Introduction to kinematics of manipulators using Maple

Roman Hašek
e-mail: hasek@pf.jcu.cz
Department of mathematics
University of South Bohemia
České Budějovice, 371 15
Czech Republic

Abstract

The paper introduces several examples of the utilization of the computer algebra system Maple in selected courses at the University of South Bohemia. These examples are devoted to the notion of geometric transformations and they span the range from planar mappings like congruences, affine maps and inversion to the kinematics of serial robot-manipulators, specifically to the forward and inverse kinematic problems. We will see that the kinematics of simple mechanisms - at the beginning we can restrict ourselves to planar mechanisms - offers real and attractive problems, the solving of which require mostly basic knowledge of linear algebra, geometry, differential geometry and methods of solving systems of nonlinear algebraic equations.

Solving of relatively simple real-world problems that require the combination of various mathematical knowledge plays an ever more important part in mathematics education. Computer algebra systems (CAS in what follows) enable teachers to bring various real-world problems into classrooms. These programmes provide students and teachers with the means to develop symbolic representation of such problems, model them graphically and solve them symbolically or numerically. Such a process, closed with the solution and a functional model, is in our opinion very beneficial to developing true understanding of all utilized mathematical terms. Suitable use of CAS relieves students of intricate computation and formulae manipulation and gives them the opportunity to concentrate on the basic facts.

The paper introduces several examples of the utilization of the computer algebra system Maple in selected courses at the University of South Bohemia. These examples are devoted to the notion of geometric transformations and they span the range from planar mappings like congruences, affine maps and inversion to the kinematics of serial robot-manipulators, specifically to the forward and inverse kinematic problems. The common features of the examples are the application of matrix algebra and emphasis on a graphical output. Their objectives reach from the simple illustration of the graphical means of Maple (Example 1, Figure 3) over showing its use in the building of models of robot-manipulators (Example 4, 5); to the demonstration of the benefits of Maple's symbolic and numerical capabilities (Examples 2, 3, 6, 7). The codes of all examples run in Maple 9.5 and higher versions of Maple.

We will see that the kinematics of simple mechanisms - at the beginning we can restrict ourselves to planar mechanisms - offers real and attractive problems, the solving of which require mostly basic knowledge of linear algebra, geometry, differential geometry and methods of solving systems of nonlinear algebraic equations.

The given examples are introduced in several university courses that are intended for students of the mathematics teacher training programmes and for students of computer science and information technologies. Some of the problems can be met in basic courses of Linear algebra and Geometry, where they serve as real-world problems to build and fasten the understanding of the notion of geometry mapping and the connection between affine mappings and matrices. The other problems are solved in Geometry courses devoted to differential geometry and kinematics or in the specialized course of problem solving in Maple. The most utilized version of Maple in these courses is Maple 9.5.

1 Introduction to geometric transformations

1.1 Affine transformations

Geometry and Linear algebra are closely interconnected and both branches are essential in many fields, including computer graphics, scientific computation and robotics [Farin, 1998]. Understanding the correspondence between matrices and affine transformations can help to clarify the meaning of such key notions from Linear algebra and Geometry such as the linear map, affine map, image and kernel of an affine map, composition of affine maps and transformation of the coordinate system. Then the application of matrices, for instance in computer graphics or in programming, becomes understandable and natural for students of computer science and information technologies. Our experience in the teaching of Linear algebra and Geometry indicates that the best means to establish the right knowledge of correspondence between Linear algebra and Geometry is some interactive application example that enables students to experience the control of a process.

The following are two examples of the above. The first one we use to illustrate the meaning of the notions linear and affine map and to explain the role of the matrix of a linear transformation.

