

Mark M. Plecnik¹

Biomimetic Millisystems Lab,
Department of Electrical Engineering
and Computer Sciences,
University of California,
Berkeley, CA 94720
e-mail: mplecnik@berkeley.edu

Ronald S. Fearing

Professor
Biomimetic Millisystems Lab,
Department of Electrical Engineering
and Computer Sciences,
University of California,
Berkeley, CA 94720
e-mail: ronf@eecs.berkeley.edu

Finding Only Finite Roots to Large Kinematic Synthesis Systems

In this work, a new method is introduced for solving large polynomial systems for the kinematic synthesis of linkages. The method is designed for solving systems with degrees beyond 100,000, which often are found to possess quantities of finite roots that are orders of magnitude smaller. Current root-finding methods for large polynomial systems discover both finite and infinite roots, although only finite roots have meaning for engineering purposes. Our method demonstrates how all infinite roots can be precluded in order to obtain substantial computational savings. Infinite roots are avoided by generating random linkage dimensions to construct startpoints and start systems for homotopy continuation paths. The method is benchmarked with a four-bar path synthesis problem.
[DOI: 10.1115/1.4035967]

1 Introduction

The kinematic synthesis of linkages often requires finding the finite isolated roots of large polynomial systems. The size of a polynomial system is measured by its number of roots, of which its degree provides a maximum. It is interesting that the kinematic design of relatively simple mechanical systems involves formulating polynomial systems with degrees beyond 100,000. For example, to find all of the four-bar linkages which can pass a coupler point through nine arbitrary points in the plane involves solving a polynomial system with degree 286,720 [1]. Furthermore, more complex mechanical systems seem to result in exponential growth of the degree of their design equations [2,3].

This growth arises from additional design parameters that increase the number of dimensions in a nonlinear design space. Finding complete solution sets to the polynomial systems which comprise kinematic design equations is a major challenge which tends to demand large-scale computations. Many methods have been devised, which avoid computing complete solution sets [4], however, the advantage of obtaining complete sets is to provide a full survey of design options that might not be found through methods based on optimization or differential evolution. The state-of-the-art for solving large polynomial systems is polynomial homotopy continuation. Homotopy algorithms compute all the roots to general polynomial systems in time proportional to the degree of the system. These methods are quite powerful but still have practical limits based on available computing resources.

Although kinematic polynomial systems tend to have large degrees, they also tend to have sparse monomial structures that indicate the number of finite roots they possess is much smaller than their degree. Returning to the four-bar example, it has been shown in Ref. [1] that 8652 of 286,720 roots are finite and represent linkage designs. The polynomial degree totals the number of finite roots, infinite roots, and their multiplicities. For engineering purposes, only finite roots are sought. Observations of the monomial structure of a system can often be made in order to intelligently multihomogenize a system and remove many roots at infinity leading to a new *multihomogeneous root count*. For smaller systems, this strategy can eliminate all the roots at infinity, but for larger systems it is not always clear how the multihomogenization should proceed. Eliminating roots at infinity is important when applying homotopy algorithms because each root requires a path tracking computation. Counting the percentage of infinite

roots for the four-bar example, it is estimated that 97% of the computational effort is dedicated to discovering roots at infinity, which are filtered out in the end.

In this work, we introduce a new method built on homotopy continuation termed *Finite Root Generation* (FRG) that obtains all or most of the finite roots of kinematic polynomial systems while avoiding all computations of roots at infinity. Traditionally, homotopy proceeds with a single start system and multiple startpoints, whereas FRG uses multiple start systems each with one startpoint, which are constructed from random linkage dimensions so that each startpoint is guaranteed to track to a finite root. Since startpoints are randomly generated, the possibility exists for tracking to a nonsingular finite root multiple times. This duplication rate is modeled by the *coupon collector problem* from probability theory and can be used to estimate the number of finite roots without actually computing all of them.

We apply FRG to a numerically general target system that is suitable for constructing parameter homotopies such that an FRG solution set only needs to be computed once for a particular family of polynomial systems. Parameter homotopies provide a means for efficiently solving all subsequent systems belonging to that family. Applying FRG to the benchmark four-bar path generator problem found over 95% of the roots in 26,658 homotopy paths and 99.988% of the roots in 87,549 paths. This is compared to 286,720 for the best known multihomogeneous homotopy [1] and 152,224 for the best known regeneration homotopy [5]. The benchmark slightly underperformed expectations calculated from modeling roots as equally probable coupons.

2 Literature Review

Continuation methods were pioneered by Roth and Freudentstein [6], who developed the *bootstrap method* for solving the nonlinear path synthesis equations of a geared five-bar. They were motivated to provide an alternative to the Newton–Raphson method which required a good initial approximation in order to converge. In their work, the authors indeed constructed continuation startpoints from arbitrary starting mechanisms. Since then, the field of numerical algebraic geometry has flourished resulting in sophisticated homotopy continuation algorithms [5,7–10] that apply to general polynomial systems where startpoints are constructed by combinatoric procedures that are advantageously blind to the physical systems represented by the equations. In this work, we return to the schema of constructing startpoints by arbitrary mechanisms, but instead of finding one or two mechanism solutions, we aim to find complete root sets of numerically general

¹Corresponding author.

Manuscript received October 17, 2016; final manuscript received January 25, 2017; published online March 9, 2017. Assoc. Editor: Venkat Krovi.

synthesis systems. These general root sets are suitable for constructing parameter homotopies to efficiently solve later synthesis systems of the same family [11], a concept that appeared 26 years after Roth and Freudenstein [6].

Other solution techniques used to solve polynomial equations include resultant elimination methods, Gröbner bases, and interval analysis. Resultant elimination methods include modifications of Sylvester's dialytic elimination method [12] and involve procedures to transform root finding into a generalized eigenvalue problem. Su and McCarthy [13] showed how to use this method to solve a synthesis problem with 64 roots. Masouleh et al. [14] used Gröbner bases combined with resultants to compute 220 roots for the forward kinematics of a spatial parallel manipulator. Gröbner bases transform multivariate root finding into root finding for a sequence of univariate polynomials. Interval analysis takes advantage of interval arithmetic to find roots within a range of the solution space. Lee et al. [15] provided an example of interval analysis applied to the synthesis of an RRR spatial serial chain.

