THEORY OF MACHINES

ME 333

Report on Plagiograph Mechanism

1.1 INTRODUCTION

A Plagiograph (*pla·gi·o·graph*) is mechanism for tracing images and scaling their size. It is inspired from another four-bar mechanism called Pantograph. Thus, it is also known as Skew – Pantograph. It is a simple but quite effective mechanism consisting of two binary and two ternary links with one end being fixed. The free-end is made to move along the target path and the exact image is traced at the other end. The input motion depends on the position *(x, y)* of the free end in the 2D space, thus it has two degrees of freedom.

The idea of a Plagiograph was first proposed and built by Sir Alfred Bray Kemple, a Kinematician. He was born on July 6,1849 in London. His contributions to the kinematics include Kempe's Theorem, Straight line, Rectilinear Translation etc. He developed the Plagiograph with Sylvester by combining two similar ternary links with two binary links and forming a parallelogram configuration.

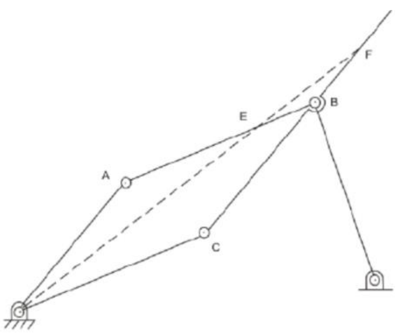


Figure 1.1.1

Sir Alfred Bray Kemple

The ratio of the lengths of the triangles determined the scale factor i.e., the extent to which the target can be scaled. Thus, this mechanism was also known as Kempe – Sylvester Plagiograph.

Kempe first reported to Sylvester that the motion of a coupler point E (Figure 1.1) on the coupler AB of a four-bar linkage OABD can be reproduced with a constant scale factor (= *AB/AE*) by point F of the six-bar linkage shown in Figure 12, where OABC is a parallelogram.

This copying mechanism is called `Plagiograph'.

1.2 Number Analysis

Based on the observations the only known parameters are the desired Degree of freedom (The motion is generated by the translation and rotation of the Ternary links) and the number of links. Performing number synthesis would give an estimate of the number, type of link and joints. From the Fig. 1.1.1, we can draw the following conclusions:

$n = 6$

$j_1 = 3;\ j_2 = 2\ (j = 7)$

$$
\begin{aligned}
DOF &= 3(n-1) - 2(j_1 + 2j_2) \\
&= 3(6-1) - 2(3+4) \\
&= 1
\end{aligned}
$$

Thus, applying Grubbler's criteria

$i_{max} = n/2 = 6/2 = 3$

$n_2 + n_3 = n \rightarrow (i)$

$2n_2 + 3n_3 = 2j \rightarrow (ii)$

On substitution

$n_2 + n_3 = 6$

$2n_2 + 3n_3 = 2(7) = 14$

On solving, $(n_2 = 4\ \&\ n_3 = 2)$

Historically, the mechanism is built by one such method by Kempe and Sylvester immediately extended it to general coupler point E not necessarily lying on AB (Figure *1.2.1*). Here triangles ABE and CBF are similar and OABC is a parallelogram. In this mechanism the curve traced out by F is a scale drawing (with a constant factor = *AB/AE*) of that produced by E and oriented at a fixed angle in both these six-link mechanisms, the link DB only decided the coupler path.

Removing this link, one gets five-link mechanism with two degrees of freedom. This works fine with our existing result of having two ternary links. The removal of extra binary links resulted in the change of Degree of freedom from 1 to 2.

