Kinematics of linkages

“Computer Aided Geometric Design”
Final Presentation
Agenda

• Introduction to mechanisms
• Mapping of the motion into algebraic constraints.
• Examples of linkages
• Major research and development
  • “Geometric Constraint Solver using Multivariate Rational Spline Functions” Gershon Elber, Myung-Soo Kim.
  • “Solving Piecewise Polynomial Constraint Systems with Decomposition and a Subdivision-Based Solver” Boris van Sosin and Gershon Elber.
  • “Precise contact motion planning for deformable planar curved shapes” Yong-Joon Kim, Gershon Elber, Myung-Soo Kim.
Introduction to mechanisms
Kinematic Mechanism

• A mechanism is a device to transform one motion into another.

• Comprised of rigid bodies connected such that each one moves with respect to another.

• The connection is a joint between two members permitting a particular kind of motion.

• The joints between links are modeled as providing ideal movement, pure rotation or sliding.

Mapping of the motion into algebraic constraints.
Representation of a Mechanism

• A kinematic mechanism \( M = \{ E, C \} \), contains:
  • \( E \) - a set of elements from which the mechanism is built.
  • \( C \) - a set of constraints among the element.

• Under motion of the mechanism, the constraints \( C \) should be preserved.
The elements of a mechanism

- The basic build block of the mechanism is a 2D or a 3D point.
- The point types:
  - *An anchored point* - point that lies fixed and does not change its position during the motion.
  - *Point on a Curve* - point that can move while its trajectory is constrained to a curve.
  - *Point on a Surface* - point that can move while its trajectory is constrained to be on a surface.
  - *A free point* - point that can move in any direction.
- Kinematic bars and kinematic faces are defined as pairs and triplets of kinematic points, respectively.

source: [Kinematic Simulation of Planar and Spatial Mechanisms using a polynomial constraint solver](https://example.com) by Michael Barton, Nadav Shragai, Gershon Elber.
The Constraints in a Mechanism

• Distance Constraints:

Point-point (black bar)

Point-bar

Bar-bar

Point-curve

Point-surface

Bar-curve

Bar-surface

source: Kinematic Simulation of Planar and Spatial Mechanisms using a polynomial constraint solver" Michael Barton, Nadav Shragai, Gershon Elber.
The Constraints in a Mechanism

• Angular Constraints:
  • bar-bar
  • bar-plane,

• Tangency:
  • bar--curve
  • bar—surface
  • face-surface

• Parallelism
  • Bar-bar

source: Kinematic Simulation of Planar and Spatial Mechanisms using a polynomial constraint solver Michael Barton, Nadav Shragai, Gershon Elber.
Examples

• Distance preserving constraints \( d \) between points \( P \) and \( Q \)
  
  \[ \| P - Q \|^2 - d^2 = 0 \]

• Angle constraints between two bars \( PQ \) and \( RT \)
  
  \[ \frac{(P-Q,R-T)^2}{\|P-Q\|^2\|R-T\|^2} - \cos^2(\alpha) = 0 \]

• Point-surface distance between point \( P \) and surface \( Q(u, v) \)

\[
\begin{align*}
|P - Q| - d^2 &= 0 \\
\left\langle \frac{\partial Q}{\partial u}, P - Q \right\rangle &= 0 \\
\left\langle \frac{\partial Q}{\partial v}, P - Q \right\rangle &= 0
\end{align*}
\]

source: "Kinematic Simulation of Planar and Spatial Mechanisms using a polynomial constraint solver" Michael Barton, Nadav Shragai, Gershon Elber.
Examples of linkages
A planar four-bar linkage consists of four rigid rods in the plane connected by pin joints. We call the rods:

- Frame link: fixed to anchor pivots $AA$ and $BB$.
- Crank link: driven by input angle $\alpha$.
- Rocker link: gives output angle $\beta$.
- Rocker link: connects the two moving pins $C$ and $D$. 

Four-bar linkage

• can be used for many mechanical purposes, for example:
  • convert rotational motion to reciprocating motion. (e.g., pumpjack)
  • convert reciprocating motion to rotational motion (e.g., bicycle)
  • constrain motion (e.g., knee joint and suspension)
  • magnify force (e.g., parrotfish jaw)

source: http://dynref.engr.illinois.edu/aml.html
Cognate Linkage

- linkages that ensure the same input-output relationship or coupler curve geometry, while being dimensionally dissimilar.
- Roberts–Chebyshev Theorem:
  - each coupler curve can be generated by three different four-bar linkages.
- Overconstrained mechanisms can be obtained by connecting two or more cognate linkages together.