Example 1 Transformation of a letter. *Consider a set of points in the plane of some clear shape, for example the letter ”F”. Define an affine transformation of the plane and apply it to the set.*

The Maple code of the solution (see [1]):

At the beginning we will open the necessary packages:

```
> with(plots):  with(linalg):
```

The set of points that will be transformed - the letter ”F” in our case - is defined as a vector of sequences of points. Its name is `F` and the corresponding plot structure is named `F_g`:

```
> F:=[seq([0,k/1000],k=0..500),seq([k/1000,0.3],k=0..150),
  seq([k/1000,0.5],k=0..230)]:
> F_g:=pointplot(F,color=red, scaling=constrained):
```

An affine map $X' = M \cdot X + T$ is a composition of a linear map, given by the matrix `M` and a translation, given by the matrix (or a column vector) `T`. To apply this general affine map to the given set of points we will use the function `AffineMap` that is based on the `map` command:

```
> AffineMap:=X->map(x->evalm(M.x+T),X);
```

Now we can define the concrete matrices to represent translation `T` and the linear map `M`:

```
> M := matrix([[1, 1/4], [1/2, 1/2]]);
```

$$M := \begin{bmatrix} 1 & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

```
> T := [1/10, 1/20];
```

$$T := \left[\frac{1}{10}, \frac{1}{20}\right]$$

The image of our letter ”F” in the map `AffineMap` is the value of the function `AffineMap(F)`. The corresponding plot structure is named `Tr_F_g`.

```
> Tr_F_g:=pointplot(AffineMap(F),color=blue, scaling=constrained):
```

Finally, we can plot the original set - the red straight letter ”F” together with its image - the blue transformed letter ”F” (See Figure 1.1).

```
> display(F_g,Tr_F_g, axes=none, scaling=constrained);
```

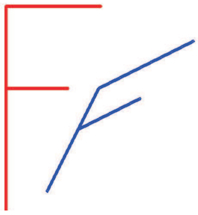


Figure 1: Transformation of the letter F

It is very beneficial to let students, after a brief introduction to the worksheet, make their own experiments with the transformation - i.e. with the matrices `M` and `T`. Then we can give them some images of transformations (see Figure 2) of the letter ”F” and challenge them to find the corresponding affine maps.

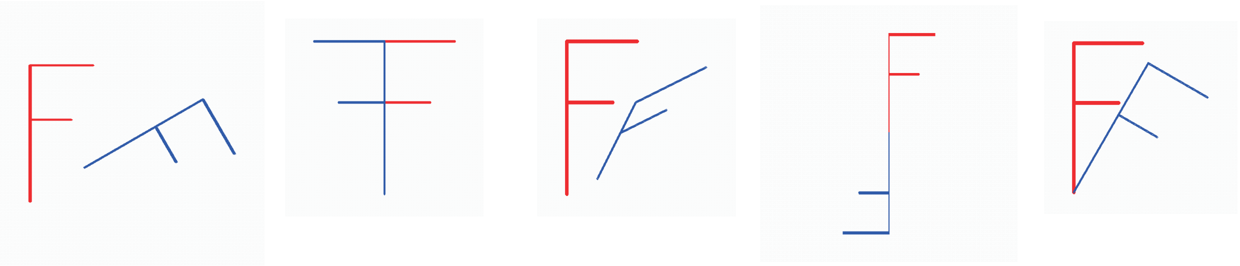


Figure 2: Find the corresponding affine maps

1.2 Circle inversion

We use the same method as for the introduction into the affine maps to show how the circle inversion works (Figure 3). For many students it is very difficult to accept the behaviour of the Inversion. Maple helps them to examine the properties of this transformation and to break the mental stereotypes that block their acceptance of such nonlinear mapping. The Maple code of the solution is almost the same as the code of the above affine map with the exception that after the establishment of the reference circle by the command

```
> u:=x^2+y^2-1=0;
```

we take advantage of the Maple command `transform` from the package `plottools` to define the circle inversion mapping `CircInv`:

```
> CircInv:=plottools[transform]((x,y)->[x/(x^2+y^2),y/(x^2+y^2)]);
```

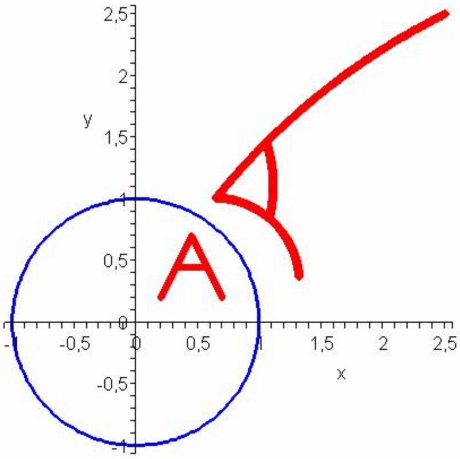


Figure 3: The Circle Inversion

The Maple code of the solution is here, or see [2].