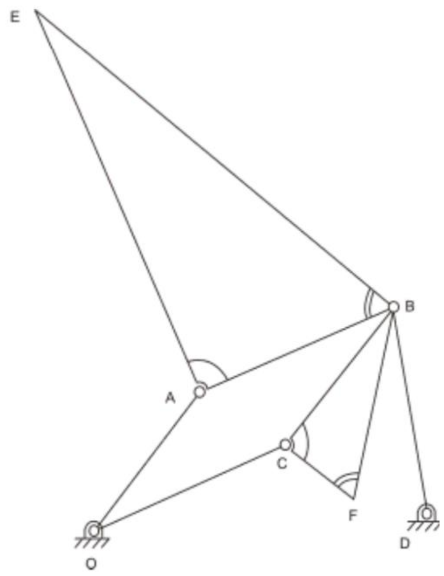


Fig 1.2.1

So, point E can now be moved to trace any arbitrary planar curve, which will be reproduced to a scale and oriented at a constant angle.

This concludes our analysis on the number of links. We will consider the mechanism to have variable link lengths from now on out and derive the relation between the input and output displacements (Kinematic Analysis).

1.3 Kinematic Analysis

The Plagiograph Mechanism has an input parameter X that depends on the location *(x, y)* of the free end. The image is traced at an angle ***α*** with respect to the target in the same plane. However, there are some restrictions to the plane motion due to links size and the joints which will be addressed in the **section 1.5.** Performing Kinematic analysis using *Vectors* will help provide an insight on the relation between output and input motion. The plane of analysis is taken as complex on ***Y*** and Real on ***X*** for easy calculations.

The vector equations representing the locations of the free end be given as:

$$\vec{X} = \vec{OA} + \vec{AX} \qquad \vec{Y} = \vec{OB} + \vec{BY}$$

The parallelogram is always maintained

$$\vec{OB} = \vec{AC} = \frac{|\vec{AC}|}{|\vec{AX}|}\vec{AX}e^{i\alpha}$$

$$\vec{BY} = \frac{|\vec{BY}|}{|\vec{BC}|}\vec{BC}\cdot e^{i\alpha} = \frac{|\vec{AC}|}{|\vec{AX}|}\vec{OA}e^{i\alpha}$$

∵ ΔACX is similar to ΔBCY and $BC \parallel OA$

From Triangle Law

$$\vec{OY} = \vec{OB} + \vec{BY}$$

$$\vec{Y} = \frac{|\vec{AC}|}{|\vec{AX}|}(\vec{OA} + \vec{AX})e^{i\alpha}$$

$$\boxed{\vec{Y} = \frac{|\vec{AC}|}{|\vec{AX}|}\cdot\vec{X}\cdot e^{i\alpha}}$$

Note that all the link lengths are constant and the angle between the sides of the triangle ***α*** is constant. e^{α} is used to rotates any vector by an angle alpha in the complex plane. It can be also written as:

$$\vec{X'} = \vec{X}\cdot e^{i\alpha}$$

or

	x	y
X'	cosα	-sinα
Y'	sinα	cosα

Here, differentiating with time on both sides

$$\frac{d}{dt}(\vec{Y}) = \frac{d}{dt}\left(\frac{AC}{AX}\cdot\vec{X}\cdot e^{i\alpha}\right)$$

$$\Rightarrow \dot{\vec{Y}} = \frac{|AC|}{|AX|}e^{i\alpha}\dot{\vec{X}} \quad : \text{Velocity Relation}$$

differentiating again…

$$\Rightarrow \ddot{\vec{Y}} = \frac{|AC|}{|AX|}e^{i\alpha}\ddot{\vec{X}} \quad : \text{Acceleration Relation}$$

Thus, all the three parameters displacement, velocity and acceleration are linearly dependent ($x \propto y$) on each other and these are functions of X *(x, y)*.