There have been modifications of homotopy methods to eliminate unwanted solutions. Tari et al. [16] appended additional equations to synthesis systems to eliminate degenerate solutions, but their method does not improve computation times. Tsai and Lu [17] devised a procedure for nine-point path synthesis of a four-bar where startpoints were constructed from five-point solutions without attempting to obtain the complete solution set. Perhaps today's most powerful method is called regeneration. Regeneration can greatly reduce the number of homotopy paths to track by solving a system "equation by equation" and discarding singular and infinite roots along the way [5]. However, regeneration still requires many roots at infinity to be tracked. Plecnik and McCarthy [2] used regeneration to approximate a solution set of 1.5×10^6 roots.

In this work, we introduce a method which constructs startpoints and start systems from randomly generated mechanisms in order to avoid tracking homotopy paths to infinite roots. We benchmark FRG on the four-bar path synthesis problem and find that the rate at which finite roots are collected is predicted by the coupon collector problem of probability theory, providing a means for estimating the size of the root set being discovered. Finally, we use the parameter homotopy method to apply the obtained root set to an example linkage design problem.

3 Description of Finite Root Generation

The FRG method generates startpoints and start systems that track to the finite roots of a target system using polynomial homotopy continuation. In order to ensure this is the case, a start system must possess a monomial structure that is no more or no less general than the target system, and a startpoint must be a finite root to that start system. These features are ensured for a single startpoint/start system pair by constructing the startpoint from a randomly generated linkage and constructing the start system from the motion of that linkage. A parameter homotopy then tracks that startpoint to a single finite root of the target system.

In order to find another finite root to the target system, a new random linkage can be generated so that another startpoint/start system pair is constructed and tracked to (hopefully) another finite root of the target system. If there are N finite roots of the target system to be discovered, and there is equal probability of happening upon any root, then the probability of the second iteration happening upon a different finite root from the first is $(N-1)/N$, which for large N are good odds. If the root of the second iteration duplicates the first, then it is discarded, or if it is unique, then it is stored along with the root from the first iteration. The goal is to obtain all or most of the finite roots of the target system. Iterations (also called trials) continue in this manner as illustrated in Fig. 1 until a sufficient number of roots have been collected. As more and more roots are collected, the possibility of finding another unique root decreases, indicating a situation of diminishing returns. By tracking the rate of success of finding new

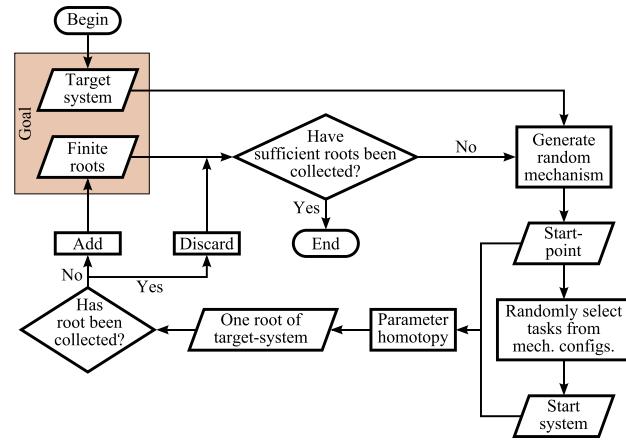


Fig. 1 Flowchart of the Finite Root Generation method

roots between iterations, the size of the total root set can be estimated.

3.1 Root Collection. The decreasing probability of finding new roots is modeled by the coupon collector problem from probability theory [18]. That is, given a set of N unique coupons, how many trials T_N of picking random single coupons are expected in order to obtain all the coupons. Coupons are sampled with replacement and we assume all the coupons have equal probability of being picked. The problem statement is slightly modified to ask how many trials T_n are expected to obtain n coupons where $n \leq N$. In our case, T_n is the number of homotopy paths and n is the number of finite roots we wish to obtain.

Once $n-1$ roots have been collected, the probability p_n of collecting the n th root is

$$p_n = \frac{N - (n-1)}{N} \quad (1)$$

Let X_n denote the expected number of trials to take place in the interval between finding $n-1$ roots and n roots. Because X_n is a random variable with a geometric distribution, it is computed as

$$X_n = \frac{1}{p_n} \quad (2)$$

That means, it is expected that N trials will be dedicated solely to finding the final root. By substituting Eq. (1) into Eq. (2) and summing X , the expected number of trials to obtain the n th finite root can be expressed as

$$T_n = \sum_{k=1}^n X_k = \sum_{k=1}^n \frac{N}{N - (k-1)} \quad (3)$$

The variance of T_n is computed as

$$\text{Var}(T_n) = \sum_{k=1}^n \frac{N(k-1)}{(N - (k-1))^2} \quad (4)$$

Equation (3) provides an estimation of the expected number of trials to collect n roots when the total number N is known.

It is useful to approximate Eq. (3) with a form that excludes summations. To obtain this, first rewrite Eq. (3) as

$$T_n = N \left(\left(\sum_{k=1}^N \frac{1}{k} \right) - \left(\sum_{k=1}^{N-n} \frac{1}{k} \right) \right) \quad (5)$$

The resulting summations are harmonic numbers which can be approximated by removing the higher order terms from Euler's asymptotic expansion [19]

$$\sum_{k=1}^N \frac{1}{k} \approx \gamma + \ln N \quad (6)$$

where γ is the Euler–Mascheroni constant. Apply Eq. (6) to Eq. (5) and simplify to obtain

$$T_n \approx N \ln \left(\frac{N}{N-n} \right) \quad (7)$$

3.2 Estimation Procedure. For large polynomial systems of interest, N is not known but can be estimated by the ratio of success α of finding new roots defined as

$$\alpha = \frac{n}{T_n} \quad (8)$$

where n is the number of roots collected, and T_n is the total number of trials it took to collect those roots. As the probability of finding new roots decreases, α tends to decrease. The aim is to derive an expression for estimating the percentage of roots obtained by the success ratio α . First, replace T_n in Eq. (8) with its approximation from Eq. (7). Then, substitute $\hat{n} = n/N$ as the percentage of finite roots found. The result is

$$\alpha = -\frac{\hat{n}}{\ln(1-\hat{n})} \quad (9)$$

where N has been eliminated from the calculation. Inversion of Eq. (9) requires the principle branch of the Lambert function W and takes the form

$$\hat{n} = \alpha W \left(-\frac{1}{\alpha} e^{-\frac{1}{\alpha}} \right) + 1 \quad (10)$$

Equation (10) provides an estimation of the percentage of roots obtained \hat{n} from the current success ratio α during the FRG solution process. Note that this equation only provides an estimate of the number of finite roots based on the success ratio, unlike Bézout numbers which provide conclusive statements on the upper bound of finite roots. FRG estimates might indicate smaller root sets, but without certainty.

