Motion Polynomials and Planar Linkages

Christoph Koutschan
Johann Radon Institute for Computational and Applied Mathematics
Austrian Academy of Sciences
Altenberger Straße 69, 4040 Linz, Austria
`christoph.koutschan@ricam.oeaw.ac.at`

Abstract

We present the Mathematica package PlanarLinkages, which provides commands for constructing and visualizing planar linkages that draw a prescribed algebraic curve of genus 0, or more generally, that follow a prescribed rational motion. Since the implemented algorithms are heavily based on the concept of motion polynomials, the functionality of the package also includes basic arithmetic of motion polynomials and a factorization procedure.

1 Introduction

A *linkage* is a mechanical device consisting of rigid bodies (called *links*) that are connected by *joints*, see Figure 1 for examples. Here we consider only *planar linkages*, i.e., linkages all whose links move in parallel planes, and *rotational joints*. As the linkage moves, the trace of a particular point located on the linkage can therefore be identified with a curve in the plane.

The problem of constructing a planar linkage that draws a given algebraic curve was first addressed and solved by Kempe [2]. In a recent article [1] the symbolic computation group at RICAM, including the author, designed a new algorithm for the same problem. The advantage of our algorithm is that it yields much simpler linkages (the number of links and joints is only linear in the

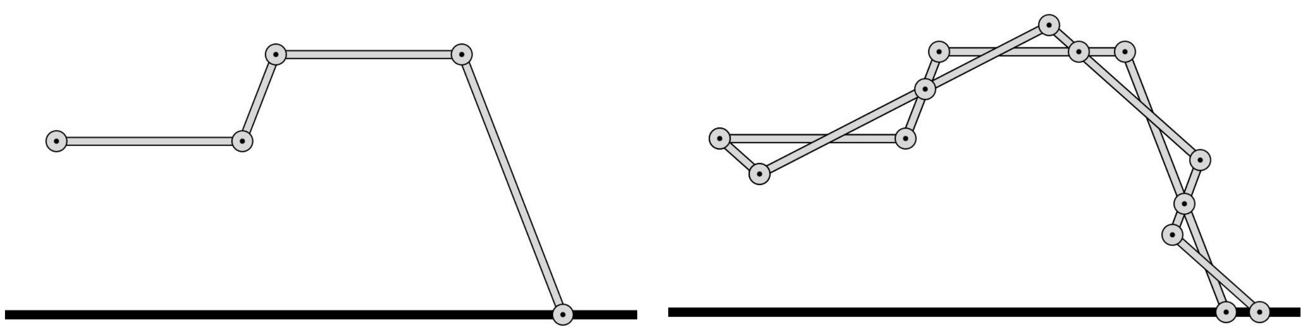

Figure 1: An open chain linkage and its extension to a linkage of mobility one, realizing the translational motion given by $P(t)$ in Equation (3).

degree of the curve), and that it allows for a simple collision detection, but the drawback is that it is only applicable to rational curves, i.e., curves that are parametrizable by rational functions.

We now sketch our method of constructing planar linkages and a bit of theoretical background. The interested reader is referred to our paper [1] where all this is laid out in detail, and from which also the following example is taken: we consider the ellipse given by the zero set of the polynomial $(x+1)^2+4y^2=1$. Our goal is to construct a linkage with rotational joints that draws this ellipse and that admits only one degree of freedom. More precisely, "drawing" means that there is a specific link (on which we put the pen) that performs a motion along the ellipse as the linkage moves.

