INPUT-OUTPUT EQUATION FOR PLANAR FOUR-BAR LINKAGES

M. John D, Hayes[1] Manfred L. Husty[2] Martin Pfurner[2]

[1]Department of Mechanical and Aerospace Engineering
Carleton University, Ottawa, ON, Canada,

[2]University of Innsbruck, Unit Geometry and CAD,
Innsbruck, Austria

ARK 2018
16th International Symposium on
Advances in Robot Kinematics
July 1 - 5, 2018
Bologna, Italy

Introduction

- A new method for deriving the I-O equations of planar $4R$ mechanisms was needed for the approximate synthesis of function generators for the following reasons.
  - It has been observed that as the cardinality of the prescribed discrete I-O data set increases the linkages that minimise the 2-norm of the design and structural errors tend to converge to the same linkage.
    - The design error indicates the error residual incurred by a specific linkage regarding the verification of the synthesis equations.
    - The structural error is the difference between the prescribed linkage output angle and the generated output angle for a prescribed input angle value.
  - The important implication of this observation is that the minimisation of the Euclidean norm of the structural error can be accomplished indirectly via the minimisation of the corresponding norm of the design error, provided that a suitably large number of I-O pairs is prescribed.

Introduction

- A new method for deriving the I-O equations of planar $4R$ mechanisms was needed for the approximate synthesis of function generators for the following reasons.
  - It has been observed that as the cardinality of the prescribed discrete I-O data set increases the linkages that minimise the 2-norm of the design and structural errors tend to converge to the same linkage.
    - The design error indicates the error residual incurred by a specific linkage regarding the verification of the synthesis equations.
    - The structural error is the difference between the prescribed linkage output angle and the generated output angle for a prescribed input angle value.
  - The important implication of this observation is that the minimisation of the Euclidean norm of the structural error can be accomplished indirectly via the minimisation of the corresponding norm of the design error, provided that a suitably large number of I-O pairs is prescribed.

Introduction

- A new method for deriving the I-O equations of planar 4R mechanisms was needed for the approximate synthesis of function generators for the following reasons.
  - It has been observed that as the cardinality of the prescribed discrete I-O data set increases the linkages that minimise the 2-norm of the design and structural errors tend to converge to the same linkage.
    - The design error indicates the error residual incurred by a specific linkage regarding the verification of the synthesis equations.
    - The structural error is the difference between the prescribed linkage output angle and the generated output angle for a prescribed input angle value.
  - The important implication of this observation is that the minimisation of the Euclidean norm of the structural error can be accomplished indirectly via the minimisation of the corresponding norm of the design error, provided that a suitably large number of I-O pairs is prescribed.

Introduction

- A new method for deriving the I-O equations of planar 4*R* mechanisms was needed for the approximate synthesis of function generators for the following reasons.
  - It has been observed that as the cardinality of the prescribed discrete I-O data set increases the linkages that minimise the 2-norm of the design and structural errors tend to converge to the same linkage.
    - The design error indicates the error residual incurred by a specific linkage regarding the verification of the synthesis equations.
    - The structural error is the difference between the prescribed linkage output angle and the generated output angle for a prescribed input angle value.
  - The important implication of this observation is that the minimisation of the Euclidean norm of the structural error can be accomplished indirectly via the minimisation of the corresponding norm of the design error, provided that a suitably large number of I-O pairs is prescribed.

Introduction

- A new method for deriving the I-O equations of planar $4R$ mechanisms was needed for the approximate synthesis of function generators for the following reasons.
  - It has been observed that as the cardinality of the prescribed discrete I-O data set increases the linkages that minimise the 2-norm of the design and structural errors tend to converge to the same linkage.
    - The design error indicates the error residual incurred by a specific linkage regarding the verification of the synthesis equations.
    - The structural error is the difference between the prescribed linkage output angle and the generated output angle for a prescribed input angle value.
  - The important implication of this observation is that the minimisation of the Euclidean norm of the structural error can be accomplished indirectly via the minimisation of the corresponding norm of the design error, provided that a suitably large number of I-O pairs is prescribed.

Introduction Continued

- Minimisation of the Euclidean norm of the design error leads to a linear least-squares problem whose solution can be obtained directly, while the minimisation of the same norm of the structural error leads to a nonlinear least-squares problem, requiring an iterative solution.
- All this has suggested the need to develop *Continuous Approximate Synthesis* (CAS) as an alternative to discrete approximate synthesis.
- CAS involves integrating the synthesis equations between the bounds of the minimum and maximum input angles.
- The resulting need for numerical integrators suggests representing the transcendental Freudenstein equations as algebraic ones using the tangent of the half angle substitutions.

Introduction Continued

- Minimisation of the Euclidean norm of the design error leads to a linear least-squares problem whose solution can be obtained directly, while the minimisation of the same norm of the structural error leads to a nonlinear least-squares problem, requiring an iterative solution.
- All this has suggested the need to develop *Continuous Approximate Synthesis* (CAS) as an alternative to discrete approximate synthesis.
- CAS involves integrating the synthesis equations between the bounds of the minimum and maximum input angles.
- The resulting need for numerical integrators suggests representing the transcendental Freudenstein equations as algebraic ones using the tangent of the half angle substitutions.

Introduction Continued

- Minimisation of the Euclidean norm of the design error leads to a linear least-squares problem whose solution can be obtained directly, while the minimisation of the same norm of the structural error leads to a nonlinear least-squares problem, requiring an iterative solution.
- All this has suggested the need to develop *Continuous Approximate Synthesis* (CAS) as an alternative to discrete approximate synthesis.
- CAS involves integrating the synthesis equations between the bounds of the minimum and maximum input angles.
- The resulting need for numerical integrators suggests representing the transcendental Freudenstein equations as algebraic ones using the tangent of the half angle substitutions.

Introduction Continued

- Minimisation of the Euclidean norm of the design error leads to a linear least-squares problem whose solution can be obtained directly, while the minimisation of the same norm of the structural error leads to a nonlinear least-squares problem, requiring an iterative solution.
- All this has suggested the need to develop *Continuous Approximate Synthesis* (CAS) as an alternative to discrete approximate synthesis.
- CAS involves integrating the synthesis equations between the bounds of the minimum and maximum input angles.
- The resulting need for numerical integrators suggests representing the transcendental Freudenstein equations as algebraic ones using the tangent of the half angle substitutions.

Introduction Continued

- We decided to establish a concomitant method for deriving the Freudenstein equations that can be applied to any kinematic architecture: planar; spherical; or spatial.
- We developed the method for planar $4R$ function generators so that we could compare our results to the tangent half-angel substitution in Freudenstein's equation for confirmation, which is the material presented in this paper.
- In the paper exact synthesis is performed with multiple sets of three I-O pairs leading to similar mechanisms, as is reasonably expected.
- In this talk discrete approximate synthesis results for 5, 10, 50, 100, and 500 I-O pairs are presented and discussed.

Introduction Continued

- We decided to establish a concomitant method for deriving the Freudenstein equations that can be applied to any kinematic architecture: planar; spherical; or spatial.
- We developed the method for planar $4R$ function generators so that we could compare our results to the tangent half-angel substitution in Freudenstein's equation for confirmation, which is the material presented in this paper.
- In the paper exact synthesis is performed with multiple sets of three I-O pairs leading to similar mechanisms, as is reasonably expected.
- In this talk discrete approximate synthesis results for 5, 10, 50, 100, and 500 I-O pairs are presented and discussed.

Introduction Continued

- We decided to establish a concomitant method for deriving the Freudenstein equations that can be applied to any kinematic architecture: planar; spherical; or spatial.
- We developed the method for planar $4R$ function generators so that we could compare our results to the tangent half-angel substitution in Freudenstein's equation for confirmation, which is the material presented in this paper.
- In the paper exact synthesis is performed with multiple sets of three I-O pairs leading to similar mechanisms, as is reasonably expected.
- In this talk discrete approximate synthesis results for 5, 10, 50, 100, and 500 I-O pairs are presented and discussed.