4 Benchmark

In order to benchmark the FRG method, it was applied to the nine-point path synthesis problem for four-bar linkages. The complete solution to this problem involves finding 8652 finite roots but the smallest known formulation of this system counts 286,720 roots (including infinite roots). Multihomogeneous homotopy has been applied to solve this problem by tracking 286,720 homotopy paths to discover the 8652 finite roots [1] (see Fig. 2). As an improvement, regeneration homotopy finds all the finite roots while only tracking 152,224 paths [5]. From Eq. (3), it is expected for FRG to find 95% of the roots in 25,922 paths and 100% in 83,430 paths (see Figs. 3 and 4). In other words, the expectation is for 31% of the computational effort to find 95% of the roots and 70% of the effort to find the final 5% of the roots. In this section, we formulate the synthesis equations and test the expectations.

4.1 Formulation of Four-Bar Path Synthesis. The objective of path generation is to find a four-bar linkage that can move a trace point attached to its coupler link through m points P_j , $j=0, \dots, m-1$. As shown in Fig. 5, the locations of the four-bar's ground pivots are represented by $A = A_x + A_y i$ and $B = B_x + B_y i$.

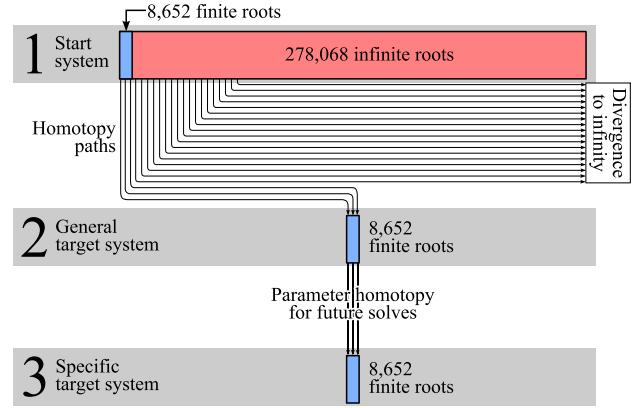


Fig. 2 Multihomogeneous homotopy applied to the four-bar benchmark problem tracks 97% of the paths to roots at infinity

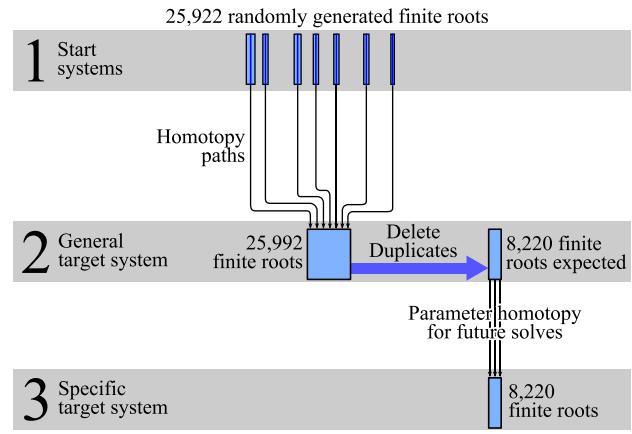


Fig. 3 The expected results of FRG found 95% of the roots in 91% less paths than multihomogeneous homotopy applied to the four-bar benchmark problem

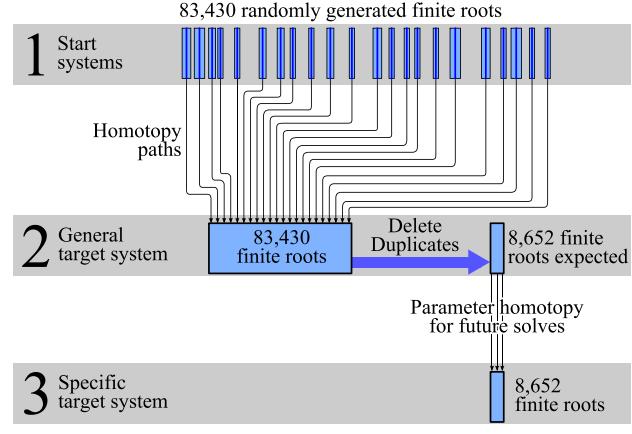


Fig. 4 The expected results of FRG found 100% of the roots in 71% less paths than multihomogeneous homotopy applied to the four-bar benchmark problem

The locations of the moving pivots in position $j=0$ are represented by $C = C_x + C_y i$ and $D = D_x + D_y i$. The rotations of links AC , BD , and CDP from position 0 to j are measured by ϕ_j , ψ_j , and θ_j , respectively. Using the exponential rotation operators

$$Q_j = e^{i\phi_j}, \quad S_j = e^{i\psi_j}, \quad T_j = e^{i\theta_j} \quad (11)$$

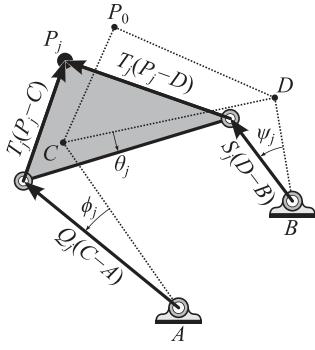


Fig. 5 A four-bar path generator displaced to position j

two loop equations are formulated for each position j

$$A + Q_j(C - A) + T_j(P_0 - C) = P_j, \quad (12)$$

$$B + S_j(D - B) + T_j(P_0 - D) = P_j, \quad j = 1, \dots, m - 1 \quad (13)$$

As well, the rotation operators necessarily satisfy

$$Q_j \bar{Q}_j = 1, \quad (14)$$

$$S_j \bar{S}_j = 1, \quad (15)$$

$$T_j \bar{T}_j = 1, \quad j = 1, \dots, m - 1 \quad (16)$$

where the overbar denotes the complex conjugate. Unknowns Q_j are eliminated by solving for them in Eq. (12) and substituting into Eq. (14). Similarly, S_j is eliminated by solving Eq. (13) and substituting into Eq. (15). These substitutions obtain

$$\begin{bmatrix} a\bar{b}_j & \bar{a}b_j \\ c\bar{d}_j & \bar{c}d_j \end{bmatrix} \begin{Bmatrix} T_j \\ \bar{T}_j \end{Bmatrix} = \begin{Bmatrix} f\bar{f} - a\bar{a} - b_j\bar{b}_j \\ g\bar{g} - c\bar{c} - d_j\bar{d}_j \end{Bmatrix} \quad (17)$$

where

$$\begin{aligned} a &= P_0 - C, & b_j &= A - P_j, & f &= C - A, \\ c &= P_0 - D, & d_j &= B - P_j, & g &= D - B, \end{aligned} \quad (18)$$