The velocity and Acceleration terms at any point can be derived by differentiating the vector equations at any point. The value of Angular velocity at O can be obtained differentiating theta w.r.t time. These will be of the form:

$$x: \quad OA = K\cdot X\cdot e^{+i\theta}$$

$$v: \quad \dot{OA} = K\dot{X}e^{+i\theta}$$

$$a: \quad \ddot{OA} = K\ddot{X}e^{+i\theta}$$

1.3.1 Code

Marking the other points of the mechanism using vector and geometry

```
#parameters
alpha = d2r(90)             # The angle XAC / YBC
beta = np.arctan(1.4)       # The angle at the free end
gamma = np.pi - beta - alpha
sf = 1.4                    # Scaling Factor
l1 = 5                      # Length of binary link 1
l2 = 3.5                    # Length of binary link 2


X = np.array([[7,1]])                                          # Input


theta = angle(l1 ,bar(X) ,l2 / (np.sin(beta) / np.sin(gamma)))  # Angle AOX
Y = rotation(X, alpha, sf)                                      # Corresponding position of Y (output)
OA = rotation(X, theta, l1 / bar(X))                            # Marking Point A by rotating OX by theta and scaling it


psi = angle(bar(X),(l2 * np.sin(gamma) / np.sin(beta)),bar(OA)) # Angle AXO
AX = -rotation(-X, -psi, (l2 * np.sin(gamma) / np.sin(beta)) / bar(X))
                                                                # Rotating OX by angle phi to get AX
AC = rotation(AX, alpha, l2 / bar(AX))                          # Rotating AX by alpha to to get AC
XC = AC - AX
OC = OA + AC


BC = OA                                                         # Parallel vectors have the same direction (and magnitude)
OB = AC
BY = rotation(BC,alpha,1 /(np.sin(gamma) / np.sin(beta)))
theta
```

```
def d2r(d):
    return d * np.pi / 180

def rotation(V,theta,sf):
    vxnew = (V[:,0] * np.cos(theta) - V[:,1] * np.sin(theta)) * sf
    vynew = (V[:,0] * np.sin(theta) + V[:,1] * np.cos(theta)) * sf

    Vnew = []
    for i in range(len(V)):
        Vnew.append([vxnew[i], vynew[i]])
    Vnew = np.array(Vnew)

    return Vnew

def angle(l1,l2,l3):
    return np.arccos((l1 ** 2 + l2 ** 2 - l3 ** 2) / (2 * l1 * l2))

def bar(V):
    dist = (V[:,0] ** 2 + V[:,1] ** 2) ** 0.5
    return dist
```

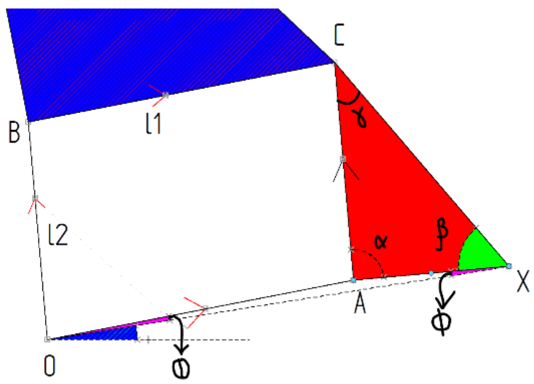


Fig 1.3.1

1.4 Kinematic Synthesis

Sno	X (x1, y1)	Y (x2, y2)
1	(7.5,0)	(0,10.5)
2	(7,0)	(0, 9.80)
3	(5.6,0)	(0,7.84)

Now that the input-output motions are known let's use this knowledge in finding all the link lengths. The method of getting required link lengths is called **Kinematic synthesis**.

From the **Section 1.3**, we learn that the input and output motions are linearly dependent on each other. This conclusion is necessary for deriving the link lengths and constructing the Plagiograph mechanism in real world. The Plagiograph Mechanism can be synthesized using the relation,

$$x \propto y$$

where, x and y are relative displacement of the free ends.