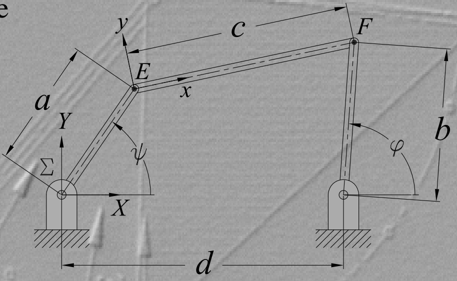
Introduction Continued

- We decided to establish a concomitant method for deriving the Freudenstein equations that can be applied to any kinematic architecture: planar; spherical; or spatial.
- We developed the method for planar $4R$ function generators so that we could compare our results to the tangent half-angel substitution in Freudenstein's equation for confirmation, which is the material presented in this paper.
- In the paper exact synthesis is performed with multiple sets of three I-O pairs leading to similar mechanisms, as is reasonably expected.
- In this talk discrete approximate synthesis results for 5, 10, 50, 100, and 500 I-O pairs are presented and discussed.

Freudenstein Planar 4R Function Generator Equation

- The Freudenstein equation relates the input to the output angles of a planar 4R four-bar mechanism:

$$k_1 + k_2 \cos(\varphi_i) - k_3 \cos(\psi_i) = \cos(\psi_i - \varphi_i).$$

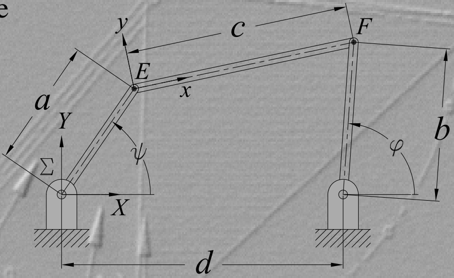


- We arbitrarily select ψ as the input angle and φ as the output angle.
  - The equation is linear in the k_i Freudenstein parmeters, which are defined in terms of the link length ratios as:

Freudenstein Planar 4R Function Generator Equation

- The Freudenstein equation relates the input to the output angles of a planar 4R four-bar mechanism:

$$k_1 + k_2 \cos(\varphi_i) - k_3 \cos(\psi_i) = \cos(\psi_i - \varphi_i).$$

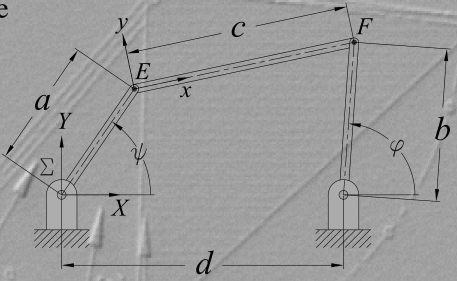


- We arbitrarily select ψ as the input angle and φ as the output angle.
  - The equation is linear in the k_i Freudenstein parmeters, which are defined in terms of the link length ratios as:

Freudenstein Planar 4R Function Generator Equation

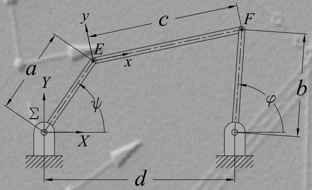
- The Freudenstein equation relates the input to the output angles of a planar 4R four-bar mechanism:

$$k_1 + k_2\cos(\varphi_i) - k_3\cos(\psi_i) = \cos(\psi_i - \varphi_i).$$



- We arbitrarily select ψ as the input angle and φ as the output angle.
  - The equation is linear in the k_i Freudenstein parmeters, which are defined in terms of the link length ratios as:

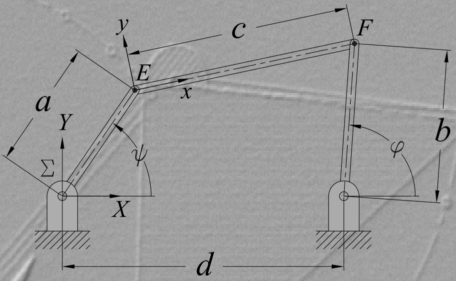
Freudenstein Planar 4R Function Generator Equation



$$
\left.\begin{aligned}
k_1 &\equiv \frac{(a^2+b^2+d^2-c^2)}{2ab},\\
k_2 &\equiv \frac{d}{a},\\
k_3 &\equiv \frac{d}{b}.
\end{aligned}\right\} \Leftrightarrow \left\{\begin{aligned}
d &= 1,\\
a &= \frac{1}{k_2},\\
b &= \frac{1}{k_3},\\
c &= (a^2+b^2+d^2-2abk_1)^{1/2}.
\end{aligned}\right.
$$

Algebraic I-O Equation Derivation

- Let Σ be a non moving Cartesian coordinate system with coordinates X and Y whose origin is located at the centre of the ground fixed link R-pair with length a.

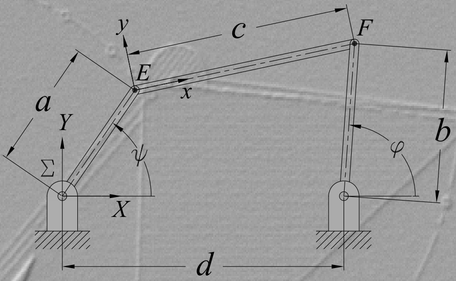


- Let E be a coordinate system that moves with the coupler of length c whose origin is at the centre of the distal R-pair of link a, having basis directions x and y.
- The displacement constraints for the origin of E can be expressed as

$$\begin{aligned} X - a\cos\psi &= 0, \\ Y - a\sin\psi &= 0. \end{aligned} \qquad (1)$$

Algebraic I-O Equation Derivation

- Let Σ be a non moving Cartesian coordinate system with coordinates X and Y whose origin is located at the centre of the ground fixed link R-pair with length a.

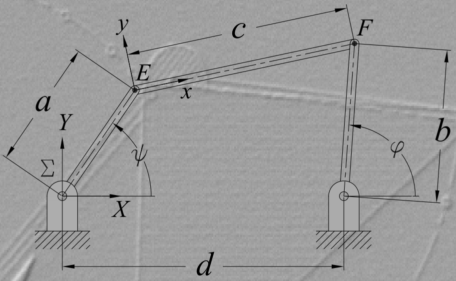


- Let E be a coordinate system that moves with the coupler of length c whose origin is at the centre of the distal R-pair of link a, having basis directions x and y.
- The displacement constraints for the origin of E can be expressed as

$$
\begin{aligned}
X - a\cos\psi &= 0, \\
Y - a\sin\psi &= 0.
\end{aligned}
\quad (1)
$$

Algebraic I-O Equation Derivation

- Let Σ be a non moving Cartesian coordinate system with coordinates X and Y whose origin is located at the centre of the ground fixed link R-pair with length a.



- Let E be a coordinate system that moves with the coupler of length c whose origin is at the centre of the distal R-pair of link a, having basis directions x and y.

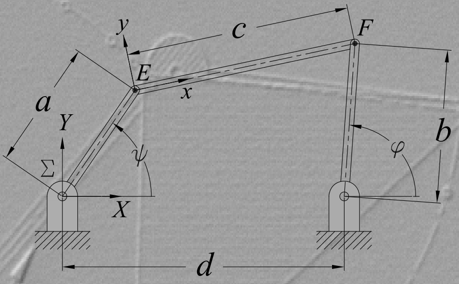
- The displacement constraints for the origin of E can be expressed as

$$\begin{aligned} X - a\cos\psi &= 0, \\ Y - a\sin\psi &= 0. \end{aligned} \tag{1}$$

Algebraic I-O Equation Derivation

- The displacement constraints for point F, located at the centre of the distal R-pair on the output link with length b are

$$\begin{aligned} X - d - b\cos\varphi &= 0, \\ Y - b\sin\varphi &= 0. \end{aligned} \quad (2)$$