Finally, solving Eq. (17) for (T_j, \bar{T}_j) , substituting into Eq. (16), clearing the denominator, and factoring, we obtain the polynomial system \mathcal{T}

$$\mathcal{T}: \mathbf{a}^T \bar{\mathbf{b}}_j \mathbf{b}_j^T \bar{\mathbf{a}} - \mathbf{c}^T \bar{\mathbf{d}}_j \mathbf{d}_j^T \bar{\mathbf{c}} = 0, \quad j = 1, \dots, 8 \quad (19)$$

where

$$\begin{aligned} \mathbf{a} &= \begin{Bmatrix} a(g\bar{g} - c\bar{c}) \\ c(f\bar{f} - a\bar{a}) \\ a \\ c \end{Bmatrix}, & \mathbf{b}_j &= \begin{Bmatrix} b_j \\ -d_j \\ -b_j d_j \bar{d}_j \\ b_j \bar{b}_j d_j \end{Bmatrix}, \\ \mathbf{c} &= \begin{Bmatrix} a\bar{c} \\ \bar{a}c \end{Bmatrix}, & \mathbf{d}_j &= \begin{Bmatrix} b_j \bar{d}_j \\ -\bar{b}_j d_j \end{Bmatrix}, \end{aligned} \quad (20)$$

Equation (19) contains the eight unknowns $\{A, B, C, D, \bar{A}, \bar{B}, \bar{C}, \bar{D}\}$ and represents a square system of eight equations for the case $m = 9$.

Our formulation is similar to those presented in Refs. [1] and [5], which provide additional information on the maximum number of roots in order to determine the number of homotopy paths to be tracked. Advantageously, FRG paths are tracked without the knowledge of root ceilings, similar to the regeneration method. For the sake of comparison, sorting the variables of \mathcal{T} into groups $\{A, \bar{A}\}$, $\{B, \bar{B}\}$, $\{C, \bar{C}\}$, and $\{D, \bar{D}\}$ finds a four-homogeneous Bézout number of 645,120, which was the same calculated in Ref. [5]. In Ref. [1], intermediate variables were introduced to further reduce the multihomogeneous Bézout number to 286,720.

4.2 Methods. FRG begins by creating a target system for which it is intended to find all the roots. The target system is constructed to be numerically general rather than represent a specific synthesis task. The reason for this is that the target system is intended to be solved once where thereafter the target system and its roots can be used to construct parameter homotopies that efficiently solve specific synthesis systems. For the numerical experiment, the target system was constructed by generating 18 random complex numbers within the box defined by corners $(-1 - i, 1 + i)$, assigning these numbers to P_j and \bar{P}_j , $j = 0, \dots, 8$,

$$\begin{aligned} P_0 &= 0.776874 - 0.642684i, & \bar{P}_0 &= 0.873111 + 0.292468i, \\ P_1 &= 0.549479 + 0.418241i, & \bar{P}_1 &= 0.689034 - 0.944901i, \\ P_2 &= -0.261986 - 0.403618i, & \bar{P}_2 &= 0.854109 - 0.482044i, \\ P_3 &= 0.767234 - 0.378801i, & \bar{P}_3 &= -0.268805 - 0.865098i, \\ P_4 &= -0.068090 + 0.919907i, & \bar{P}_4 &= -0.263339 - 0.451987i, \\ P_5 &= 0.055654 + 0.675009i, & \bar{P}_5 &= -0.221326 + 0.775947i, \\ P_6 &= -0.134960 - 0.424099i, & \bar{P}_6 &= 0.165736 + 0.319065i, \\ P_7 &= -0.239858 + 0.041043i, & \bar{P}_7 &= 0.248468 + 0.077381i, \\ P_8 &= 0.635501 - 0.474101i, & \bar{P}_8 &= 0.342355 - 0.501518i \end{aligned} \quad (21)$$

and then substituting these parameters into \mathcal{T} . To obtain a numerically general system, conjugate relationships were broken.

FRG requires the construction of startpoints and start systems from randomly generated mechanisms, which is accomplished by generating five random complex numbers within the box defined by $(-0.5 - 0.5i, 0.5 + 0.5i)$. Four of these numbers define a startpoint

$$s_p = \{A, B, C, D, \bar{A}, \bar{B}, \bar{C}, \bar{D}\} \quad (22)$$

and the fifth is assigned to P_0 . Here, conjugate relationships remain intact so that these five numbers define a four-bar mechanism. Eight configurations are selected from this mechanism with disregard to whether they belong to the same circuit. These eight configurations define points (P_j, \bar{P}_j) , $j = 1, \dots, 8$, which along with (P_0, \bar{P}_0) are substituted into \mathcal{T} to obtain a start system that corresponds to s_p . Random distributions were uniform.

Path tracking was completed using the algorithm within the software BERTINI [20]. This path tracking algorithm utilizes random numbers to form projective patches and defines homotopies based on a random parameter γ [5] (unrelated to the constant of Eq. (6)). Occasionally, BERTINI would experience a path tracking failure, e.g., reaching a minimum step size. In these cases, an FRG iteration would be considered invalid. Table 1 shows a few startpoints and start system parameters used to find roots of the target system defined by the parameters of Eq. (21).

Table 1 Startpoints and start system parameters used to track homotopies to roots of the target system defined by parameters shown in Eq. (21). Homotopies for the roots below were constructed with $\gamma = 0.573937 + 1.472072i$.