Mathematically speaking, a motion is a one-dimensional family of direct isometries. We encode direct isometries as elements of the algebra $\mathbb{K}=\mathbb{C}[\eta]\,/\,(\eta^2, i\eta+\eta i)$. Its elements hence are of the form $z+\eta w$ with $z,w\in\mathbb{C}$, and according to the above rules they are multiplied as follows:

$$(z_1+\eta w_1)\cdot(z_2+\eta w_2) \;=\; (z_1\,z_2)+\eta\,(\overline{z_1}w_2+z_2w_1). \tag{1}$$

By defining on $\mathbb{K}$ the equivalence relation

$$k_1\sim k_2 \;:\Longleftrightarrow\; k_1=\alpha k_2 \text{ for some } \alpha\in\mathbb{R}\setminus\{0\}, \tag{2}$$

we can show that the multiplicative group $\{z+\eta w\in\mathbb{K}\mid z\neq 0\}\,/\sim$ is isomorphic to the special Euclidean group SE_2, i.e., the set of direct isometries in the plane with composition as the group operation; see `Out[7]` below for the isomorphism. A univariate polynomial in $\mathbb{K}[t]$ is then called a *motion polynomial*. Motions that can be represented in this way are called *rational motions*.

Our ellipse admits the rational parametrization $\varphi(t)=\big(-2/(t^2+1),t/(t^2+1)\big)$, from which we deduce that a translational motion along this ellipse is represented by the motion polynomial

$$P(t)=(t^2+1)+\eta\,(it-2) \tag{3}$$

("translational" means that the orbit of *any* point under this motion is a translate of the ellipse). In order to construct a linkage that realizes this motion, we want to decompose it into simpler motions, namely into revolute motions; the latter ones correspond exactly to those motions that we can realize by a single (rotational) joint. We find [1, Lemma 4.3] that each linear motion polynomial, whose orbits are bounded, represents a revolute motion. Therefore, the desired decomposition is obtained by a factorization of P into linear polynomials.

In our example, however, one can check (e.g., by an ansatz with undetermined coefficients) that such a factorization does not exist. On the other hand, by the equivalence relation $\sim$ defined in (2), the motion polynomial $RP\in\mathbb{K}[t]$ describes the same motion as P for any $R\in\mathbb{R}[t]$. Here we can take $R=t^2+1$ and observe that RP indeed admits a factorization into linear polynomials:

$$R(t)\cdot P(t)=\big(t+i-\eta\,i\big)\cdot\big(t-i+\tfrac{1}{2}\eta\,i\big)\cdot\big(t-i+\tfrac{3}{2}\eta\,i\big)\cdot(t+i). \tag{4}$$

In [1, Theorem 5.15] we show that for any (bounded) motion polynomial P such a real polynomial R exists, and we give an algorithm to compute R and the complete factorization of RP.

The factorization (4) allows us to construct a linkage, in the form of an open chain (left part of Figure 1), whose links move according to the revolutions represented by the linear factors. Since such a linkage has many degrees of freedom, we need to constrain its mobility by adding more links and joints (right part of Figure 1). This is achieved by an iteration of the so-called flip procedure [1, Sections 6–7]. A minor simplification yields the final linkage, depicted in Figure 2.

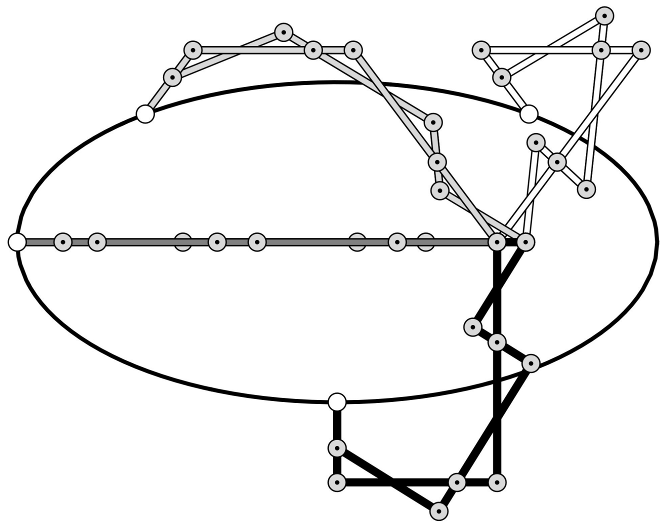

Figure 2: The linkage that draws an ellipse. The same linkage is shown in different positions: $t = 2$ (white), $t = \frac{1}{2}$ (light gray), $t = 0$ (dark gray), and $t = -1$ (black).