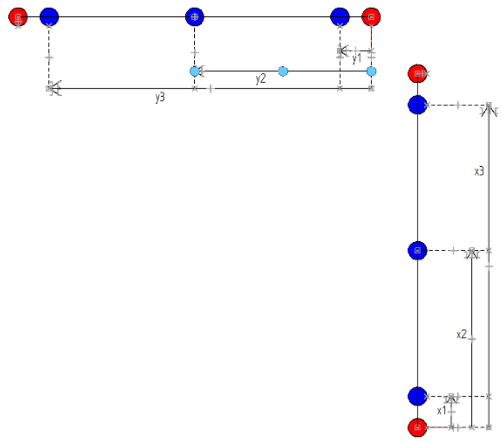


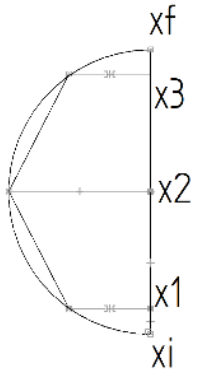
Fig. 1.4.1

For simplicity, let's consider a 1D problem. i.e., the input motion is only along one direction. From the figure, the resultant output moment is always in the direction perpendicular to the input direction.

This phenomenon is explained in **Analysis section.** Now, let's Find the dimensions of the four links using. The following points are tabulated below.

USING THREE-POINT SYNTHESIS

Let x_i = *5.6* and x_f = *7.5 (abscissa)* and their corresponding image y_i = *7.84* and y_f = *10.5 (ordinates)*



from three point synthesis

$x_i = 5.6$; $x_f = 7.5$

$a = (x_i + x_f)/2 = 6.55$

$h = (x_f - x_i)/2 = 0.95$

$x_j = a + h\cos\left(\frac{2j-1}{2n}\right)\pi$ where $n = 3$, $j = 1, 2, 3$

$= 6.55 + 0.95\cos\left(\frac{2j-1}{6}\right)\pi$

$x_1 = 7.3127$

$x_2 = 6.55$

$x_3 = 5.727$

Note that

$x_K \propto K y_K$

thus,

$\Rightarrow (x_f - x_i) = K'(y_f - y_i)$

$\Rightarrow \frac{(x_i - x_f)}{(y_i - y_f)} = K' = \frac{5.6 - 7.5}{7.84 - 10.5} = \frac{1}{1.4}$

$\Rightarrow \boxed{\frac{1}{K'} = K = 1.4}$

The corresponding values of y_{12}, y_{23} can be estimated using the relation $\boxed{y_{jK} = K x_{jK}}$

Now, all the possible locations of Y can be estimated using the above relation with X.

Now using function generation to estimate the length using **Analytical method**.

We know ***k* = *1.4.*** Substitute the values of ***X*** (*x1, y1*) and ***Y*** (*x2, y2*) in the displacement equation to estimate the value of ***AC / AX*** and ***BY / BC***.

Note that when $\alpha = 90$ and $e^{i\alpha} = i$. Applying Modulus on both sides.

then

$$|\vec{Y}| = \frac{l_1}{l_2}|\vec{X}|$$

substitute the values of $|\vec{X}(x_1, y_1)|$ and $|\vec{Y}(x_2, y_2)|$

then,

$$\frac{l_1}{l_2} = \frac{|\vec{Y}|}{|\vec{X}|}$$

We need to assume at least two arbitrary link lengths to continue our synthesis. Let the length of the smallest binary link be *25 mm.*