Chebyshev Linkage

• A mechanical linkage (4-bar) that converts rotational motion to approximate straight-line motion.
• invented by Pafnuty Chebyshev.
• Algebraic relation between the lengths:
  • \( L_4 = L_3 + \sqrt{L_2^2 - L_1^2} \)
• Lengths proportions:
  • \( L1:L2:L3 = 4:5:2 \)
• From the proportions and constraints it follows that:
  • \( L_2 = L_4 \)

Chebyshev’s Lambda Mechanism

• A four-bar mechanism that converts rotational motion to approximate straight-line motion with approximate constant velocity.

• Cognate linkage of the Chebyshev linkage

source: http://en.wikipedia.org/wiki/Chebyshev%27s_Lambda_Mechanism
Jansen Linkage

• A planar leg mechanism

• Designed by the kinetic sculptor Theo Jansen

• Generates a smooth walking motion.

• One degree of freedom

• Applications in mobile robotics and in gait analysis

source: http://en.wikipedia.org/wiki/Jansen%27s_linkage
Jansen Linkage

source: http://en.wikipedia.org/wiki/Jansen%27s_linkage
Pantograph

• mechanical linkage connected in a manner based on parallelograms so that the movement of one pen, in tracing an image, produces identical movements in a second pen.

• If a line drawing is traced by the first point, an identical, enlarged, or miniaturized copy will be drawn by a pen fixed to the other.

• Using the same principle, different kinds of pantographs are used for other forms of duplication in areas such as sculpture, minting, engraving, and milling.

source:https://en.wikipedia.org/wiki/Pantograph
Major Research and development
“Geometric Constraint Solver using Multivariate Rational Spline Functions”
Gershon Elber, Myung-Soo Kim
The problem

• given \( n \) multivariate piecewise rational constraints
  \[
  F_i(u_1, ... u_{m-1}) = 0, \quad i = 1, ..., n
  \]

• We seek all \( u^s \in \mathbb{R}^{m-1} \), such that \( F_i(u^s) = 0 \) for all \( i = 1, ..., n \)

• The \( F_i's \) are represented as B-splines or Bezier multivariates scalar surfaces.

• Inequalities \( F_i(u_1, ... u_{m-1}) > 0 \) are also supported.

source: “Geometric Constraint Solver using Multivariate Rational Spline Functions” Gershon Elber, Myung-Soo Kim
The convex-Hull test

• The convex-Hull property states that a Bézier curve is contained inside the convex hull of its control mesh.

• By the convex-hull property, the domain of $F_i(u)$ contains zeros only if the control coefficients of $F_i$ have different signs.

source: “Geometric Constraint Solver using Multivariate Rational Spline Functions” Gershon Elber, Myung-Soo Kim
The uniqueness test

• Bound the set of tangent directions in a pair of cones.
• If we position the pair of cones on any point on $c$, the entire curve will be contained in the cones.
• If the cones of tangent directions of $c_0$ and $c_1$ do not overlap (except for the apex), then $c_0$ and $c_1$ intersect at most once.
• Can be extended multivariates.

source: “Geometric Constraint Solver using Multivariate Rational Spline Functions” Gershon Elber, Myung-Soo Kim
The solver

• subdivision based.
• For each sub-domain:
  • The convex-hull test is performed: if all the control points have the same sign, the solution is not in the sub-domain and the sub-domain is trimmed from the search domain.
  • The uniqueness test is performed: if there is a unique solution, the subdivision stops and a numerical procedure is operated. the segment is numerically traced up to a user defined accuracy.
  • Otherwise, the subdomain $D \in \mathbb{R}^n$ is recursively divided until a condition for the existence of a single univariate solution segment can be met.
• At the end it generates a set of discrete points which are the simultaneous zero-set of $F_i$.
• Can support inequality constraints by checking if the control points have the same signs as the constraint.

source: “Geometric Constraint Solver using Multivariate Rational Spline Functions” Gershon Elber, Myung-So Kim
“Solving Piecewise Polynomial Constraint Systems with Decomposition and a Subdivision-Based Solver” Boris van Sosin and Gershon Elber.
Motivation

