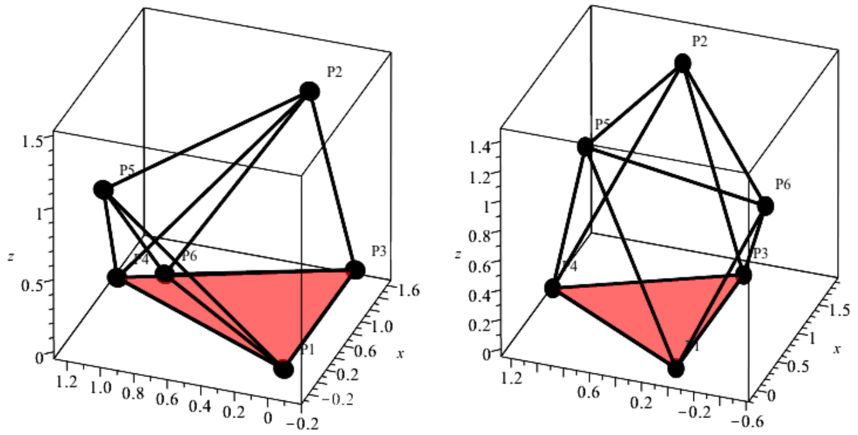


Figure 6: Two valid configurations for the analyzed bipyramid of order 2. The two mirror configuration with respect to the plane defined by the reference triangle in red are also valid configurations not represented here. The problem solved here is equivalent to that of solving the forward kinematics of the octahedral manipulator.

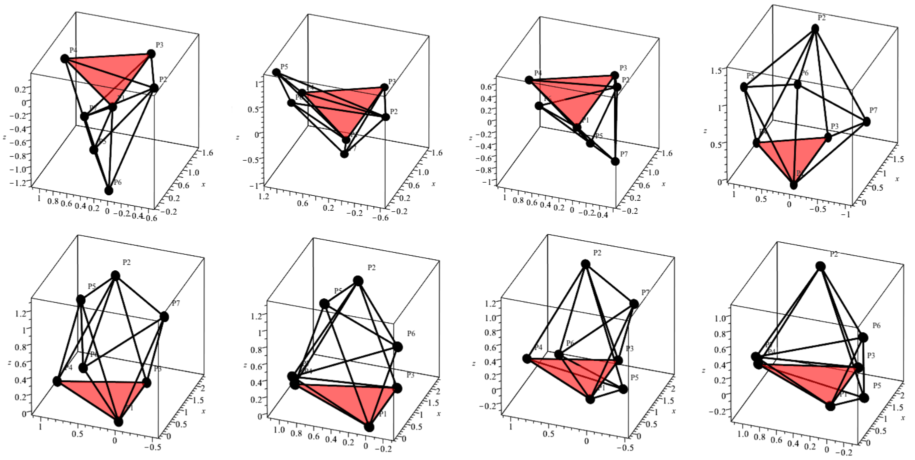


Figure 7: Eight of the sixteen valid configurations for the analyzed bipyramid of order 3. The other eight configurations result from obtaining the mirror symmetric ones to these with respect to the plane defined by the reference triangle in red.

The presented procedure can obviously be applied to solve the position analysis of rigid graphs other than those with the topology of bipyramids. For instance, in [29] it is applied to solve the forward kinematics of some parallel robots. Identifying the family of graphs to which this technique can be applied is certainly a point that deserves to be investigated.

Acknowledgments

This work has been partially supported by the Spanish Ministry of Economy and Competitiveness under project DPI2014-57220-C2-2-P.

References

[1] P. Bosscher, I. Ebert-Uphoff, A novel mechanism for implementing multiple collocated spherical joints, in: IEEE International Conference on Robotics and Automation, Vol. 1, 2003, pp. 336–341.

[2] J. Borràs, F. Thomas, C. Torras, Singularity-invariant leg rearrangements in Stewart-Gough platforms, in: Advances in Robot Kinematics, 2010, pp. 421–428.

[3] N. Rojas, J. Borràs, F. Thomas, The octahedral manipulator revisited, in: IEEE International Conference on Robotics and Automation, 2012, pp. 2293–2298.

[4] K. Miura, Variable geometry truss concept, Tech. Rep. 614, The Institute of Space and Astronautical Science (1984).

[5] V. Arun, C. F. Reinholtz, L. T. Watson, Enumeration and analysis of variable geometry truss manipulators, Tech. Rep. TR-90-10, Virginia Tech (1990).