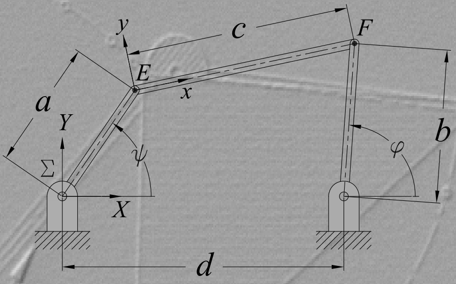
- Any displacement in Euclidean space, E_3, can be mapped in terms of the coordinates of a 7-dimensional projective image space using the transformation

$$\mathbf{T} = \begin{bmatrix} x_0^2+x_1^2+x_2^2+x_3^2 & 0 & 0 & 0 \\ 2(-x_0y_1+x_1y_0-x_2y_3+x_3y_2) & x_0^2+x_1^2-x_2^2-x_3^2 & 2(x_1x_2-x_0x_3) & 2(x_1x_3+x_0x_2) \\ 2(-x_0y_2+x_1y_3+x_2y_0-x_3y_1) & 2(x_1x_2+x_0x_3) & x_0^2-x_1^2+x_2^2-x_3^2 & 2(x_2x_3-x_0x_1) \\ 2(-x_0y_3-x_1y_2+x_2y_1+x_3y_0) & 2(x_1x_3-x_0x_2) & 2(x_2x_3+x_0x_1) & x_0^2-x_1^2-x_2^2+x_3^2 \end{bmatrix}.$$

Algebraic I-O Equation Derivation

- The displacement constraints for point F, located at the centre of the distal R-pair on the output link with length b are

$$\begin{aligned} X - d - b\cos\varphi &= 0, \\ Y - b\sin\varphi &= 0. \end{aligned} \qquad (2)$$



- Any displacement in Euclidean space, E_3, can be mapped in terms of the coordinates of a 7-dimensional projective image space using the transformation

$$\mathbf{T} = \begin{bmatrix} x_0^2+x_1^2+x_2^2+x_3^2 & 0 & 0 & 0 \\ 2(-x_0y_1+x_1y_0-x_2y_3+x_3y_2) & x_0^2+x_1^2-x_2^2-x_3^2 & 2(x_1x_2-x_0x_3) & 2(x_1x_3+x_0x_2) \\ 2(-x_0y_2+x_1y_3+x_2y_0-x_3y_1) & 2(x_1x_2+x_0x_3) & x_0^2-x_1^2+x_2^2-x_3^2 & 2(x_2x_3-x_0x_1) \\ 2(-x_0y_3-x_1y_2+x_2y_1+x_3y_0) & 2(x_1x_3-x_0x_2) & 2(x_2x_3+x_0x_1) & x_0^2-x_1^2-x_2^2+x_3^2 \end{bmatrix}.$$

Algebraic I-O Equation Derivation

- **T** transforms the coordinates of any point described in a moving 3D coordinate system E to the coordinates of the same point in a relatively fixed 3D coordinate system Σ in terms of the coordinates of a point on the Study quadric, S_6^2:

$$S_6^2: \quad x_0y_0 + x_1y_1 + x_2y_2 + x_3y_3 \quad = \quad 0.$$

- In order for a point in the image space to represent a real displacement, and therefore to be located on S_6^2, the non-zero condition of $x_0^2 + x_1^2 + x_2^2 + x_3^2 \neq 0$ must be satisfied.
- The transformation matrix **T** simplifies considerably when we consider displacements that are restricted to the plane.
- Three degrees of freedom are lost and hence four Study parameters vanish.

Algebraic I-O Equation Derivation

- **T** transforms the coordinates of any point described in a moving $3D$ coordinate system E to the coordinates of the same point in a relatively fixed $3D$ coordinate system Σ in terms of the coordinates of a point on the Study quadric, S_6^2:

$$S_6^2: \quad x_0y_0 + x_1y_1 + x_2y_2 + x_3y_3 \quad = \quad 0.$$

- In order for a point in the image space to represent a real displacement, and therefore to be located on S_6^2, the non-zero condition of $x_0^2 + x_1^2 + x_2^2 + x_3^2 \neq 0$ must be satisfied.
- The transformation matrix **T** simplifies considerably when we consider displacements that are restricted to the plane.
- Three degrees of freedom are lost and hence four Study parameters vanish.

Algebraic I-O Equation Derivation

- **T** transforms the coordinates of any point described in a moving $3D$ coordinate system E to the coordinates of the same point in a relatively fixed $3D$ coordinate system Σ in terms of the coordinates of a point on the Study quadric, S_6^2:

$$S_6^2: \quad x_0y_0 + x_1y_1 + x_2y_2 + x_3y_3 \quad = \quad 0.$$

- In order for a point in the image space to represent a real displacement, and therefore to be located on S_6^2, the non-zero condition of $x_0^2 + x_1^2 + x_2^2 + x_3^2 \neq 0$ must be satisfied.
- The transformation matrix **T** simplifies considerably when we consider displacements that are restricted to the plane.
- Three degrees of freedom are lost and hence four Study parameters vanish.

Algebraic I-O Equation Derivation

- **T** transforms the coordinates of any point described in a moving $3D$ coordinate system E to the coordinates of the same point in a relatively fixed $3D$ coordinate system Σ in terms of the coordinates of a point on the Study quadric, S_6^2:

$$S_6^2 : \quad x_0 y_0 + x_1 y_1 + x_2 y_2 + x_3 y_3 \quad = \quad 0.$$

- In order for a point in the image space to represent a real displacement, and therefore to be located on S_6^2, the non-zero condition of $x_0^2 + x_1^2 + x_2^2 + x_3^2 \neq 0$ must be satisfied.
- The transformation matrix **T** simplifies considerably when we consider displacements that are restricted to the plane.
- Three degrees of freedom are lost and hence four Study parameters vanish.

Algebraic I-O Equation Derivation

- We arbitrarily select the plane $Z = 0$ to contain our displacements.
- Since E and Σ are assumed to be initially coincident, this means

$$\begin{bmatrix} W \\ X \\ Y \\ 0 \end{bmatrix} = \mathbf{T} \begin{bmatrix} w \\ x \\ y \\ 0 \end{bmatrix}. \tag{3}$$

- This leaves us with the four soma coordinates

$$(x_0 : x_3 : y_1 : y_2). \tag{4}$$

- The non-zero condition reduces to $x_0^2 + x_3^2 \neq 0$.

Algebraic I-O Equation Derivation

- We arbitrarily select the plane $Z = 0$ to contain our displacements.
- Since E and Σ are assumed to be initially coincident, this means

$$\begin{bmatrix} W \\ X \\ Y \\ 0 \end{bmatrix} = \mathbf{T} \begin{bmatrix} w \\ x \\ y \\ 0 \end{bmatrix}. \tag{3}$$

- This leaves us with the four soma coordinates

$$(x_0 : x_3 : y_1 : y_2). \tag{4}$$

- The non-zero condition reduces to $x_0^2 + x_3^2 \neq 0$.

Algebraic I-O Equation Derivation

- We arbitrarily select the plane $Z = 0$ to contain our displacements.
- Since E and Σ are assumed to be initially coincident, this means

$$\begin{bmatrix} W \\ X \\ Y \\ 0 \end{bmatrix} = \mathbf{T} \begin{bmatrix} w \\ x \\ y \\ 0 \end{bmatrix}. \tag{3}$$

- This leaves us with the four soma coordinates

$$(x_0 : x_3 : y_1 : y_2). \tag{4}$$

- The non-zero condition reduces to $x_0^2 + x_3^2 \neq 0$.

Algebraic I-O Equation Derivation

- We arbitrarily select the plane $Z = 0$ to contain our displacements.
- Since E and Σ are assumed to be initially coincident, this means

$$\begin{bmatrix} W \\ X \\ Y \\ 0 \end{bmatrix} = \mathbf{T} \begin{bmatrix} w \\ x \\ y \\ 0 \end{bmatrix}. \tag{3}$$

- This leaves us with the four soma coordinates

$$(x_0 : x_3 : y_1 : y_2). \tag{4}$$

- The non-zero condition reduces to $x_0^2 + x_3^2 \neq 0$.