$\boldsymbol{l2} = 25\ mm$

⇨ $\boldsymbol{l1} = k * \boldsymbol{l2} = 1.4 * 25 = 35\ mm$

Assume that The bigger triangle's length $\boldsymbol{L2} = 50\ mm$

⇨ $\boldsymbol{L1} = k * \boldsymbol{L2} = 1.4 * 50 = 70\ mm$

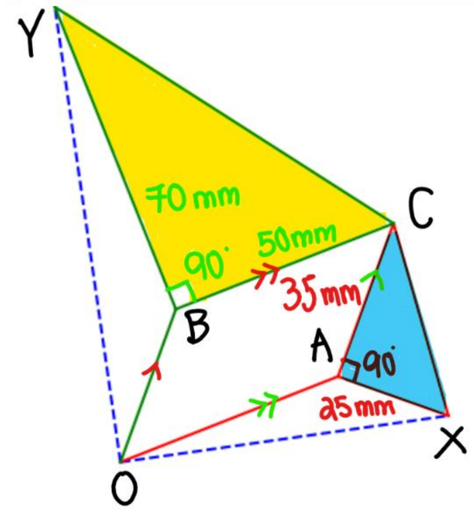
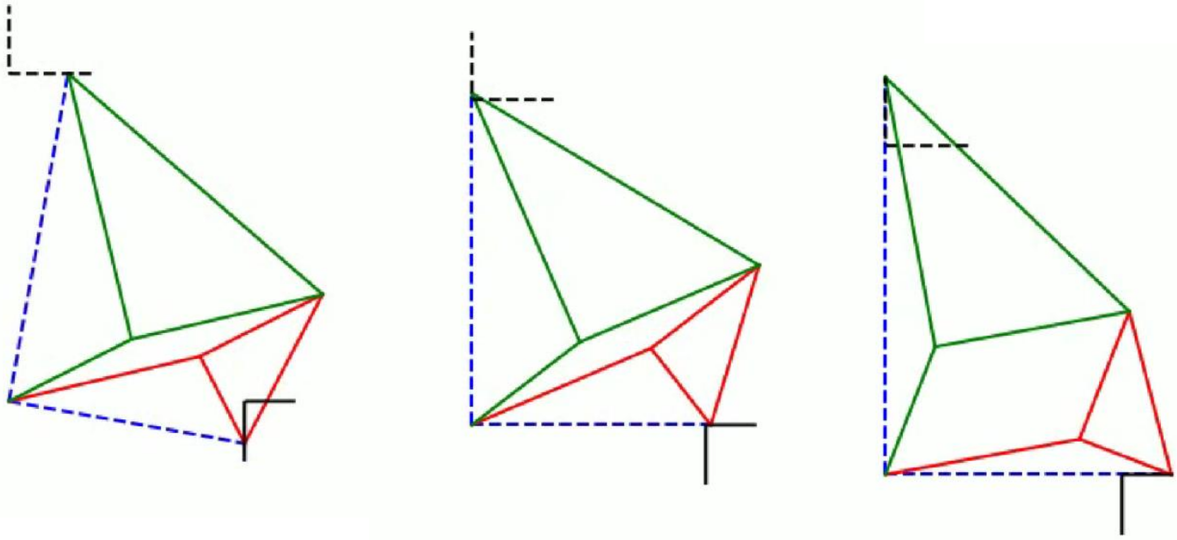


Fig 1.4.2

Thus, this analysis delivers the link lengths required for our mechanism. Plugging these values into a rough sketch will give an exact image of our mechanism. Thus, this mechanism will generate images that are scaled by a factor of ***1.4*** in the direction perpendicular to *X.* The complete mechanism after assembling the parts looks like the following figure.