• Constraint systems are used in all computational geometry and CAD systems.
• In many applications, geometric constraint systems can get very complicated.
• Even a constraint as simple as a Euclidean distance between two points is quadratic
• Therefore, most applications require numerical solvers for constraints systems.
• The time it takes for most constraint systems solvers to run scales non-linearly, even exponentially, with the size of the problem.
• Therefore, decomposing the problem into a series of sub-problems in sequence can be very effective at speeding up the solution process.

source: “Solving Piecewise Polynomial Constraint Systems with Decomposition and a Subdivision-Based Solver” Boris van Sosin and Gershon Elber.
The problem

• Solving non-linear constraint systems by decomposing them into subsystems.

• Each sub-system is solved by using a subdivision-based polynomial solver.

• The input constraint system:
  • represented as Bezier or B-spline multivariate functions.
  • with DOF=0 or DOF=1
  • Contains equality and inequality constraints.

\[
\begin{align*}
|A - P_0|^2 - c_0^2 &= 0, & c_0 &= 0.9 \\
|A - B|^2 - c_2^2 &= 0, & c_2 &= 2 \\
|B - P_1|^2 - c_3^2 &= 0, & c_3 &= 0.9
\end{align*}
\]
Algebraic representation of the constraints

• Constraints can be expressed in algebraic form as equations.
• Two types:
  • Zero constraints – have the form \( f(x_1, x_2, \ldots, x_n) = 0 \).
  • Inequality constraints- usually have the form \( f(x_1, x_2, \ldots, x_n) \geq 0 \).
• The number of Zero constraints together with the number of variables determine the number of degrees of freedom of the system (DOF).
• Inequality constraints do not affect the total degrees of freedom of the system, but restrict the domain of the solution search.

source: “Solving Piecewise Polynomial Constraint Systems with Decomposition and a Subdivision-Based Solver” Boris van Sosin and Gershon Elber.
Algebraic representation of the constraints

• All the constraints are represented as B-spline functions.

• Extended to multivariate as a tensor product:

\[ M(t) = \sum_{i_0=0}^{n_0} \sum_{i_0=0}^{n_0} \cdots \sum_{i_0=0}^{n_0} \left( P_{i_0,i_1,\ldots,i_{k-1}} \prod_{j=0}^{k-1} B_{i_j}^t(t_j) \right) \]

• Where:

  • \( B_{i_j}^t \) - the B-spline basis functions of order \( q_j \)

source: “Solving Piecewise Polynomial Constraint Systems with Decomposition and a Subdivision-Based Solver” Boris van Sosin and Gershon Elber.
Constraint system

- A constraint system is $M = M^\text{zero} \cup M^\text{ineq}$
- where:
  - $M^\text{zero} = \begin{cases} \ M_0^\text{zero}(t) = 0 \\
                             \ M_1^\text{zero}(t) = 0 \\
                             \vdots \\
                             \ M_{m-1}^\text{zero}(t) = 0 \end{cases}$, $M^\text{ineq} = \begin{cases} \ M_0^\text{ineq}(t) \geq 0 \\
                             \ M_1^\text{ineq}(t) \geq 0 \\
                             \vdots \\
                             \ M_{p-1}^\text{ineq}(t) \geq 0 \end{cases}$
  - $t = (t_0, t_1, \ldots t_{k-1})$ the variables
  - $k$ is number of variables
  - $m > 0$ number of zero constraints.
  - $p \geq 0$ number of inequality constraints.

source: “Solving Piecewise Polynomial Constraint Systems with Decomposition and a Subdivision-Based Solver” Boris van Sosin and Gershon Elber.
Solution plan graph

- Let $M$ be a constraint system with $m$ zero constraints and $k$ variables.