Algebraic I-O Equation Derivation

- We normalise the coordinates with the nonzero condition giving the planar kinematic mapping transformation

$$\mathbf{T} = \frac{1}{x_0^2+x_3^2}\begin{bmatrix} x_0^2+x_3^2 & 0 & 0 \\ 2(-x_0y_1+x_3y_2) & x_0^2-x_3^2 & -2x_0x_3 \\ -2(x_0y_2+x_3y_1) & 2x_0x_3 & x_0^2-x_3^2 \end{bmatrix}. \tag{5}$$

- We can now express a point in Σ in terms of the soma coordinates and the corresponding point coordinates in E as

$$\begin{bmatrix} 1 \\ X \\ Y \end{bmatrix} = \mathbf{T}\begin{bmatrix} 1 \\ x \\ y \end{bmatrix} = \frac{1}{x_0^2+x_3^2}\begin{bmatrix} x_0^2+x_3^2 \\ 2(-x_0y_1+x_3y_2)+(x_0^2-x_3^2)x-(2x_0x_3)y \\ -2(x_0y_2+x_3y_1)+(2x_0x_3)x+(x_0^2-x_3^2)y \end{bmatrix}.$$

Algebraic I-O Equation Derivation

- We normalise the coordinates with the nonzero condition giving the planar kinematic mapping transformation

$$\mathbf{T} = \frac{1}{x_0^2 + x_3^2}\begin{bmatrix} x_0^2 + x_3^2 & 0 & 0 \\ 2(-x_0y_1 + x_3y_2) & x_0^2 - x_3^2 & -2x_0x_3 \\ -2(x_0y_2 + x_3y_1) & 2x_0x_3 & x_0^2 - x_3^2 \end{bmatrix}. \tag{5}$$

- We can now express a point in Σ in terms of the soma coordinates and the corresponding point coordinates in E as

$$\begin{bmatrix} 1 \\ X \\ Y \end{bmatrix} = \mathbf{T}\begin{bmatrix} 1 \\ x \\ y \end{bmatrix} = \frac{1}{x_0^2 + x_3^2}\begin{bmatrix} x_0^2 + x_3^2 \\ 2(-x_0y_1 + x_3y_2) + (x_0^2 - x_3^2)x - (2x_0x_3)y \\ -2(x_0y_2 + x_3y_1) + (2x_0x_3)x + (x_0^2 - x_3^2)y \end{bmatrix}.$$

Algebraic I-O Equation Derivation

- The novelty of this approach begins with creating two Cartesian vector constraint equations containing the nonhomogeneous coordinates from Equations (1) and (2), i.e. the displacement constraints for points E and F, but substituting the values from the transformation equation for (X, Y).

- These two vector equations are:

$$\mathbf{F}_1 = \frac{1}{x_0^2+x_3^2}\begin{bmatrix} 2(-x_0y_1+x_3y_2)+(x_0^2-x_3^2)x-2x_0x_3y-(a\cos\psi)(x_0^2+x_3^2) \\ -2(x_0y_2+x_3y_1)+2x_0x_3x+(x_0^2-x_3^2)y-(a\sin\psi)(x_0^2+x_3^2) \end{bmatrix} = \mathbf{0};$$

$$\mathbf{F}_2 = \frac{1}{x_0^2+x_3^2}\begin{bmatrix} 2(-x_0y_1+x_3y_2)+(x_0^2-x_3^2)x-2x_0x_3y-(b\cos\varphi+d)(x_0^2+x_3^2) \\ -2(x_0y_2+x_3y_1)+2x_0x_3x+(x_0^2-x_3^2)y-(b\sin\varphi)(x_0^2+x_3^2) \end{bmatrix} = \mathbf{0}.$$

Algebraic I-O Equation Derivation

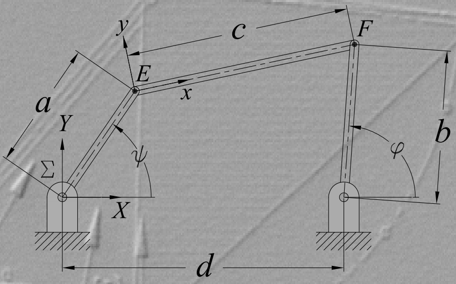
- The novelty of this approach begins with creating two Cartesian vector constraint equations containing the nonhomogeneous coordinates from Equations (1) and (2), i.e. the displacement constraints for points E and F, but substituting the values from the transformation equation for (X, Y).
- These two vector equations are:

$$\mathbf{F}_1 = \frac{1}{x_0^2+x_3^2}\begin{bmatrix} 2(-x_0y_1+x_3y_2)+(x_0^2-x_3^2)x-2x_0x_3y-(a\cos\psi)(x_0^2+x_3^2) \\ -2(x_0y_2+x_3y_1)+2x_0x_3x+(x_0^2-x_3^2)y-(a\sin\psi)(x_0^2+x_3^2) \end{bmatrix} = \mathbf{0};$$

$$\mathbf{F}_2 = \frac{1}{x_0^2+x_3^2}\begin{bmatrix} 2(-x_0y_1+x_3y_2)+(x_0^2-x_3^2)x-2x_0x_3y-(b\cos\varphi+d)(x_0^2+x_3^2) \\ -2(x_0y_2+x_3y_1)+2x_0x_3x+(x_0^2-x_3^2)y-(b\sin\varphi)(x_0^2+x_3^2) \end{bmatrix} = \mathbf{0}.$$

Algebraic I-O Equation Derivation

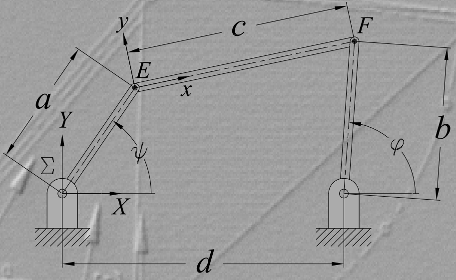
- Now we determine equations for the coupler.
- The coordinate system that moves with the coupler has its origin, point E, on the centre of the R-pair, $(x, y) = (0, 0)$.
- Point F is on the R-pair centre on the other end having coordinates $(x, y) = (c, 0)$.



- One more vector equation, $\mathbf{H}_1$ is obtained by substituting $(x, y) = (0, 0)$ in $\mathbf{F}_1$, and another, $\mathbf{H}_2$ is obtained by substituting $(x, y) = (c, 0)$ in $\mathbf{F}_2$.

Algebraic I-O Equation Derivation

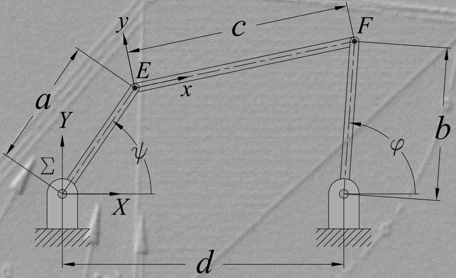
- Now we determine equations for the coupler.
- The coordinate system that moves with the coupler has its origin, point E, on the centre of the R-pair, $(x, y) = (0, 0)$.
- Point F is on the R-pair centre on the other end having coordinates $(x, y) = (c, 0)$.



- One more vector equation, $\mathbf{H}_1$ is obtained by substituting $(x, y) = (0, 0)$ in $\mathbf{F}_1$, and another, $\mathbf{H}_2$ is obtained by substituting $(x, y) = (c, 0)$ in $\mathbf{F}_2$.

Algebraic I-O Equation Derivation

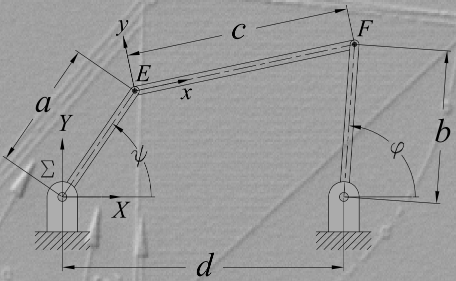
- Now we determine equations for the coupler.
- The coordinate system that moves with the coupler has its origin, point E, on the centre of the R-pair, $(x, y) = (0, 0)$.
- Point F is on the R-pair centre on the other end having coordinates $(x, y) = (c, 0)$.



- One more vector equation, $\mathbf{H}_1$ is obtained by substituting $(x, y) = (0, 0)$ in $\mathbf{F}_1$, and another, $\mathbf{H}_2$ is obtained by substituting $(x, y) = (c, 0)$ in $\mathbf{F}_2$.