2 The Mathematica package PlanarLinkages

We now give a brief demonstration of our Mathematica package PlanarLinkages. The package, its source code, and a Mathematica notebook with some sample computations are freely available [3].

In[1]:= << **PlanarLinkages.m**

Motion polynomials are input using the special symbol **eta** and Mathematica's **NonCommutativeMultiply** (written as double star **). As output we obtain a pretty-printed version of the motion polynomial, which internally is represented as a Mathematica expression with head **MP**.

In[2]:= $\boldsymbol{P = t + \mathrm{I} + \mathrm{eta} ** (2 - \mathrm{I})}$

Out[2]= $(\mathbb{i} + t) + \eta \cdot (2 - \mathbb{i})$

In[3]:= $\mathbf{FullForm}[\boldsymbol{P}]$

Out[3]= $\mathrm{MP}[\mathrm{Plus}[\mathrm{Complex}[0,1], t], \mathrm{Complex}[2,-1]]$

Arithmetic can be done in the usual way, by taking into account the noncommutative multiplication.

In[4]:= $\boldsymbol{P + 1 + \mathrm{eta} ** \mathrm{I}}$

Out[4]= $((1 + \mathbb{i}) + t) + \eta \cdot 2$

In[5]:= $\boldsymbol{P ** (1 - \mathrm{eta}) ** P}$

Out[5]= $\left(-1 + 2\,\mathbb{i}\,t + t^2\right) + \eta \cdot \left(-1 + (4 - 2\,\mathbb{i})\,t - t^2\right)$

When executing the multiplication symbolically, we recover Equation (1):

In[6]:= $\mathbf{MP}[\boldsymbol{z_1, w_1}] ** \mathbf{MP}[\boldsymbol{z_2, w_2}]$

Out[6]= $z_1\,z_2 + \eta \cdot (\mathrm{Conjugate}[z_1]\,w_2 + w_1\,z_2)$

The command **ActR2** performs the action of an element of $\mathbb{K}$, which itself represents a direct isometry in SE_2, on a point $(a,b) \in \mathbb{R}^2$, see [1, (4.2)].

In[7]:= $\mathbf{Simplify[ActR2[(x_1 + I\,x_2) + eta ** (y_1 + I\,y_2),\ \{a, b\}],\ Element[\{x_1, x_2, y_1, y_2\},\ Reals]]}$

Out[7]= $\left\{ \dfrac{a\,x_1^2 - a\,x_2^2 - 2\,b\,x_2\,x_1 + x_1\,y_1 - x_2\,y_2}{x_1^2 + x_2^2}, \dfrac{2\,a\,x_2\,x_1 + b\,x_1^2 - b\,x_2^2 + x_1\,y_2 + x_2\,y_1}{x_1^2 + x_2^2} \right\}$

The command **AnimateMP** visualizes the action of a motion polynomial. Since a linear bounded motion polynomial corresponds to a revolute motion, we can compute its fixed point.

In[8]:= $\mathbf{FixPoint[t + I + eta ** (2w)]}$

Out[8]= $\{-\mathrm{Im}(w), \mathrm{Re}(w)\}$

Next, we provide a command to compute a factorization of a motion polynomial into linear factors.

In[9]:= $\mathbf{FactorMP[(t + I)\hat{\ }5 + eta ** t]}$

Out[9]= $\left((\mathbb{i} + t) + \eta \cdot \dfrac{\mathbb{i}}{16}\right) \cdot \left((\mathbb{i} + t) - \eta \cdot \dfrac{\mathbb{i}}{8}\right) \cdot \left((\mathbb{i} + t) + \eta \cdot 0\right) \cdot \left((\mathbb{i} + t) + \eta \cdot \dfrac{\mathbb{i}}{8}\right) \cdot \left((\mathbb{i} + t) - \eta \cdot \dfrac{\mathbb{i}}{16}\right)$

In[10]:= **Expand[%]**

Out[10]= $\left(t^5 + 5\,\mathbb{i}\,t^4 - 10\,t^3 - 10\,\mathbb{i}\,t^2 + 5\,t + \mathbb{i}\right) + \eta \cdot t$

If the polynomial itself cannot be factored, then the command automatically determines a minimal-degree real polynomial such that the product of the two polynomials factors completely.