		Root 1	Root 2	Root 3
Startpoints	A	-0.185931 -0.490157 <i>i</i>	0.226527 -0.447966 <i>i</i>	0.296386 -0.339809 <i>i</i>
	B	-0.441407 -0.138234 <i>i</i>	0.416623 -0.143895 <i>i</i>	0.037220 -0.462856 <i>i</i>
	C	0.064826 +0.359267 <i>i</i>	-0.023306 -0.024924 <i>i</i>	-0.476241 -0.281314 <i>i</i>
	D	-0.477876 +0.440497 <i>i</i>	-0.344134 -0.420386 <i>i</i>	-0.044317 +0.028224 <i>i</i>
	\bar{A}	-0.185931 +0.490157 <i>i</i>	0.226527 +0.447966 <i>i</i>	0.296386 +0.339809 <i>i</i>
	\bar{B}	-0.441407 +0.138234 <i>i</i>	0.416623 +0.143895 <i>i</i>	0.037220 +0.462856 <i>i</i>
	\bar{C}	0.064826 -0.359267 <i>i</i>	-0.023306 +0.024924 <i>i</i>	-0.476241 +0.281314 <i>i</i>
	\bar{D}	-0.477876 -0.440497 <i>i</i>	-0.344134 +0.420386 <i>i</i>	-0.044317 -0.028224 <i>i</i>
Start system parameters	P_0	-0.207946 +0.369011 <i>i</i>	0.384005 +0.244221 <i>i</i>	-0.155698 -0.422040 <i>i</i>
	P_1	-0.115369 +0.351828 <i>i</i>	-0.321480 -0.783846 <i>i</i>	0.273905 -1.211253 <i>i</i>
	P_2	0.393717 -0.190376 <i>i</i>	-0.042075 -1.265047 <i>i</i>	-0.064827 -1.195472 <i>i</i>
	P_3	-0.312109 +0.319537 <i>i</i>	-0.524641 -0.489862 <i>i</i>	-0.086717 +0.118875 <i>i</i>
	P_4	-0.608101 -0.973176 <i>i</i>	1.141800 -0.508283 <i>i</i>	-0.025993 -0.630360 <i>i</i>
	P_5	-1.065637 -0.419560 <i>i</i>	-0.329956 +0.122805 <i>i</i>	0.404738 -0.750646 <i>i</i>
	P_6	-0.754964 -0.722653 <i>i</i>	-0.152737 -0.865539 <i>i</i>	-0.736730 -0.262438 <i>i</i>
	P_7	-1.022605 -0.077858 <i>i</i>	0.224757 -0.522349 <i>i</i>	-0.100214 -0.501486 <i>i</i>
	P_8	-0.778031 -0.906804 <i>i</i>	0.152844 -0.728661 <i>i</i>	-0.340800 +0.386734 <i>i</i>
	\bar{P}_0	-0.207946 -0.369011 <i>i</i>	0.384005 -0.244221 <i>i</i>	-0.155698 +0.422040 <i>i</i>
	\bar{P}_1	-0.115369 -0.351828 <i>i</i>	-0.321480 +0.783846 <i>i</i>	0.273905 +1.211253 <i>i</i>
	\bar{P}_2	0.393717 +0.190376 <i>i</i>	-0.042075 +1.265047 <i>i</i>	-0.064827 +1.195472 <i>i</i>
	\bar{P}_3	-0.312109 -0.319537 <i>i</i>	-0.524641 +0.489862 <i>i</i>	-0.086717 -0.118875 <i>i</i>
	\bar{P}_4	-0.608101 +0.973176 <i>i</i>	1.141800 +0.508283 <i>i</i>	-0.025993 +0.630360 <i>i</i>
	\bar{P}_5	-1.065637 +0.419560 <i>i</i>	-0.329956 -0.122805 <i>i</i>	0.404738 +0.750646 <i>i</i>
	\bar{P}_6	-0.754964 +0.722653 <i>i</i>	-0.152737 +0.865539 <i>i</i>	-0.736730 +0.262438 <i>i</i>
	\bar{P}_7	-1.022605 +0.077858 <i>i</i>	0.224757 +0.522349 <i>i</i>	-0.100214 +0.501486 <i>i</i>
	\bar{P}_8	-0.778031 +0.906804 <i>i</i>	0.152844 +0.728661 <i>i</i>	-0.340800 -0.386734 <i>i</i>
Roots obtained	A	1.208688 -0.064110 <i>i</i>	-0.278014 -3.903201 <i>i</i>	1.363042 +0.327298 <i>i</i>
	B	0.229275 +0.728512 <i>i</i>	-2.851690 -4.028216 <i>i</i>	-0.236355 -0.424120 <i>i</i>
	C	1.149090 -0.376585 <i>i</i>	-0.356765 -3.870173 <i>i</i>	0.744465 -0.469276 <i>i</i>
	D	0.724958 -0.465222 <i>i</i>	-2.452992 -3.479546 <i>i</i>	0.939745 -0.691008 <i>i</i>
	\bar{A}	0.322662 -0.323042 <i>i</i>	0.942263 -5.099374 <i>i</i>	0.026897 -0.657391 <i>i</i>
	\bar{B}	-0.968032 +1.383404 <i>i</i>	-5.662954 -2.367803 <i>i</i>	-0.079412 -0.799504 <i>i</i>
	\bar{C}	0.512174 +0.503870 <i>i</i>	2.411689 -5.258779 <i>i</i>	0.931018 +0.261914 <i>i</i>
	\bar{D}	-1.469395 +0.697904 <i>i</i>	-5.912599 -3.042769 <i>i</i>	-1.178260 -0.792065 <i>i</i>

5 Results

The FRG method was applied to find all 8652 roots of a numerically general version of \mathcal{T} . The algorithm was executed for 100,000 trials. By the 87,549th trial, the algorithm found 8651 roots, but did not find the final root beyond that. The expectation

was to find all the roots on the 84,430th trial within a standard deviation of 11,092 trials. Curtailing results before the brunt of diminishing returns, the algorithm found 95% of the roots by 26,658 trials, underperforming the expected value by 736 trials. Similar to expectations, FRG found over 95% of the roots in 30% of the iterations required to find 99.988% of the roots. This is depicted in Fig. 6, which shows the number of roots obtained as trials progressed. Roots accumulated quickly in the beginning leading to a diminishing rate of returns such that at least 70% of the calculations were dedicated for obtaining the final 5% of the roots. Table 2 gives the number of trials performed versus expected to acquire various percentages of the complete solution set.