Algebraic I-O Equation Derivation

- Now we determine equations for the coupler.
- The coordinate system that moves with the coupler has its origin, point E, on the centre of the R-pair, $(x, y) = (0, 0)$.
- Point F is on the R-pair centre on the other end having coordinates $(x, y) = (c, 0)$.



- One more vector equation, $\mathbf{H}_1$ is obtained by substituting $(x, y) = (0, 0)$ in $\mathbf{F}_1$, and another, $\mathbf{H}_2$ is obtained by substituting $(x, y) = (c, 0)$ in $\mathbf{F}_2$.

Algebraic I-O Equation Derivation

- Next $\mathbf{H}_1$ and $\mathbf{H}_2$, two rational expressions, are converted to factored normal form.
- This is the form where the numerator and denominator are relatively prime polynomials with integer coefficients.
- The denominators for both $\mathbf{H}_1$ and $\mathbf{H}_2$ are the nonzero condition $x_0^2 + x_3^2$, which can safely be factored out of each equation leaving the following two vector equations with polynomial elements:

$$\mathbf{H}_1 = \begin{bmatrix} -a\cos\psi(x_0^2+x_3^2)+2(-x_0y_1+x_3y_2) \\ -a\sin\psi(x_0^2+x_3^2)-2(x_0y_1+x_3y_2) \end{bmatrix} = \mathbf{0}; \tag{6}$$

$$\mathbf{H}_2 = \begin{bmatrix} -(b\cos\varphi+d)(x_0^2+x_3^2)+c(x_0^2-x_3^2)+2(-x_0y_1+x_3y_2) \\ -b\sin\varphi(x_0^2+x_3^2)+2c(x_0x_3)-2(x_0y_2+x_3y_1) \end{bmatrix} = \mathbf{0}. \tag{7}$$

Algebraic I-O Equation Derivation

- Next $\mathbf{H}_1$ and $\mathbf{H}_2$, two rational expressions, are converted to factored normal form.
- This is the form where the numerator and denominator are relatively prime polynomials with integer coefficients.
- The denominators for both $\mathbf{H}_1$ and $\mathbf{H}_2$ are the nonzero condition $x_0^2 + x_3^2$, which can safely be factored out of each equation leaving the following two vector equations with polynomial elements:

$$\mathbf{H}_1 = \begin{bmatrix} -a\cos\psi(x_0^2+x_3^2)+2(-x_0y_1+x_3y_2) \\ -a\sin\psi(x_0^2+x_3^2)-2(x_0y_1+x_3y_2) \end{bmatrix} = \mathbf{0}; \tag{6}$$

$$\mathbf{H}_2 = \begin{bmatrix} -(b\cos\varphi+d)(x_0^2+x_3^2)+c(x_0^2-x_3^2)+2(-x_0y_1+x_3y_2) \\ -b\sin\varphi(x_0^2+x_3^2)+2c(x_0x_3)-2(x_0y_2+x_3y_1) \end{bmatrix} = \mathbf{0}. \tag{7}$$

Algebraic I-O Equation Derivation

- Next $\mathbf{H}_1$ and $\mathbf{H}_2$, two rational expressions, are converted to factored normal form.
- This is the form where the numerator and denominator are relatively prime polynomials with integer coefficients.
- The denominators for both $\mathbf{H}_1$ and $\mathbf{H}_2$ are the nonzero condition $x_0^2 + x_3^2$, which can safely be factored out of each equation leaving the following two vector equations with polynomial elements:

$$\mathbf{H}_1 = \begin{bmatrix} -a\cos\psi(x_0^2+x_3^2)+2(-x_0y_1+x_3y_2) \\ -a\sin\psi(x_0^2+x_3^2)-2(x_0y_1+x_3y_2) \end{bmatrix} = \mathbf{0}; \tag{6}$$

$$\mathbf{H}_2 = \begin{bmatrix} -(b\cos\varphi+d)(x_0^2+x_3^2)+c(x_0^2-x_3^2)+2(-x_0y_1+x_3y_2) \\ -b\sin\varphi(x_0^2+x_3^2)+2c(x_0x_3)-2(x_0y_2+x_3y_1) \end{bmatrix} = \mathbf{0}. \tag{7}$$

Algebraic I-O Equation Derivation

- $\mathbf{H}_1 = \mathbf{0}$ and $\mathbf{H}_2 = \mathbf{0}$ are trigonometric equations.
- We convert them to algebraic ones using the Weierstraß (tangent of the half-angle) substitutions

$$u = \tan\frac{\psi}{2}, \quad v = \tan\frac{\varphi}{2},$$

and

$$\cos\psi = \frac{1-u^2}{1+u^2}, \quad \sin\psi = \frac{2u}{1+u^2},$$
$$\cos\varphi = \frac{1-v^2}{1+v^2}, \quad \sin\varphi = \frac{2v}{1+v^2}.$$

Algebraic I-O Equation Derivation

- $\mathbf{H}_1 = \mathbf{0}$ and $\mathbf{H}_2 = \mathbf{0}$ are trigonometric equations.
- We convert them to algebraic ones using the Weierstraß (tangent of the half-angle) substitutions

$$u = \tan\frac{\psi}{2}, \quad v = \tan\frac{\varphi}{2},$$

and

$$\begin{aligned} \cos\psi &= \frac{1-u^2}{1+u^2}, & \sin\psi &= \frac{2u}{1+u^2}, \\ \cos\varphi &= \frac{1-v^2}{1+v^2}, & \sin\varphi &= \frac{2v}{1+v^2}. \end{aligned}$$

Algebraic I-O Equation Derivation

- We make the Weierstraß substitutions and convert $\mathbf{H}_1$ and $\mathbf{H}_2$ to factored normal form.
- The denominators are $u^2 + 1$ and $v^2 + 1$ which can safely be factored out because they are always non-vanishing.
- The resulting four algebraic equations are:

$$\mathbf{K}_1 = \left[\begin{array}{c} (au^2-a)(x_0^2+x_3^2)+2u^2(-x_0y_1+x_3y_2)+2(-x_0y_2+x_3y_1) \\ -2au(x_0^2+x_3^2)-2(1+u^2)(-x_0y_2+x_3y_1) \end{array}\right] = \mathbf{0}; \quad (8)$$

$$\mathbf{K}_2 = \left[\begin{array}{c} (v^2(b-d)+b-d)(x_0^2+x_3^2)+(cv^2+c)(x_0^2-x_3^2)+ \\ 2(1+v^2)(-x_0y_1+x_3y_2) \\ 2(v^2+1)(cx_0x_3-x_0y_2-x_3y_1)-2bv(x_0^2+x_3^2) \end{array}\right] = \mathbf{0}. \quad (9)$$

Algebraic I-O Equation Derivation

- We make the Weierstraß substitutions and convert $\mathbf{H}_1$ and $\mathbf{H}_2$ to factored normal form.
- The denominators are $u^2 + 1$ and $v^2 + 1$ which can safely be factored out because they are always non-vanishing.
- The resulting four algebraic equations are:

$$\mathbf{K}_1 = \begin{bmatrix} (au^2-a)(x_0^2+x_3^2)+2u^2(-x_0y_1+x_3y_2)+2(-x_0y_2+x_3y_1) \\ -2au(x_0^2+x_3^2)-2(1+u^2)(-x_0y_2+x_3y_1) \end{bmatrix} = \mathbf{0}; \quad (8)$$

$$\mathbf{K}_2 = \begin{bmatrix} (v^2(b-d)+b-d)(x_0^2+x_3^2)+(cv^2+c)(x_0^2-x_3^2)+ \\ 2(1+v^2)(-x_0y_1+x_3y_2) \\ 2(v^2+1)(cx_0x_3-x_0y_2-x_3y_1)-2bv(x_0^2+x_3^2) \end{bmatrix} = \mathbf{0}. \quad (9)$$