1.3 Kinematic geometry

The next example combines the ability to apply an affine map with the very elementary knowledge of differential geometry. Students solve it to prove that they have mastered these issues together with the application of Maple in the course Solving problems using Maple.

Example 2 *Draw an arbitrary curve given by the vector-valued function. Define some figure (n-angle) and animate a motion of this figure along the curve. The figure should respect the tangent direction to the curve.*

Selected commands from the Maple code of the solution:

As a guide curve, for instance, we use the Bernoulli's lemniscate with the following parametric expression:

```
> X:=[10*cos(t)/(1+sin(t)^2),8*sin(t)*cos(t)/(1+sin(t)^2)];
```

$$X := \left[\frac{10\cos(t)}{1+\sin(t)^2}, \frac{8\sin(t)\cos(t)}{1+\sin(t)^2}\right]$$

For the later plotting we generate the plot structure of this curve:

```
> X_g:=plot([X[1],X[2],t=0..2*Pi],scaling=constrained):
```

As a figure that should be moved along the curve we use an 8-angle shaped into the letter T. First we define the home position of the letter through the coordinates of its vertices (see Figure 4). We assign the list of coordinates of these eight consecutive vertices, followed with the coordinates of the first of them, to the variable `Letter_T0`.

```
> Letter_T0:=[[2,0],[2,5],[4,5],[4,7],[-2,7],[-2,5],[0,5],[0,0],
  [2,0]]:
```

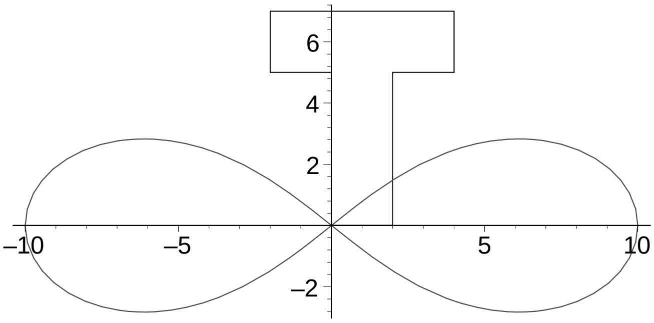


Figure 4: The letter T in its home position

As we know, the instantaneous position of the letter, corresponding to a specific value of the parameter t, can be described by the affine map $X' = M(t) \cdot X + T(t)$. Then $M(t)$ is the matrix of revolution of the tangent vector $\vec{v}(t)$ to the curve at the contact point $X(t)$ between the letter and the curve. The entries of the matrix $M(t)$ are simply direction cosines of the tangent vector $\vec{v}(t)$ that are the coordinates of the normalized tangent vector $\vec{v}_n(t) = \dfrac{\vec{v}(t)}{|\vec{v(t)}|}$:

$$M(t) = \begin{bmatrix} \cos\varphi & \sin\varphi \\ \sin\varphi & -\cos\varphi \end{bmatrix} = \begin{bmatrix} \cos\varphi & \cos(\frac{\pi}{2}-\varphi) \\ \cos(\frac{\pi}{2}-\varphi) & -\cos\varphi \end{bmatrix} = \begin{bmatrix} \frac{v_1}{|\vec{v}|} & \frac{v_2}{|\vec{v}|} \\ \frac{v_2}{|\vec{v}|} & -\frac{v_1}{|\vec{v}|} \end{bmatrix}$$

The translation $T(t)$ is defined by the radius vector of the point $X(t)$, therefore we can write that $T(t) = X(t)$.

```
> v:=diff(X,t);
```

$$v := \left[-\frac{10\sin(t)}{1+\sin(t)^2} - \frac{20\cos(t)^2\sin(t)}{(1+\sin(t)^2)^2}, \frac{8\cos(t)^2}{1+\sin(t)^2} - \frac{8\sin(t)^2}{1+\sin(t)^2} - \frac{16\sin(t)^2\cos(t)^2}{(1+\sin(t)^2)^2}\right]$$

```
> vn:=map(x->(1/norm(v,2)*x),v):
> M:=matrix([[vn[1],vn[2]],[vn[2],-vn[1]]]):
```

```
> T:=X:
```