As trials progressed, an estimate of the percentage of roots obtained was calculated from the success ratio using Eq. (10). This is a useful feature for systems in which the total number of roots is unknown. These estimates are compared to the actual percentages for the known benchmark in Table 2, and the running difference between the two is graphed in Fig. 7. Estimation error was greatest at the beginning of FRG, however, as the number of sample trials that comprise the calculation of the success ratio increased, the estimation became more accurate. Predictions of the total number of roots are given in Table 2. The success ratio as a function of the percentage of roots obtained for the benchmark is shown in Fig. 8 compared to the theoretical function from Eq. (9). This function illustrates the sharp decrease in success near the end of FRG, tending toward zero for systems with more roots.

The error estimates of this paper are based on the assumption that roots appear with equal probability. Conducting Pearson's

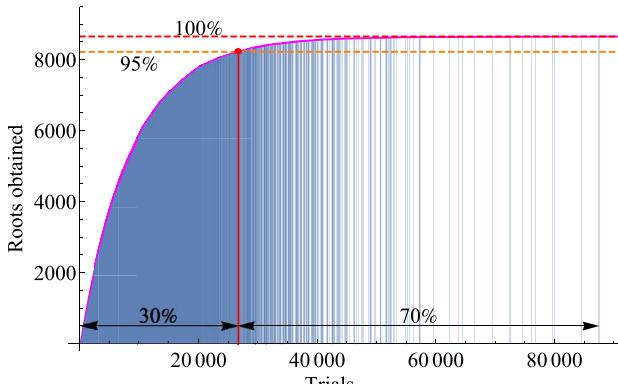


Fig. 6 The number of roots obtained as FRG trials progressed for the four-bar benchmark problem. Vertical lines mark trials that resulted in a new root. Their frequency forms a solid colored section at the beginning of the algorithm. The computational effort to find 95% of roots and the final 5% of roots are dimensioned as percentages of the total number trials needed to find 99.988% of roots.

Table 2 Trials required to find various percentages of roots versus the expectation for the benchmark problem, and estimations of the total root count based on the success ratio

Percentage of roots obtained	Roots obtained	Trials performed	Trials expected	Standard deviation of trials expected	Success ratio	Estimated percentage of roots obtained	Estimated total root count
10.009%	866	905	912	7.1	95.691%	8.493%	10,196
20.007%	1731	1918	1931	15.2	90.250%	18.844%	9186
30.005%	2596	3083	3086	24.9	84.204%	29.834%	8701
40.002%	3461	4446	4420	36.7	77.845%	40.767%	8490
50.000%	4326	6068	5997	51.5	71.292%	51.313%	8431
60.009%	5192	8063	7929	71.1	64.393%	61.554%	8435
70.007%	6057	10,477	10,418	98.9	57.812%	70.420%	8601
80.005%	6922	14,179	13,925	143.8	48.819%	80.952%	8551
90.002%	7787	20,085	19,919	240.7	38.770%	90.249%	8628
95.007%	8220	26,658	25,922	372.2	30.835%	95.479%	8609
96.001%	8306	28,617	27,840	423.7	29.025%	96.388%	8617
97.006%	8393	31,105	30,341	499.5	26.983%	97.282%	8627
98.000%	8479	34,912	33,824	623.7	24.287%	98.250%	8630
99.006%	8566	40,990	39,846	903.8	20.898%	99.129%	8641
99.965%	8649	84,632	67,568	4601.1	10.220%	99.994%	8649
99.977%	8650	87,384	70,452	5430.0	9.899%	99.996%	8650
99.988%	8651	87,549	74,778	6942.2	9.881%	99.996%	8651

chi-squared test on tabulated root appearances revealed that this was not the case. Nonetheless, Figs. 6–8 and Table 2 demonstrate reasonable accuracy of the estimates. The path failure rate was 5% for the benchmark.

6 Example

The motivation behind FRG is to obtain solution sets to large polynomial systems for kinematic design. In this paper, the method is introduced and only applied to four-bar path synthesis for benchmarking. In order to bridge FRG with tangible kinematic

design, this section presents an example. The objective is to design a linkage that traces a D-shaped path potentially for a walking mechanism.

To begin, nine points were selected

$$\begin{aligned}
 P_0 &= 0 - 0.0625i, & P_3 &= 0.95 + 0.2i, & P_6 &= -0.95 + 0.2i, \\
 P_1 &= 0.5 - 0.05i, & P_4 &= 0.35 + 0.4i, & P_7 &= -1 + 0i, \\
 P_2 &= 1 - 0i, & P_5 &= -0.35 + 0.4i, & P_8 &= -0.5 - 0.05i
 \end{aligned} \tag{23}$$

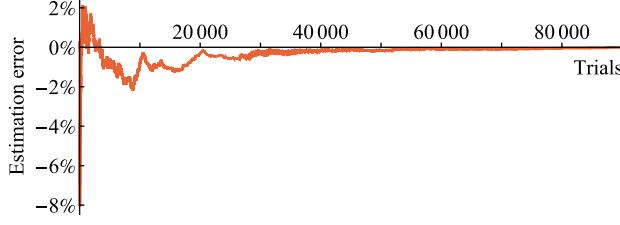


Fig. 7 Difference between the estimated and actual percentage of roots obtained as trials progressed

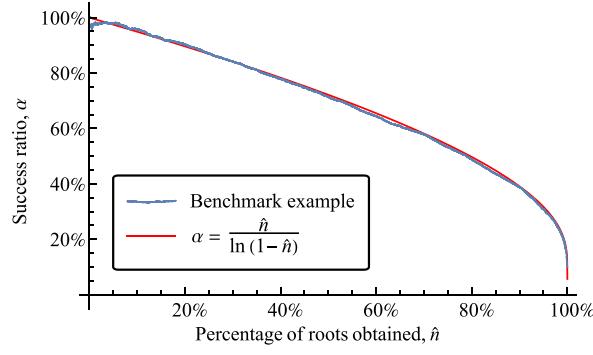
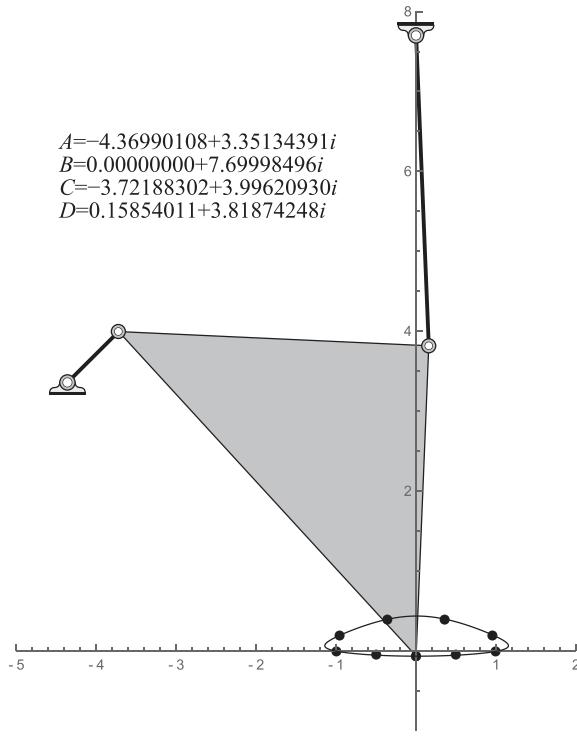