Algebraic I-O Equation Derivation

- We make the Weierstraß substitutions and convert $\mathbf{H}_1$ and $\mathbf{H}_2$ to factored normal form.
- The denominators are u^2+1 and v^2+1 which can safely be factored out because they are always non-vanishing.
- The resulting four algebraic equations are:

$$\mathbf{K}_1 = \begin{bmatrix} (au^2-a)(x_0^2+x_3^2)+2u^2(-x_0y_1+x_3y_2)+2(-x_0y_2+x_3y_1) \\ -2au(x_0^2+x_3^2)-2(1+u^2)(-x_0y_2+x_3y_1) \end{bmatrix} = \mathbf{0}; \tag{8}$$

$$\mathbf{K}_2 = \begin{bmatrix} (v^2(b-d)+b-d)(x_0^2+x_3^2)+(cv^2+c)(x_0^2-x_3^2)+ \\ 2(1+v^2)(-x_0y_1+x_3y_2) \\ 2(v^2+1)(cx_0x_3-x_0y_2-x_3y_1)-2bv(x_0^2+x_3^2) \end{bmatrix} = \mathbf{0}. \tag{9}$$

Algebraic I-O Equation Derivation

- Factoring the resultant of the first and second elements of $\mathbf{K}_1 = \mathbf{0}$ with respect to u, as well as the first and second elements of $\mathbf{K}_2 = \mathbf{0}$ with respect to v yields the two usual displacement constraint equations in the image space:

$$\begin{aligned} a^2(x_0^2 + x_3^2) - 4(y_1^2 + y_2^2) &= 0, \\ (b^2 - c^2 - d^2)(x_0^2 + x_3^2) + 2cd(x_0^2 - x_3^2) + 4c(x_0y_1 + x_3y_2) + & \\ 4d(-x_0y_1 + x_3y_2) - 4(y_1^2 + y_2^2) &= 0. \end{aligned}$$

- Inspection of the quadratic forms of these two equations reveals that they are two hyperboloids of one sheet, which is exactly what is expected for two *RR* dyads.
- But, these are not the constraints we are looking for!

Algebraic I-O Equation Derivation

- Factoring the resultant of the first and second elements of $\mathbf{K}_1 = \mathbf{0}$ with respect to u, as well as the first and second elements of $\mathbf{K}_2 = \mathbf{0}$ with respect to v yields the two usual displacement constraint equations in the image space:

$$a^2(x_0^2 + x_3^2) - 4(y_1^2 + y_2^2) = 0,$$

$$(b^2 - c^2 - d^2)(x_0^2 + x_3^2) + 2cd(x_0^2 - x_3^2) + 4c(x_0y_1 + x_3y_2) + 4d(-x_0y_1 + x_3y_2) - 4(y_1^2 + y_2^2) = 0.$$

- Inspection of the quadratic forms of these two equations reveals that they are two hyperboloids of one sheet, which is exactly what is expected for two *RR* dyads.
- But, these are not the constraints we are looking for!

Algebraic I-O Equation Derivation

- Factoring the resultant of the first and second elements of $\mathbf{K}_1 = \mathbf{0}$ with respect to u, as well as the first and second elements of $\mathbf{K}_2 = \mathbf{0}$ with respect to v yields the two usual displacement constraint equations in the image space:

$$
\begin{aligned}
a^2(x_0^2 + x_3^2) - 4(y_1^2 + y_2^2) &= 0, \\
(b^2 - c^2 - d^2)(x_0^2 + x_3^2) + 2cd(x_0^2 - x_3^2) + 4c(x_0y_1 + x_3y_2) + & \\
4d(-x_0y_1 + x_3y_2) - 4(y_1^2 + y_2^2) &= 0.
\end{aligned}
$$

- Inspection of the quadratic forms of these two equations reveals that they are two hyperboloids of one sheet, which is exactly what is expected for two *RR* dyads.
- But, these are not the constraints we are looking for!

I-O Equation Derivation

- We want to eliminate the image space coordinates using $\mathbf{K}_1 = \mathbf{0}$ and $\mathbf{K}_2 = \mathbf{0}$ to obtain an algebraic polynomial with the tangent half angles u and v as variables and link lengths as coefficients.
- To obtain this algebraic polynomial we start by setting the homogenising coordinate $x_0 = 1$, which can safely be done since we are only concerned with real finite displacements.
- Observe that the two equations represented by the components of $\mathbf{K}_1 = \mathbf{0}$ are linear in y_1 and y_2, solving leads to:

$$y_1 = \frac{1}{2}\frac{a(u^2 - 2ux_3 - 1)}{u^2 + 1}, \tag{10}$$

$$y_2 = \frac{1}{2}\frac{a(u^2x_3 + 2u - x_3)}{u^2 + 1}. \tag{11}$$

I-O Equation Derivation

- We want to eliminate the image space coordinates using $\mathbf{K}_1 = \mathbf{0}$ and $\mathbf{K}_2 = \mathbf{0}$ to obtain an algebraic polynomial with the tangent half angles u and v as variables and link lengths as coefficients.
- To obtain this algebraic polynomial we start by setting the homogenising coordinate $x_0 = 1$, which can safely be done since we are only concerned with real finite displacements.
- Observe that the two equations represented by the components of $\mathbf{K}_1 = \mathbf{0}$ are linear in y_1 and y_2, solving leads to:

$$y_1 = \frac{1}{2}\frac{a(u^2 - 2ux_3 - 1)}{u^2 + 1}, \tag{10}$$

$$y_2 = \frac{1}{2}\frac{a(u^2 x_3 + 2u - x_3)}{u^2 + 1}. \tag{11}$$

I-O Equation Derivation

- We want to eliminate the image space coordinates using $\mathbf{K}_1 = \mathbf{0}$ and $\mathbf{K}_2 = \mathbf{0}$ to obtain an algebraic polynomial with the tangent half angles u and v as variables and link lengths as coefficients.
- To obtain this algebraic polynomial we start by setting the homogenising coordinate $x_0 = 1$, which can safely be done since we are only concerned with real finite displacements.
- Observe that the two equations represented by the components of $\mathbf{K}_1 = \mathbf{0}$ are linear in y_1 and y_2, solving leads to:

$$y_1 = \frac{1}{2}\frac{a(u^2 - 2ux_3 - 1)}{u^2 + 1}, \tag{10}$$

$$y_2 = \frac{1}{2}\frac{a(u^2x_3 + 2u - x_3)}{u^2 + 1}. \tag{11}$$

Algebraic I-O Equation Derivation

- Equations (10) and (11) reveal the common denominator of $u^2 + 1$, which can never be less than 1, and hence may be factored out.
- Now we back-substitute these expressions for y_1 and y_2 into the array components of $\mathbf{K}_2 = \mathbf{0}$, thereby eliminating these image space coordinates, and factor the resultant of its elements with respect to x_3 which yields four factors.
- The first three are
$$4c^2, \quad (u^2 + 1)^3, \quad (v^2 + 1)^3.$$
- None of these three factors can ever be zero and at the same time represent a real displacement constraint, hence they are eliminated.

Algebraic I-O Equation Derivation

- Equations (10) and (11) reveal the common denominator of $u^2 + 1$, which can never be less than 1, and hence may be factored out.
- Now we back-substitute these expressions for y_1 and y_2 into the array components of $\mathbf{K}_2 = \mathbf{0}$, thereby eliminating these image space coordinates, and factor the resultant of its elements with respect to x_3 which yields four factors.
- The first three are
$$4c^2, \quad (u^2 + 1)^3, \quad (v^2 + 1)^3.$$
- None of these three factors can ever be zero and at the same time represent a real displacement constraint, hence they are eliminated.

Algebraic I-O Equation Derivation

- Equations (10) and (11) reveal the common denominator of $u^2 + 1$, which can never be less than 1, and hence may be factored out.
- Now we back-substitute these expressions for y_1 and y_2 into the array components of $\mathbf{K}_2 = \mathbf{0}$, thereby eliminating these image space coordinates, and factor the resultant of its elements with respect to x_3 which yields four factors.
- The first three are

$$4c^2, \quad (u^2 + 1)^3, \quad (v^2 + 1)^3.$$

- None of these three factors can ever be zero and at the same time represent a real displacement constraint, hence they are eliminated.