- A solution plan graph is a graph $H_{plan} = (V_{plan}, E_{plan})$ with the following properties:
  - $H_{plan}$ is a DAG (directed acyclic graph)
  - Each vertex $v_i \in V_{plan}$ represents a step in the solution plan with attached $\text{constraints}(v_i) = \{M_j\}_{v_i}$, the set of constraints of the subproblem solved in this step.
  - For $v_i \neq v_j$, $\text{constraints}(v_i) \cap \text{constraints}(v_j) = \phi$ and $\bigcup_{v \in V_{plan}} \text{constraints}(v) = M$.
  - There is an edge $v_i \rightarrow v_j \in E_{plan}$ if the subproblem $v_j$ is dependent on $v_i$.

source: “Solving Piecewise Polynomial Constraint Systems with Decomposition and a Subdivision-Based Solver” Boris van Sosin and Gershon Elber.
Variable-constraint graph

- A bipartite graph $G = (V_v \cup V_M, E)$ with the following properties:
  1. $V_v = \{V_{t_i}\}_{i=0}^{k-1}$, a vertex for each variable $t_i$ in the vector $t$
  2. $V_M = \{v_{M_j}\}_{j=0}^{m-1}$, a vertex for each constraint in $M^{zero}$.
  3. There is an edge $(v_t, v_{M_j}) \in E$ iff constraint $M_j$ is dependent on variable $t_i$.

source: “Solving Piecewise Polynomial Constraint Systems with Decomposition and a Subdivision-Based Solver” Boris van Sosin and Gershon Elber.
Building of $H_{plan}$ from $G$

1. Maximum matching $V_v$ in $G$
2. Build $G'$
3. Condense $G'$
4. Strongly connected components (SCC)

source: “Solving Piecewise Polynomial Constraint Systems with Decomposition and a Subdivision-Based Solver” Boris van Sosin and Gershon Elber.
Building the solution graph $H_{plan}$ from the variable-constraints graph $G = (V_v \cup V_M, E)$

1. Finding a maximum matching, $M_m$ in $G$

source: “Solving Piecewise Polynomial Constraint Systems with Decomposition and a Subdivision-Based Solver” Boris van Sosin and Gershon Elber.
Building the solution graph $H_{plan}$ from the variable-constraints graph $G = (V_v \cup V_M, E)$

2. Building a new directed graph $G' = (V', E')$ where:

- $V' = V_v \cup V_M$
- converting each of the matched edges into a pair of anti parallel directed edges.
- All the unmatched edges are copied to $G'$ as directed edges from the variable vertices to the constraint vertices.

source: “Solving Piecewise Polynomial Constraint Systems with Decomposition and a Subdivision-Based Solver” Boris van Sosin and Gershon Elber.
Building the solution graph $H_{plan}$ from the variable-constraints graph $G = (V_v \cup V_M, E)$

3. Condensing $G'$ to build $H$

by taking the set of vertices of $G'$ and connecting a pair of vertices $v_{M_i}, v_{M_j}$ by a directed edge $v_{M_i} \rightarrow v_{M_j}$ iff in $G'$ there is a directed path from $v_{M_i}$ to $v_{M_j}$ going through a single variable vertex.
Building the solution graph \( H_{\text{plan}} \) from the variable-constraints graph \( G = (V_v \cup V_M, E) \)

4. \( H_{\text{plan}} \) is the Strongly connected components (SCC) graph of \( H \) (without inequality constraints).
Building the solution graph $H_{plan}$ from the variable-constraints graph $G = (V_v \cup V_M, E)$

5. Adding inequality constraints to $H_{plan}$

The inequality constraints are added to the subsystems in which they can be solved.

Each inequality constraint $M_i^{ineq}$ is added to the subsystem in which at least one of the variables on which $M_i^{ineq}$ depends is being solved for, in the subsystem, and all the variables which are not being solved for in the current subsystem, already have solutions.

source: “Solving Piecewise Polynomial Constraint Systems with Decomposition and a Subdivision-Based Solver” Boris van Sosin and Gershon Elber.
The solution phase

• The subsystems in $H_{plan}$ are solved in a topological order.
• Solving in topological order assuring that when a subsystem is solved, there are already values assigned to all the variables which are required to solve it.
• When the first subsystem is solved, the computed values are assigned to the variables in the subsystem.
• Zero dimensional or univariate assigned as a solution to a variable.
• Univariate solutions need to be parameterized in order to have the same representation as the constraints for further processing.
• The parametrization of univariate solutions is done either by parameterizing the piecewise-linear solutions directly, or by fitting a parametric curve (a B-spline curve) which approximates the piecewise-linear solution.

source: “Solving Piecewise Polynomial Constraint Systems with Decomposition and a Subdivision-Based Solver” Boris van Sosin and Gershon Elber.
Solving the subsystems (vertices in $H_{plan}$)

• For each subsystem, all the variables that already have solutions are applied to the constraints.