In[11]:= $\mathbf{fact = FactorMP[t\hat{\ }2 + 1 + eta ** (-2 + I\,t)]}$

FactorMP::R : Multiply the input with $R = 1 + t^2$

Out[11]= $\left((\mathbb{i} + t) + \eta \cdot \left(\mathrm{C}[2] - \dfrac{\mathbb{i}}{2}\right)\right) \cdot \left((-\mathbb{i} + t) - \eta \cdot \mathrm{C}[2]\right) \cdot \left((-\mathbb{i} + t) + \eta \cdot \left(\mathrm{C}[1] + \dfrac{3\mathbb{i}}{2}\right)\right) \cdot \left((\mathbb{i} + t) - \eta \cdot \mathrm{C}[1]\right)$

This factorization can now be used to construct a linkage; we instantiate the free parameters C[1] and C[2], and give a random polynomial as second argument according to [1, Lemma 7.5]. In the output the links are labeled with integers from 1 to 10. Each triple $\{i, j, p\}$ stands for "link i is connected to link j by a joint and their relative motion is given by the motion polynomial p".

In[12]:= $\mathbf{ConstructLinkage[fact\ \ /.\ \{C[1] \to 0, C[2] \to -I/2\}, t + (9/5)\,I + eta ** 0]}$

Out[12]= $\big\{\{1, 2, (\mathbb{i} + t) + \eta \cdot 0\}, \{2, 3, (t - \mathbb{i}) + \eta \cdot \tfrac{3\mathbb{i}}{2}\}, \{3, 4, (t - \mathbb{i}) + \eta \cdot \tfrac{\mathbb{i}}{2}\}, \{4, 5, (\mathbb{i} + t) - \eta \cdot \mathbb{i}\},$
$\{6, 7, (\mathbb{i} + t) - \eta \cdot \tfrac{45\mathbb{i}}{56}\}, \{7, 8, (t - \mathbb{i}) + \eta \cdot \tfrac{3\mathbb{i}}{8}\}, \{8, 9, (t - \mathbb{i}) + \eta \cdot \tfrac{41\mathbb{i}}{28}\}, \{9, 10, (\mathbb{i} + t) + \eta \cdot \tfrac{2\mathbb{i}}{7}\},$
$\{1, 6, (t + \tfrac{9\mathbb{i}}{5}) - \eta \cdot \tfrac{9\mathbb{i}}{28}\}, \{2, 7, (t + \tfrac{9\mathbb{i}}{5}) - \eta \cdot \tfrac{9\mathbb{i}}{8}\}, \ldots, \{5, 10, (t + \tfrac{9\mathbb{i}}{5}) + \eta \cdot 0\}\big\}$

When we feed this output into the command **ShowLinkage**, we have many options to visualize and animate the linkage, in 2D (as in Figure 2) or in 3D. We refer to our website [3] for more examples.

References

[1] Matteo Gallet, Christoph Koutschan, Zijia Li, Georg Regensburger, Josef Schicho, and Nelly Villamizar. Planar linkages following a prescribed motion. *Mathematics of Computation*, 2016. To appear (preprint on arXiv:1502.05623), DOI: 10.1090/mcom/3120.

[2] Alfred B. Kempe. On a general method of describing plane curves of the n^{th} degree by linkwork. *Proceedings of the London Mathematical Society*, s1-7(1):213–216, 1876.

[3] Christoph Koutschan. Mathematica package PlanarLinkages and electronic supplementary material for the paper "Planar linkages following a prescribed motion", 2015. Available at `http://www.koutschan.de/data/link/`.