Algebraic I-O Equation Derivation

- Equations (10) and (11) reveal the common denominator of $u^2 + 1$, which can never be less than 1, and hence may be factored out.
- Now we back-substitute these expressions for y_1 and y_2 into the array components of $\mathbf{K}_2 = \mathbf{0}$, thereby eliminating these image space coordinates, and factor the resultant of its elements with respect to x_3 which yields four factors.
- The first three are
$$4c^2, \quad (u^2 + 1)^3, \quad (v^2 + 1)^3.$$
- None of these three factors can ever be zero and at the same time represent a real displacement constraint, hence they are eliminated.

Algebraic I-O Equation Derivation

- The remaining factor is a polynomial of degree 4 in the variables u and v, and quadratic in the link length coefficients a, b, c, and d, which is exactly the constraint equation we desire:

$$Au^2v^2 + Bu^2 + Cv^2 - 8abuv + D = 0; \qquad (12)$$

where

$$\begin{aligned}
A &= (a-b+c+d)(a-b-c+d);\\
B &= (a+b-c+d)(a+b+c+d);\\
C &= (a+b-c-d)(a+b+c-d);\\
D &= (a-b+c-d)(a-b-c-d).
\end{aligned}$$

Algebraic I-O Equation Derivation

- As a concomitant check on the validity of Equation (12) we see that the same equation is obtained using the Weierstraß substitution in the Freudenstein equation.

Discrete Approximate Synthesis Example

- We illustrate the utility of the algebraic form of the I-O equation to discrete approximate dimensional synthesis with an example.
- The function we wish to approximate is defined for $v = f(u)$ as:

$$v = 2 + \tan\left(\frac{u}{u^2+1}\right).$$

- The range of the input is $-2 \le u \le 2$, and the output v varies according to the function.
- The cardinality of the I-O data sets is $m = 5$, 10, 50, 100, and 500, respectively.
- We employ Equation (12) as the synthesis equation.
- Now we use a Newton-Gauss algorithm to identify the link lengths that minimise the design error, the residual of the m synthesis equations.

Discrete Approximate Synthesis Example

- We illustrate the utility of the algebraic form of the I-O equation to discrete approximate dimensional synthesis with an example.
- The function we wish to approximate is defined for $v = f(u)$ as:

$$v = 2 + \tan\left(\frac{u}{u^2+1}\right).$$

- The range of the input is $-2 \leq u \leq 2$, and the output v varies according to the function.
- The cardinality of the I-O data sets is $m = 5, 10, 50, 100$, and 500, respectively.
- We employ Equation (12) as the synthesis equation.
- Now we use a Newton-Gauss algorithm to identify the link lengths that minimise the design error, the residual of the m synthesis equations.

Discrete Approximate Synthesis Example

- We illustrate the utility of the algebraic form of the I-O equation to discrete approximate dimensional synthesis with an example.
- The function we wish to approximate is defined for $v = f(u)$ as:

$$v = 2 + \tan\left(\frac{u}{u^2+1}\right).$$

- The range of the input is $-2 \leq u \leq 2$, and the output v varies according to the function.
- The cardinality of the I-O data sets is $m = 5, 10, 50, 100$, and 500, respectively.
- We employ Equation (12) as the synthesis equation.
- Now we use a Newton-Gauss algorithm to identify the link lengths that minimise the design error, the residual of the m synthesis equations.

Discrete Approximate Synthesis Example

- We illustrate the utility of the algebraic form of the I-O equation to discrete approximate dimensional synthesis with an example.
- The function we wish to approximate is defined for $v = f(u)$ as:

$$v = 2 + \tan\left(\frac{u}{u^2+1}\right).$$

- The range of the input is $-2 \le u \le 2$, and the output v varies according to the function.
- The cardinality of the I-O data sets is $m = 5, 10, 50, 100$, and 500, respectively.
- We employ Equation (12) as the synthesis equation.
- Now we use a Newton-Gauss algorithm to identify the link lengths that minimise the design error, the residual of the m synthesis equations.

Discrete Approximate Synthesis Example

- We illustrate the utility of the algebraic form of the I-O equation to discrete approximate dimensional synthesis with an example.
- The function we wish to approximate is defined for $v = f(u)$ as:

$$v = 2 + \tan\left(\frac{u}{u^2+1}\right).$$

- The range of the input is $-2 \leq u \leq 2$, and the output v varies according to the function.
- The cardinality of the I-O data sets is $m = 5, 10, 50, 100$, and 500, respectively.
- We employ Equation (12) as the synthesis equation.
- Now we use a Newton-Gauss algorithm to identify the link lengths that minimise the design error, the residual of the m synthesis equations.

Discrete Approximate Synthesis Example

- We illustrate the utility of the algebraic form of the I-O equation to discrete approximate dimensional synthesis with an example.
- The function we wish to approximate is defined for $v = f(u)$ as:

$$v = 2 + \tan\left(\frac{u}{u^2+1}\right).$$

- The range of the input is $-2 \leq u \leq 2$, and the output v varies according to the function.
- The cardinality of the I-O data sets is $m = 5, 10, 50, 100$, and 500, respectively.
- We employ Equation (12) as the synthesis equation.
- Now we use a Newton-Gauss algorithm to identify the link lengths that minimise the design error, the residual of the m synthesis equations.

Discrete Approximate Synthesis Example

- Since a function generator is scalable, we only require the ratios of the link lengths to generate a function.
- Hence, we set $d = 1$ generic unit of length, and identify the a, b, and c that minimise the residual error in the synthesis equations given an arbitrary number $m > 3$ of (u_m, v_m) values of the prescribed function.
- We wish to minimise the residual of the synthesis equations with respect to changes in the unknown link lengths.
- The synthesis equation is

$$Au^2v^2 + Bu^2 + Cv^2 - 8abuv + D = 0.$$

Discrete Approximate Synthesis Example

- Since a function generator is scalable, we only require the ratios of the link lengths to generate a function.
- Hence, we set $d = 1$ generic unit of length, and identify the a, b, and c that minimise the residual error in the synthesis equations given an arbitrary number $m > 3$ of (u_m, v_m) values of the prescribed function.
- We wish to minimise the residual of the synthesis equations with respect to changes in the unknown link lengths.
- The synthesis equation is

$$Au^2v^2 + Bu^2 + Cv^2 - 8abuv + D = 0.$$

Discrete Approximate Synthesis Example

- Since a function generator is scalable, we only require the ratios of the link lengths to generate a function.
- Hence, we set $d = 1$ generic unit of length, and identify the a, b, and c that minimise the residual error in the synthesis equations given an arbitrary number $m > 3$ of (u_m, v_m) values of the prescribed function.
- We wish to minimise the residual of the synthesis equations with respect to changes in the unknown link lengths.
- The synthesis equation is

$$Au^2v^2 + Bu^2 + Cv^2 - 8abuv + D = 0.$$

Discrete Approximate Synthesis Example

- Since a function generator is scalable, we only require the ratios of the link lengths to generate a function.
- Hence, we set $d = 1$ generic unit of length, and identify the a, b, and c that minimise the residual error in the synthesis equations given an arbitrary number $m > 3$ of (u_m, v_m) values of the prescribed function.
- We wish to minimise the residual of the synthesis equations with respect to changes in the unknown link lengths.
- The synthesis equation is

$$Au^2v^2 + Bu^2 + Cv^2 - 8abuv + D = 0.$$

Discrete Approximate Synthesis Example

- We define the i^{th} Jacobian of the synthesis equations to be the $m \times 3$ array

$$\mathbf{J}_i = \begin{bmatrix} \frac{\partial f}{\partial a_i}(u_1, v_1) & \frac{\partial f}{\partial b_i}(u_1, v_1) & \frac{\partial f}{\partial c_i}(u_1, v_1) \\ \frac{\partial f}{\partial a_i}(u_2, v_2) & \frac{\partial f}{\partial b_i}(u_2, v_2) & \frac{\partial f}{\partial c_i}(u_2, v_2) \\ \vdots & \vdots & \vdots \\ \frac{\partial f}{\partial a_i}(u_m, v_m) & \frac{\partial f}{\partial b_i}(u_m, v_m) & \frac{\partial f}{\partial c_i}(u_m, v_m) \end{bmatrix}.$$

- The design error is the i^{th} residual $\mathbf{r}_i$ of the synthesis equations for the current values of a_i, b_i, and c_i.