[6] P. C. Hughes, W. C. Sincarsin, K. A. Carroll, Trussarm – A variable-geometry-truss manipulator, Journal of Intelligent Material Systems and Structures 2 (2) (1991) 148–160.

[7] G. S. Chirikjian, J. W. Burdick, A hyper-redundant manipulator, IEEE Robotics and Automation Magazine 1 (4) (1994) 22–29.

[8] C. Sultan, M. Corless, R. E. Skelton, Tensegrity flight simulator, Journal of Guidance, Control, and Dynamics 26 (6) (2000) 1055–1064.

[9] P. Dadone, W. Lacarbonara, A. H. Nayfeh, H. F. Vanlandingham, Payload pendulation reduction using a variable-geometry-truss architecture with LQR and fuzzy controls, Journal of Vibration and Control 9 (7) (2003) 805–837.

[10] R. S. Stoughton, J. C. Tucker, A variable geometry truss manipulator for positioning large payloads, in: American Nuclear Society meeting on robotics and remote systems, 1995.

[11] A. D. Finistauri, X. Fengfeng, Type synthesis and kinematics of a modular variable geometry truss mechanism for aircraft wing morphing, in: International Conference on Reconfigurable Mechanisms and Robots, 2009, pp. 478–485.

[12] K. Miura, H. Furuya, K. Suzuki, Variable geometry truss and its application to deployable truss and space crane arm, Acta Astronautica 12 (7) (1985) 599–607.

[13] K. Kurita, F. Inoue, N. Furuya, T. Shiokawa, M. Natori, Development of adaptive roof structure by variable geometry truss, in: International Symposium on Automation and Robotics in Construction, 2001, pp. 1–6.

[14] N. Rojas, F. Thomas, The closure condition of the double banana and its application to robot position analysis, in: International Conference on Robotics and Automation, 2013, pp. 4641–4646.

[15] K. H. Hunt, P. R. McAree, The octahedral manipulator: Geometry and mobility, The International Journal of Robotics Research 17 (8) (1998) 868–885.

[16] Y. Jin, F. Hai-rong, Forward displacement analysis of the decahedral variable geometry truss manipulator, Robotics and Autonomous Systems 15 (3) (1995) 173–178.

[17] Y. Luo, B. Zeng, Z. He, Hyper-chaotic mathematical programming method and its application to dodecahedron variable geometry truss manipulator, in: International Symposium on Neural Networks, 2009, pp. 1033–1040.

[18] E. L. Allgower, K. Georg, Introduction to Numerical Continuation Methods, Vol. 45, SIAM Classics in Applied Mathematics, 2003.

[19] D. J. Bates, J. D. Hauenstein, A. J. Sommese, C. W. Wampler, Numerically Solving Polynomial Systems with Bertini, SIAM, 2013.

[20] R. E. Moore, Interval Analysis, Prentice-Hall, 1966.

[21] J. M. Porta, L. Ros, F. Thomas, C. Torras, A branch-and-prune algorithm for solving systems of distance constraints, in: IEEE International Conference on Robotics and Automation, 2003, pp. 342–348.

[22] D. Cox, J. Little, D. O'Shea, An Introduction to Computational Algebraic Geometry and Commutative Algebra, 2nd Edition, Springer, 1997.

[23] T. Saxena, Efficient variable elimination using resultants, Ph.D. thesis, State University of New York at Albany (1997).

[24] T. Havel, Some examples of the use of distances as coordinates for Euclidean geometry, Journal of Symbolic Computation 11 (5-6) (1991) 579–593.

[25] F. Thomas, L. Ros, Revisiting trilateration for robot localization, IEEE Transactions on Robotics 21 (1) (2005) 93–101.

[26] J. M. Porta, L. Ros, F. Thomas, Inverse kinematics by distance matrix completion, in: International Workshop on Computational Kinematics, 2005.

[27] L.-W. Tsai, Mechanism Design: Enumeration of Kinematic Structures According to Function, CRC Press, 2000.

[28] J. M. Porta, L. Ros, O. Bohigas, M. Manubens, C. Rosales, L. Jaillet, The cuik suite: Analyzing the motion closed-chain multibody systems, IEEE Robotics and Automation Magazine 21 (3) (2014) 105–114.

[29] F. Thomas, J. M. Porta, Closure polynomials for strips of tetrahedra, in: Advances in Robot Kinematics, Springer Proceedings on Advanced Robotics, 2016, in press.