• For zero dimensional solutions, the constraints multivariates are reduced to iso-parametric sub-multivariate.

• For univariate solutions, symbolic composition of the univariate solution B-spline into the constraint multivariates is performed.

source: “Solving Piecewise Polynomial Constraint Systems with Decomposition and a Subdivision-Based Solver” Boris van Sosin and Gershon Elber.
Symbolic composition of the univariate solution

• If the variable vector of the problem is $t = (t_0, t_1, ..., t_{k-1})$, and we have univariate solution for $t_0, ... t_{l-1}$, parameterized as $t_0(v), ..., t_{l-1}(v)$, the constraint $M_i$ undergoes the composition:

$$M_i(\tau(v), t_1, ..., t_{k-1}) = M_{i, \text{comp}}$$
Results and examples
Example – 2D point-and-bar problem

- Well-constrained system

- Algebraic representation:

\[
M_{\text{zero}} = \begin{cases} 
|A - P_0|^2 - c_0^2 = 0, & c_0 = 0.9 \\
|A - P_1|^2 - c_1^2 = 0, & c_1 = 2.5 \\
|A - B|^2 - c_2^2 = 0, & c_2 = 2 \\
|B - P_1|^2 - c_3^2 = 0, & c_3 = 0.9 
\end{cases}
\]

source: “Solving Piecewise Polynomial Constraint Systems with Decomposition and a Subdivision-Based Solver” Boris van Sosin and Gershon Elber.
2D point-and-bar problem

$$M^{\text{zero}} = \begin{cases} |A - P_0|^2 - c_0^2 = 0, & c_0 = 0.9 \\ |A - P_1|^2 - c_1^2 = 0, & c_1 = 2.5 \\ |A - B|^2 - c_2^2 = 0, & c_2 = 2 \\ |B - P_1|^2 - c_3^2 = 0, & c_3 = 0.9 \end{cases}$$

- First the subsystem \( \{c_0, c_1\} \) is solved and finds \( A_x, A_y \)

\[
\begin{align*}
\{c_0, c_1\} & \\
(A_x = -1.36, A_y = 0.825) & \quad (A_x = -1.36, A_y = 0.825)
\end{align*}
\]

source: “Solving Piecewise Polynomial Constraint Systems with Decomposition and a Subdivision-Based Solver” Boris van Sosin and Gershon Elber.
2D point-and-bar problem

\[ M^{\text{zero}} = \begin{cases} 
|A - P_0|^2 - c_0^2 = 0, & c_0 = 0.9 \\
|A - P_1|^2 - c_1^2 = 0, & c_1 = 2.5 \\
|A - B|^2 - c_2^2 = 0, & c_2 = 2 \\
|B - P_1|^2 - c_3^2 = 0, & c_3 = 0.9
\end{cases} \]

- Next, the subsystem \( \{c_2, c_3\} \) is solved and finds \( B_x, B_y \)

(\( A_x = -1.36, A_y = 0.825 \))

(\( A_x = -1.36, A_y = -0.825 \))

source: “Solving Piecewise Polynomial Constraint Systems with Decomposition and a Subdivision-Based Solver” Boris van Sosin and Gershon Elber.
2D point-and-bar problem

\[ M^{\text{zero}} = \begin{cases} 
|A - P_0|^2 - c_0^2 = 0, & c_0 = 0.9 \\
|A - P_1|^2 - c_1^2 = 0, & c_1 = 2.5 \\
|A - B|^2 - c_2^2 = 0, & c_2 = 2 \\
|B - P_1|^2 - c_3^2 = 0, & c_3 = 0.9 
\end{cases} \]

- Next, the subsystem \( \{c_2, c_3\} \) is solved and finds \( B_x, B_y \)

\[(A_x = -1.36, A_y = 0.825)\]
\[(A_x = -1.36, A_y = -0.825)\]

source: “Solving Piecewise Polynomial Constraint Systems with Decomposition and a Subdivision-Based Solver” Boris van Sosin and Gershon Elber.
2D point-and-bar problem

\[ M^{\text{zero}} = \begin{cases} 
|A - P_0|^2 - c_0^2 = 0, & c_0 = 0.9 \\
|A - P_1|^2 - c_1^2 = 0, & c_1 = 2.5 \\
|A - B|^2 - c_2^2 = 0, & c_2 = 2 \\
|B - P_1|^2 - c_3^2 = 0, & c_3 = 0.9 
\end{cases} \]