Discrete Approximate Synthesis Example

- We define the i^{th} Jacobian of the synthesis equations to be the $m \times 3$ array

$$\mathbf{J}_i = \begin{bmatrix} \frac{\partial f}{\partial a_i}(u_1, v_1) & \frac{\partial f}{\partial b_i}(u_1, v_1) & \frac{\partial f}{\partial c_i}(u_1, v_1) \\ \frac{\partial f}{\partial a_i}(u_2, v_2) & \frac{\partial f}{\partial b_i}(u_2, v_2) & \frac{\partial f}{\partial c_i}(u_2, v_2) \\ \vdots & \vdots & \vdots \\ \frac{\partial f}{\partial a_i}(u_m, v_m) & \frac{\partial f}{\partial b_i}(u_m, v_m) & \frac{\partial f}{\partial c_i}(u_m, v_m) \end{bmatrix}.$$

- The design error is the i^{th} residual $\mathbf{r}_i$ of the synthesis equations for the current values of a_i, b_i, and c_i.

Discrete Approximate Synthesis Example

- The design error has the form of an $m \times 1$ array

$$\mathbf{r}_i = \begin{bmatrix} f(a_i, b_i, c_i, u_1, v_1) \\ f(a_i, b_i, c_i, u_2, v_2) \\ \vdots \\ f(a_i, b_i, c_i, u_m, v_m) \end{bmatrix}.$$

- The Newton-Gauss algorithm iterates until a desired minimum threshold value is attained.

- We define the i^{th} estimate of the link lengths as the parameter vector

$$\mathbf{x}_i = \begin{bmatrix} a_i \\ b_i \\ c_i \end{bmatrix}.$$

Discrete Approximate Synthesis Example

- The design error has the form of an $m \times 1$ array

$$\mathbf{r}_i = \begin{bmatrix} f(a_i, b_i, c_i, u_1, v_1) \\ f(a_i, b_i, c_i, u_2, v_2) \\ \vdots \\ f(a_i, b_i, c_i, u_m, v_m) \end{bmatrix}.$$

- The Newton-Gauss algorithm iterates until a desired minimum threshold value is attained.
- We define the i^{th} estimate of the link lengths as the parameter vector

$$\mathbf{x}_i = \begin{bmatrix} a_i \\ b_i \\ c_i \end{bmatrix}.$$

Discrete Approximate Synthesis Example

- The design error has the form of an $m \times 1$ array

$$\mathbf{r}_i = \begin{bmatrix} f(a_i, b_i, c_i, u_1, v_1) \\ f(a_i, b_i, c_i, u_2, v_2) \\ \vdots \\ f(a_i, b_i, c_i, u_m, v_m) \end{bmatrix}.$$

- The Newton-Gauss algorithm iterates until a desired minimum threshold value is attained.
- We define the i^{th} estimate of the link lengths as the parameter vector

$$\mathbf{x}_i = \begin{bmatrix} a_i \\ b_i \\ c_i \end{bmatrix}.$$

Discrete Approximate Synthesis Example

- The algorithm iteratively proceeds updating $\mathbf{x}_i$ as

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \mathbf{r}_i(\mathbf{J}_i^T\mathbf{J}_i)^{-1}\mathbf{J}_i^T.$$

- The iteration continues until the following condition is met

$$||\mathbf{x}_{i+1} - \mathbf{x}_i|| \leq \epsilon.$$

- For the example we set $\epsilon = 1 \times 10^{-10}$.
- The initial guess for the link lengths was $a = b = c = 1$.

Discrete Approximate Synthesis Example

- The algorithm iteratively proceeds updating $\mathbf{x}_i$ as

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \mathbf{r}_i(\mathbf{J}_i^T\mathbf{J}_i)^{-1}\mathbf{J}_i^T.$$

- The iteration continues until the following condition is met

$$||\mathbf{x}_{i+1} - \mathbf{x}_i|| \leq \epsilon.$$

- For the example we set $\epsilon = 1 \times 10^{-10}$.
- The initial guess for the link lengths was $a = b = c = 1$.

Discrete Approximate Synthesis Example

- The algorithm iteratively proceeds updating $\mathbf{x}_i$ as

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \mathbf{r}_i(\mathbf{J}_i^T\mathbf{J}_i)^{-1}\mathbf{J}_i^T.$$

- The iteration continues until the following condition is met

$$||\mathbf{x}_{i+1} - \mathbf{x}_i|| \leq \epsilon.$$

- For the example we set $\epsilon = 1 \times 10^{-10}$.
- The initial guess for the link lengths was $a = b = c = 1$.

Discrete Approximate Synthesis Example

- The algorithm iteratively proceeds updating $\mathbf{x}_i$ as

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \mathbf{r}_i(\mathbf{J}_i^T\mathbf{J}_i)^{-1}\mathbf{J}_i^T.$$

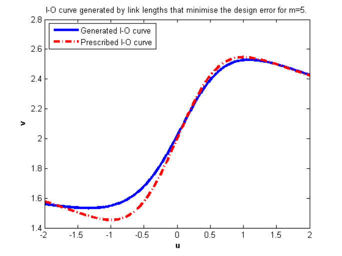
- The iteration continues until the following condition is met

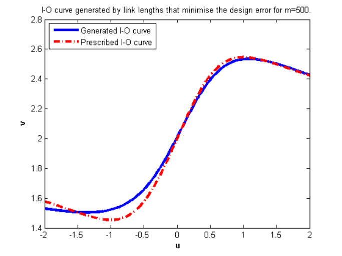
$$||\mathbf{x}_{i+1} - \mathbf{x}_i|| \leq \epsilon.$$

- For the example we set $\epsilon = 1 \times 10^{-10}$.
- The initial guess for the link lengths was $a = b = c = 1$.

Discrete Approximate Synthesis Example

Link lengths	Sample set cardinality					
	3	5	10	50	100	500
a	0.2331	0.1991	0.2033	0.2072	0.2077	0.2082
b	1.4879	1.4879	1.4893	1.4836	1.4827	1.4818
c	1.2202	1.2202	1.2249	1.2239	1.2236	1.2233
d	1	1	1	1	1	1





Conclusions

- We have presented a new method for deriving the algebraic form of the I-O equations of planar $4R$ function generators employing, then eliminating appropriate sets of Study's soma coordinates.
- The next step in this work is to generalise this method for any 4-bar kinematic architecture: planar; spherical; or spatial.
- The ultimate goal of this work is to adapt it to use continuous I-O data sets to synthesise the very best linkage to generate an arbitrary function.
- Derivation of the planar algebraic I-O equation presented here is one of the first steps towards this goal.

Conclusions

- We have presented a new method for deriving the algebraic form of the I-O equations of planar $4R$ function generators employing, then eliminating appropriate sets of Study's soma coordinates.
- The next step in this work is to generalise this method for any 4-bar kinematic architecture: planar; spherical; or spatial.
- The ultimate goal of this work is to adapt it to use continuous I-O data sets to synthesise the very best linkage to generate an arbitrary function.
- Derivation of the planar algebraic I-O equation presented here is one of the first steps towards this goal.

Conclusions

- We have presented a new method for deriving the algebraic form of the I-O equations of planar $4R$ function generators employing, then eliminating appropriate sets of Study's soma coordinates.
- The next step in this work is to generalise this method for any 4-bar kinematic architecture: planar; spherical; or spatial.
- The ultimate goal of this work is to adapt it to use continuous I-O data sets to synthesise the very best linkage to generate an arbitrary function.
- Derivation of the planar algebraic I-O equation presented here is one of the first steps towards this goal.

Carleton UNIVERSITY

Conclusions

- We have presented a new method for deriving the algebraic form of the I-O equations of planar $4R$ function generators employing, then eliminating appropriate sets of Study's soma coordinates.
- The next step in this work is to generalise this method for any 4-bar kinematic architecture: planar; spherical; or spatial.
- The ultimate goal of this work is to adapt it to use continuous I-O data sets to synthesise the very best linkage to generate an arbitrary function.
- Derivation of the planar algebraic I-O equation presented here is one of the first steps towards this goal.