Fig. 8 The success ratio α plotted against the percentage of roots obtained \hat{n} for both the benchmark experiment and theoretical curve

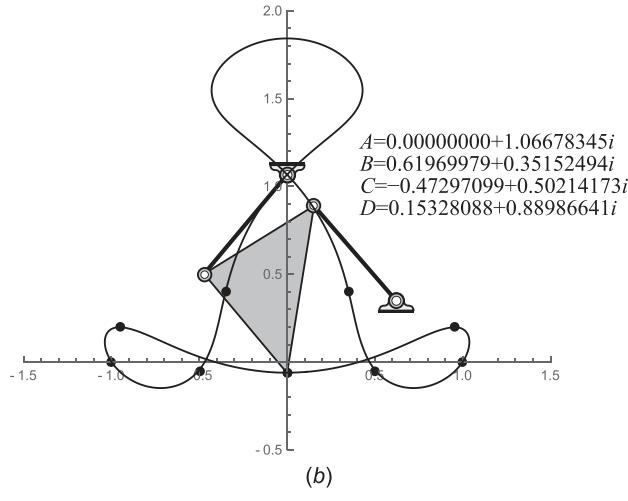
Note that the straight portion of the “D” was curved (Figs. 9 and 10) in the hope that this would result in four bars with smaller link lengths. Substituting Eq. (23) into \mathcal{T} forms a synthesis system which was solved by a parameter homotopy constructed from the 8651 roots obtained in Sec. 5. The results include 554 roots that represent physical link geometry of which 16 were found to be free of branch and circuit defects. Note that a finite root is necessary but not sufficient for a solution to represent physical geometry. In other words, the number of finite roots provides a bound for useful engineering solutions.

The number of linkage solutions found should be divisible by six because of known symmetries in the synthesis system. This was not the case due to a path failure rate of 7% during parameter homotopy. Solutions should exist in symmetric pairs with values $A \leftrightarrow B$ and $C \leftrightarrow D$ swapped, as well as in sets of three cognates [21], creating a six-way symmetry. With this in mind, the benchmark problem may be reduced to collecting $N = 1442$ sets of roots.

Examples of defect-free and defective linkages are given in Figs. 9 and 10. The linkage shown in Fig. 9(a) has the potential to be used as a walking mechanism although its large link lengths are not ideal. The linkages shown in Fig. 10 are similar to Chebyshev cognate linkages, which are known to trace a similar “D”. A third Chebyshev-like linkage was found as well, which is the horizontal mirror of Fig. 10(a). The linkage shown in Fig. 10(b) is branch-defective meaning it cannot be actuated from a base joint. Linkage designs were analyzed for branch and circuit defects by computing mechanism singularities and ranges of motion. This analysis did not consider order defects. The lack of strong design candidates from the example’s results motivates a search into



(a)



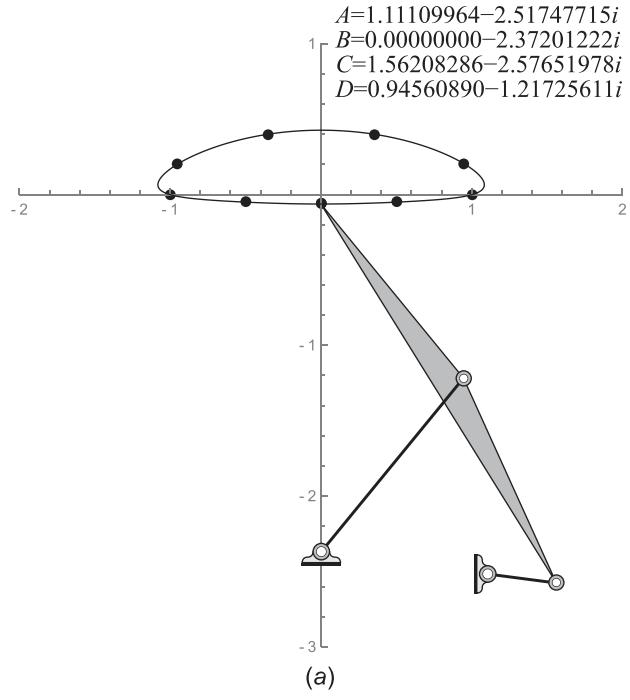
(b)

Fig. 9 (a) A defect-free mechanism and **(b)** a mechanism with a branch defect

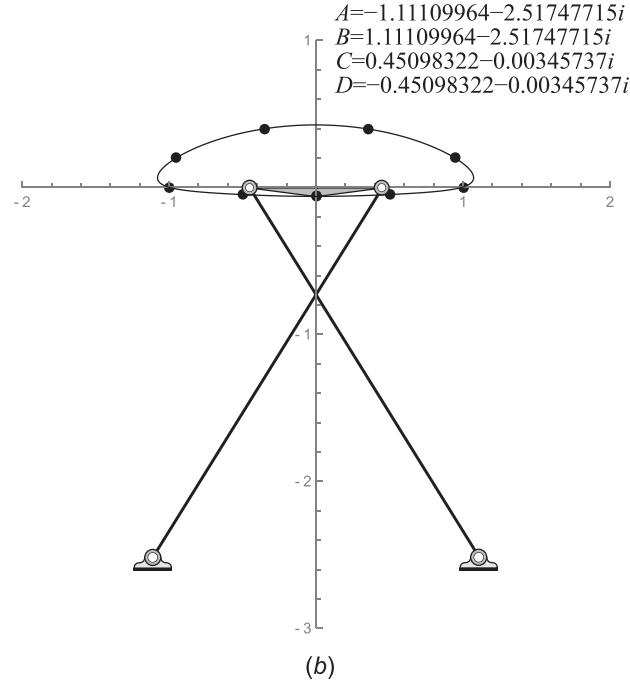
more complex linkage topologies, which is out of the scope of this paper.