1.4.1 Working



1.5 Constrain Analysis

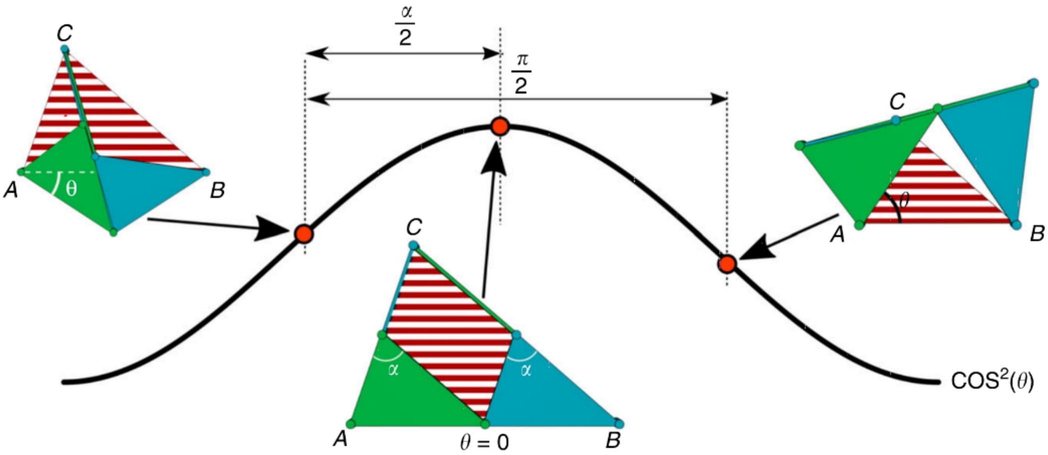


Fig 1.5.1

The derived equation from the earlier section, which gives the relation between ***X*** and ***Y*** is valid for any in the domain $\mathbb{R}^2$. This means the Plagiograph mechanism is capable of tracing images at any point in the 2D coordinate System! This sounds too good to be true and it's actually not. Translating the mechanism into real world in itself has its own sets of challenges or constraints. The links have a specific thickness which restricts the free motion of the movable ends. The joints also contribute to this feature and there is no physical solution to this problem.

Here θ varies for different positions of A.

The max possible value of θ possible is given by

$$\theta = \cos^{-1}(1/K)$$

The range of motions is restricted due to collisions between the rigid bodies and pairs. This is best represented from the above figure.

Substituting the value of scaling factor $k = 1.4$ in the equation gives a value of *45* degree. This estimates that all possible angles of theta lie in the range *(-45, 45*)

1.5.1 Computational Analysis

Plotting all the possible combinations on paper is a hassle. So, I built a code which plots the spectrum of all possible orientations of the mechanism. Note that all the locations (except the purple patches) are in the range of scanning i.e., any image with size within the spectrum can be traced with ease!

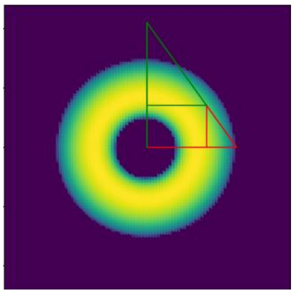


Fig. 1.5.1

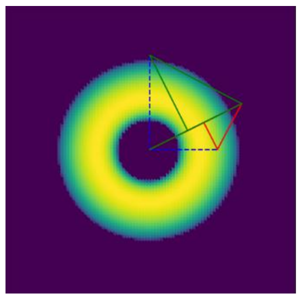
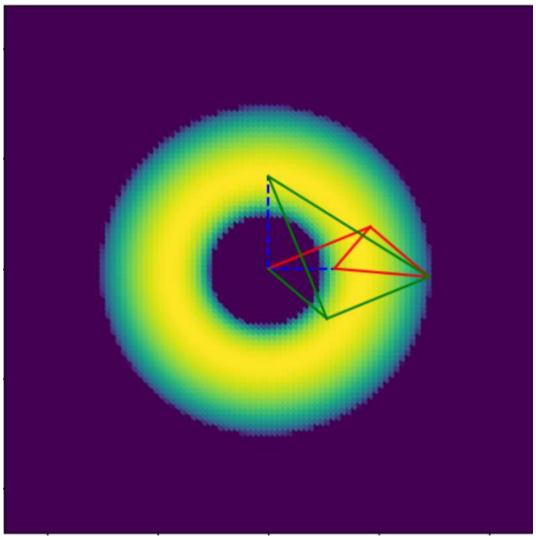
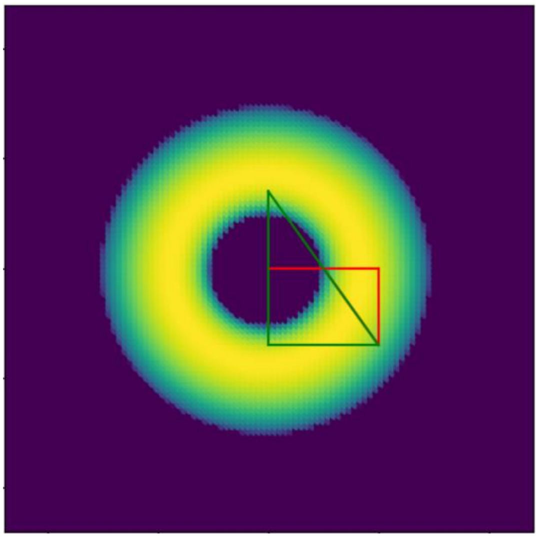