- Adding inequality constraints to the system:

\[ M^{\text{ineq}} = \begin{cases} 
A_y \geq 0 \\
z((B - p_1) \times (A - P_1)) \geq 0 
\end{cases} \]

source: “Solving Piecewise Polynomial Constraint Systems with Decomposition and a Subdivision-Based Solver” Boris van Sosin and Gershon Elber.
2D point-and-bar problem

\[ M_{\text{zero}} = \begin{cases} |A - P_0|^2 - c_0^2 = 0, & c_0 = 0.9 \\ |A - P_1|^2 - c_1^2 = 0, & c_1 = 2.5 \\ |A - B|^2 - c_2^2 = 0, & c_2 = 2 \\ |B - P_1|^2 - c_3^2 = 0, & c_3 = 0.9 \end{cases} \]

\[ M_{\text{ineq}} = \begin{cases} A_y \geq 0 \\ z((B - p_1) \times (A - P_1)) \geq 0 \end{cases} \]

- The constraint \( z((B - p_1) \times (A - P_1)) \geq 0 \) is added to the subsystem \( \{c_2, c_3\} \)

source: “Solving Piecewise Polynomial Constraint Systems with Decomposition and a Subdivision-Based Solver” Boris van Sosin and Gershon Elber.
2D point-and-bar problem

\[ M^{\text{zero}} = \begin{cases} \|A - P_0\|^2 - c_0^2 = 0, & c_0 = 0.9 \\ \|A - P_1\|^2 - c_1^2 = 0, & c_1 = 2.5 \\ \|A - B\|^2 - c_2^2 = 0, & c_2 = 2 \\ \|B - P_1\|^2 - c_3^2 = 0, & c_3 = 0.9 \end{cases} \]

\[ M^{\text{ineq}} = \begin{cases} A_y \geq 0 \\ z((B - p_1) \times (A - P_1)) \geq 0 \end{cases} \]

- The constraint \( A_y \geq 0 \) is added to the system \( \{c_0, c_1\} \)

source: “Solving Piecewise Polynomial Constraint Systems with Decomposition and a Subdivision-Based Solver” Boris van Sosin and Gershon Elber.
4-bar linkage problem

- under-constrained system
- same inequality constraints:

\[ M_{ineq} = \begin{cases} 
A_y \geq 0 \\
z((B - p_1) \times (A - P_1)) \geq 0
\end{cases} \]

source: “Solving Piecewise Polynomial Constraint Systems with Decomposition and a Subdivision-Based Solver” Boris van Sosin and Gershon Elber.
The Jansen’s linkage problem

- Decomposed into 6 subsystems.
- Found 38 solutions without inequalities.
- Added inequalities:
  \[
  \begin{align*}
  (P_0 - B) \times (A - B) & \geq 0 \\
  (A - D) \times (P_0 - D) & \geq 0 \\
  (C - B) \times (P_0 - B) & \geq 0 \\
  (E - D) \times (F - D) & \geq 0 \\
  (D - E) \times (C - E) & \geq 0
  \end{align*}
  \]

source: “Solving Piecewise Polynomial Constraint Systems with Decomposition and a Subdivision-Based Solver” Boris van Sosin and Gershon Elber.
The kinematic over splines problem

- Fixed point $P_0$, curves $c_1, c_2$, surface $S$.
- The variables:
  - $A = c_1(t_A)$
  - $B = (B_x, B_y)$
  - $C = c_2(t_c)$
  - $D = c_2(t_d)$
  - $E = S(t_E, v_E)$
- The constraints:

$$
|B - P_0|^2 - L_1^2 = 0 \\
|c_1(t_A) - B|^2 - L_2^2 = 0 \\
|c_2(t_c) - B|^2 - L_3^2 = 0 \\
|c_2(t_d) - B|^2 - L_4^2 = 0 \\
|c_2(t_c) - S(u_E, v_E)|^2 - L_5^2 = 0 \\
|c_2(t_d) - S(u_E, v_E)|^2 - L_6^2 = 0
$$