7 Conclusion

The FRG technique was introduced for solving large polynomial systems for the kinematic design of linkages. The technique provides a means of generating homotopy continuation startpoints that track to the finite roots of a numerically general target system, precluding infinite roots which often consume the large majority of computations. The rate at which FRG accumulates unique roots is modeled as a coupon collector problem which also provides means for estimating the entire root set. Applying FRG to a benchmark four-bar design problem found all but one root in 42% less paths than the best known regeneration method. Characteristic of FRG is a diminishing rate of returns for finding the final roots. FRG found



(a)



(b)

Fig. 10 (a) Synthesis results that resemble Chebyshev linkages: **(a)** a defect-free mechanism and **(b)** a mechanism with a branch defect

95% of roots in 82% less paths than the best known regeneration method. The results of the algorithm were applied to design a few four bars.

Acknowledgment

The authors gratefully acknowledge the support of the National Science Foundation Award Nos. CMMI-1549667 and CMMI-1636302. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] Wampler, C. W., Sommese, A. J., and Morgan, A. P., 1992, "Complete Solution of the Nine-Point Path Synthesis Problem for Four-Bar Linkages," *ASME J. Mech. Des.*, **114**(1), pp. 153–159.

[2] Plecnik, M., and McCarthy, J. M., 2015, "Computational Design of Stephenson II Six-Bar Function Generators for 11 Accuracy Points," *ASME J. Mech. Rob.*, **8**(1), p. 011017.

[3] Plecnik, M., and McCarthy, J. M., 2016, "Kinematic Synthesis of Stephenson III Six-Bar Function Generators," *Mech. Mach. Theory*, **97**, pp. 112–126.

[4] Root, R. R., and Ragsdell, K. M., 1976, "A Survey of Optimization Methods Applied to the Design of Mechanisms," *J. Eng. Ind.*, **98**(3), pp. 1036–1041.

[5] Bates, D. J., Hauenstein, J. D., Sommese, A. J., and Wampler, C. W., 2013, *Numerically Solving Polynomial Systems With Bertini*, SIAM Press, Philadelphia, PA, p. 25.

[6] Roth, B., and Freudenstein, F., 1963, "Synthesis of Path-Generating Mechanisms by Numerical Methods," *J. Eng. Ind.*, **85**(3), pp. 298–304.

[7] Wise, S. M., Sommese, A. J., and Watson, L. T., 2000, "Algorithm 801: Polysys_PLP—A Partitioned Linear Product Homotopy Code for Solving Polynomial Systems of Equations," *ACM Trans. Math. Soft.*, **26**(1), pp. 176–200.

[8] Su, H. J., McCarthy, J. M., Sosonkina, M., and Watson, L. T., 2006, "Algorithm 857: POLSYS_GLP—A Parallel General Linear Product Homotopy Code for Solving Polynomial Systems of Equations," *ACM Trans. Math. Soft.*, **32**(4), pp. 561–579.

[9] Verschelde, J., 2011, "Polynomial Homotopy Continuation With PHCpack," *ACM Commun. Comput. Algebra*, **44**(3/4), pp. 217–220.

[10] Chen, T., Lee, T. L., and Li, T. Y., 2014, "Hom4PS-3: A Parallel Numerical Solver for Systems of Polynomial Equations Based on Polyhedral Homotopy Continuation Methods," 4th International Conference on Mathematical Software, (ICMS) Seoul, South Korea, Aug. 5–9, pp. 183–190.

[11] Li, T. Y., Sauer, T., and Yorke, J. A., 1989, "The Cheater's Homotopy: An Efficient Procedure for Solving Systems of Polynomial Equations," *SIAM J. Numer. Anal.*, **26**(5), pp. 1241–1251.

[12] Raghavan, M., and Roth, B., 1995, "Solving Polynomial Systems for the Kinematic Analysis and Synthesis of Mechanisms and Robot Manipulators," *ASME J. Vib. Acoust.*, **117**(B), pp. 71–79.

[13] Su, H., and McCarthy, J. M., 2005, "The Synthesis of an RPS Serial Chain to Reach a Given Set of Task Positions," *Mech. Mach. Theory*, **40**(7), pp. 757–775.

[14] Masouleh, M. T., Gosselin, C., Husty, M., and Walter, D., 2011, "Forward Kinematic Problem of 5-RPUR Parallel Mechanisms (3T2R) With Identical Limb Structures," *Mech. Mach. Theory*, **46**(7), pp. 945–959.

[15] Lee, E., Mavroidis, C., and Merlet, J. P., 2002, "Five Precision Points Synthesis of Spatial RRR Manipulators Using Interval Analysis," *ASME Paper No. DETC2002/MECH-34272*.

[16] Tari, H., Su, H., and Li, T., 2010, "A Constrained Homotopy Technique for Excluding Unwanted Solutions From Polynomial Equations Arising in Kinematics Problems," *Mech. Mach. Theory*, **45**(6), pp. 898–910.

[17] Tsai, L. W., and Lu, J. J., 1990, "Coupler-Point-Curve Synthesis Using Homotopy Methods," *ASME J. Mech. Des.*, **112**(3), pp. 384–389.

[18] Parzen, E., 1960, *Modern Probability Theory and Its Applications*, Wiley, New York, p. 369.

[19] Mortici, C., and Villarino, M. B., 2015, "On the Ramanujan–Lodge Harmonic Number Expansion," *Appl. Math. Comput.*, **251**, pp. 423–430.

[20] Bates, D. J., Hauenstein, J. D., Sommese, A. J., and Wampler, C. W., "Bertini: Software for Numerical Algebraic Geometry," Bertini Software, University of Notre Dame, Notre Dame, IN, www.Bertini.nd.edu

[21] Roberts, S., 1875, "Three-Bar Motion in Plane Space," *Proc. London Math. Soc.*, **s1-7**(1), pp. 14–23.