Fig 1.5.2

But sadly, this is where the fairy tale ends. When applied to real world many constraints come in to play a major role. These narrow down the range of measurement to a very small value which hinders the compatibility. Kinematically, a structure constructed from the proposed Plagiograph mechanisms can scale down to a point. In reality, the range of motion of the planar pantographs is limited because of collisions among the rigid bodies that make up the Plagiograph



You might wonder what happens if we go beyond this range, for example take a point at (*3,0*) or (*4.5,1*). The resulted in some bizarre configurations which are impossible to achieve in real world. Thus, the mechanism becomes useless for tracing large images / objects.

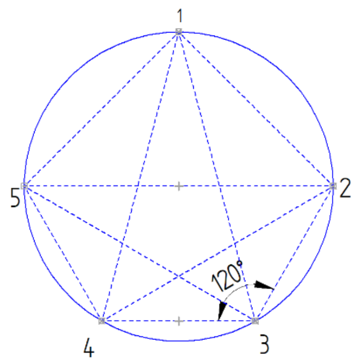


The maximum and minimum possible distances from the fixed end which can be traced are ~*7.5* cm and ~*5.6* cm.

i.e., $\boldsymbol{5.6} < (x, y) < \boldsymbol{7.5}$

1.6 Aronhold Kennedy theorem.

Aronhold Kennedy theorem is a graphical method used for plotting the instantaneous centers and their velocity. This theorem is limited to mechanisms with six links or less. Since the number of links used in this mechanism are **5**, this theorem can be applied with no hassle.



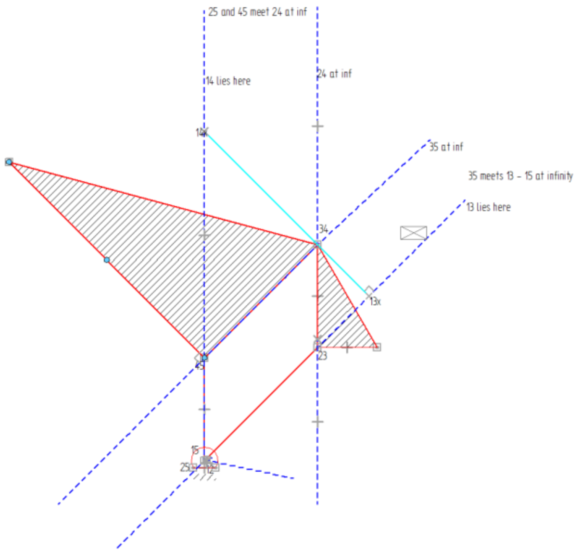


Fig. 1.6.1

From the above analysis, the instantaneous centers between the links 2-5, 4-5, 2-4, and so-on all meet at infinity. The points 1-3, 1-4 lie along the same line passing through 3-4.

Sources

1. Sir Alfred Bray Kempe – An Amateur Kinematician, article by A.K Mallik, 2007.
2. A general method for the creation of dilational surfaces, Freek G. J. Broeren, e.t, 2019.
3. Image credits
    a. https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.wikidata.org%2Fwiki%2FQ370037&psig=AOvVaw339f7_dT4rPeijvs5wmf6O&ust=1701104923628000&source=images&cd=vfe&opi=89978449&ved=0CBIQjRxqFwoTCOCmqOiT4oIDFQAAAAAdAAAAABAE
    b. Python - Matplotlib