- Six constraints, 7 unknowns.
- Four disjoint solutions.

source: “Solving Piecewise Polynomial Constraint Systems with Decomposition and a Subdivision-Based Solver” Boris van Sosin and Gershon Elber.
Inverse kinematics problem

- The tea pot is represented as $S(u, v)$
- The letter “C” is represented as a planar curve $c(t)$
- $c(t)$ is embedded in $S(u, v)$ by $c_s(t) = S(c(t))$
- Constraints:
  \[
  \begin{align*}
  |c_s(t) - B|^2 &= L_3^2 \\
  \langle S_u(c(T)), B - c_s(t) \rangle &= 0 \\
  \langle S_v(c(T)), B - c_s(t) \rangle &= 0
  \end{align*}
  \]
- $L_i$ are constants, $S_u = \frac{\partial S}{\partial u}, S_v = \frac{\partial S}{\partial v}$
- 6 constraints and 7 unknowns.
- Since the planar letter “C” is a quadratic curve, and the body of the teapot is a bi-cubic surface, the maximal polynomial orders of the first three constraints are 25, 23, and 23.

source: “Solving Piecewise Polynomial Constraint Systems with Decomposition and a Subdivision-Based Solver” Boris van Sosin and Gershon Elber.
Flecnoval curves

- A flecnodal curve is defined as a locus of all the points at which the surface has a third-order contact with a ray.
- $S(u,v)$ is a $C^3$ continuous surface.
- $N(u,v)$ the unnormalized normal at $u,v$.
- The flecnodal curves of the surface are the solution to the following system:

$$\begin{align*}
\langle a^2 S_{uu}(u,v) + 2ab S_{uv}(u,v) + b^2 S_{vv}(u,v), n(u,v) \rangle &= 0 \\
\langle a^3 S_{uuu}(u,v) + 3a^2 b S_{uuv}(u,v) + 3ab^2 S_{uvv}(u,v) + b^3 S_{vvv}(u,v), n(u,v) \rangle &= 0 \\
& \quad a^2 + b^2 - 1 = 0
\end{align*}$$

- The constraint $a^2 + b^2 - 1 = 0$ was solved first.
- The solution propagated into the first two constraints allowing them to be solved more efficiently.

source: “Solving Piecewise Polynomial Constraint Systems with Decomposition and a Subdivision-Based Solver” Boris van Sosin and Gershon Elber.
Table 1
The performance of the framework with decomposition, compared to the subdivision-based solver without decomposition.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Time (seconds)</th>
<th>Speedup factor</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No decomposition</td>
<td>With decomposition</td>
</tr>
<tr>
<td>2D point-and-bar</td>
<td>0.000474</td>
<td>0.000449</td>
</tr>
<tr>
<td>2D point-and-bar, with inequity</td>
<td>0.00034</td>
<td>0.0003</td>
</tr>
<tr>
<td>Four-bar</td>
<td>0.27</td>
<td>0.21</td>
</tr>
<tr>
<td>Four-bar, with inequality</td>
<td>0.09</td>
<td>0.074</td>
</tr>
<tr>
<td>Jansen’s linkage</td>
<td>19730</td>
<td>61.2</td>
</tr>
<tr>
<td>Jansen’s linkage, with inequality</td>
<td>9690.4</td>
<td>2.7</td>
</tr>
<tr>
<td>kinematic over splines</td>
<td>4544.2</td>
<td>7.6</td>
</tr>
<tr>
<td>Inverse kinematics</td>
<td>84.9</td>
<td>3.5</td>
</tr>
<tr>
<td>Flecnodal curves</td>
<td>0.70</td>
<td>0.36</td>
</tr>
</tbody>
</table>

Source: “Solving Piecewise Polynomial Constraint Systems with Decomposition and a Subdivision-Based Solver” Boris van Sosin and Gershon Elber.
“Precise contact motion planning for deformable planar curved shapes” Yong-Joon Kim, Gershon Elber, Myung-Soo Kim.
The problem

• Given a $C^1$ continuous robot $\phi_t(C(u))$ in parametric or implicit form
• And given parametric obstacles $D(v)$, in the plane.
• $\phi_t$ is a one-parameter smooth freeform deformation of $C(u)$
• $\phi_t(C(u))$ has:
  • two translation DOF $(x, y)$
  • one rotation DOF $\theta$
  • One DOF $t$ provides shape control over $\phi_t$
• Contact motion planning for the robot.

source: “Precise contact motion planning for deformable planar curved shapes” Yong-Joon Kim, Gershon Elber, Myung-Soo Kim.
Deformable robots representation

- Two ways of prescribing the robot (a planar deforming shape).
  - Parametric:
    - $C_i(u_i), i = (1,2), u_i \in [0,1]$ two regular, smooth, parametric curves.
    - $\phi_t(C(u)) = (1-t)C_1(u) + tC_2(u)$ (Alternatively, $\phi_t(C(u))$ can be any $S(u,t)$)
  - Implicit:
    - $C_i(x,y), = 0 i = (1,2)$ two smooth implicit curves
    - $\phi_t(C(x,y)) = (1-t)C_1((x,y)) + tC_2((x,y))$ (Alternatively, can be any $v(x,y,t) = 0$)
  - The advantage is that it can change the topology

source: “Precise contact motion planning for deformable planar curved shapes” Yong-Joon Kim, Gershon Elber, Myung-Soo Kim.
Algebraic condition for k-contact motion

• \( (\phi_t(C(u))_x, \phi_t(C(u))_y) \) \( C^1 \) continuous regular parametric curve.

• \( D(v) = (D(v)_x, D(v)_y) \) stationary obstacle.

• Rigid transformation of \( \phi_t(c(u)) \): \( T[\phi_t(c(u))] = R_{\theta}[\phi_t(c(u))] + (x, y) \)

• The conditions for K-contact between \( T[\phi_t(c(u_i))] \) and \( D(v_i) \) i=1,...k:
  • \( 0 = R_{\theta}[\phi_t(c(u))]_x + x - D(v_i)_x \)
  • \( 0 = R_{\theta}[\phi_t(c(u))]_y + y - D(v_i)_y \)
  • \( 0 = F_i(u_i, v_i, \theta, t) = R_{\theta}[\phi_t(c(u))] \times D'(v_i) \)

source: “Precise contact motion planning for deformable planar curved shapes” Yong-Joon Kim, Gershon Elber, Myung-Soo Kim.
Algebraic condition for k-contact motion

- Isolating x and y:
  - \( x = G_i(u_i, v_i, \theta, t) = D(v_i)_x - R_\theta[\phi_t(c(u))]_x \)
  - \( y = G_i(u_i, v_i, \theta, t) = D(v_i)_y - R_\theta[\phi_t(c(u))]_y \)

- We can write:
  - \( 0 = G_1(u_i, v_i, \theta, t) - G_i(u_i, v_i, \theta, t) \) for \( 2 \leq i \leq k \)
  - \( 0 = H_1(u_i, v_i, \theta, t) - H_i(u_i, v_i, \theta, t) \) for \( 2 \leq i \leq k \)
  - \( 0 = F_i(u_i, v_i, \theta, t) \) for \( 1 \leq i \leq k \)
  - Which is \( 3K-2 \) constraints in \( 2K+2 \) variables \((u_i, v_i, \theta, t)\)

source: “Precise contact motion planning for deformable planar curved shapes” Yong-Joon Kim, Gershon Elber, Myung-Soo Kim.
Degrees of freedom

• For $k=2$ we have 4 constraints in 6 variables. If we fix $t$ (i.e., rigid robot) we have 4 constraints and 5 variables. Therefore 1 degree of freedom.

• For $k=3$ we have 7 constraints in 8 variables. Therefore, 1 degree of freedom.

source: “Precise contact motion planning for deformable planar curved shapes” Yong-Joon Kim, Gershon Elber, Myung-Soo Kim.
The K-contact motion graph

• Consists of 2 or 3 contact motions.
• Fixing \( t = t^* \)

source: “Precise contact motion planning for deformable planar curved shapes” Yong-Joon Kim, Gershon Elber, Myung-Soo Kim.
Disconnected components

source: “Precise contact motion planning for deformable planar curved shapes” Yong-Joon Kim, Gershon Elber, Myung-Soo Kim.
3 – contact graph

source: “Precise contact motion planning for deformable planar curved shapes” Yong-Joon Kim, Gershon Elber, Myung-Soo Kim.