Using the `map` command we transform the fixed set of points `Letter_T0` into the moving `Letter_T`, whose relative position to the `Letter_T0` depends on the parameter t.

```
> Letter_T:=map(x->evalm(M&*x+T),Letter_T0):
```

To animate the parametric motion of the letter T (`Letter_T`) along the given curve (`X_g`) we take advantage of the option `insequence=true` of the command `plots[display]`.

```
> Letter_T_g:=display(seq(pointplot(eval(Letter_T,t=u/10),style=line,
  color=red),u=0..63),insequence=true,scaling=constrained):
> display([X_g,Letter_T_g]);
```

The result of the later command `display` is shown in Figure 5. To get the animation menu we simply click on the plot. Then we press the button Play to play the animation.

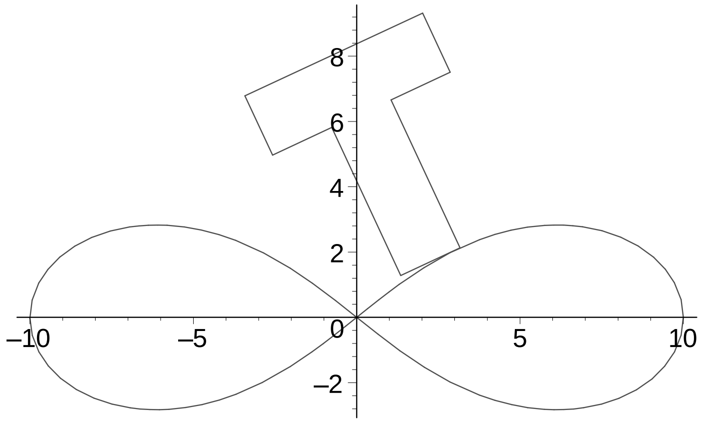


Figure 5: The letter T moves along the Bernoulli's lemniscate

The Maple code of the solution is here, or see [3].

1.4 Projects

The course Solving problems using Maple is concluded by submission of student projects. Some of them are devoted to congruent transformations in 2D or 3D.

1.4.1 Planar motion - Cycloids

Let's look at the complete assignment of the project Cycloids to introduce the way that such projects are given. It contains setting of a problem, a brief introduction to the subject of the problem and a short Maple reference to provide some hints.

Student project - CYCLOIDS

Cycloid. Consider a circle of radius r which is rolling along another circle of radius R or along a straight line. Then the point firmly connected to the moving circle at distance d from its centre draws a curve of the name cycloid. According the relation between d and r we speak about a curtate (contracted) cycloid ($d < r$) or about a prolate one ($d > r$). If the circle is rolling outside another circle we call the given curve the epicycloid. If it is rolling inside another one we call it the hypocycloid.

PROBLEM: Plot the epicycloids and hypocycloids mentioned above. Choose your own values of R and r. Consider the both cases $d < r$ and $d > r$. Animate the corresponding rolling.

Maple reference. We can take advantage of homogeneous coordinates. Using them planar transformations are represented by square matrices $(3, 3)$ and compositions of transformations are produced by multiplication of the matrices. First define the matrix of revolution `Rev(`α`)` and the matrix of translation `T(t1,t2)` as functions of variable parameters. Then assign the initial location of the point A (cycloid is the locus of this point); $A = \langle 1, d, 0\rangle$. Finally transform the coordinates of the point A using the convenient composition of transformations `Rev(`α`)` and `T(t1,t2)` with the common parameter t.

```
> Rev:=alpha->matrix([[1,0,0],[0,cos(alpha),-sin(alpha)],
  [0,sin(alpha),cos(alpha)]]);
```

$$Rev := \alpha \to \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{bmatrix}$$

```
> T:=(t1,t2)->matrix([[1,0,0],[t1,1,0],[t2,0,1]]);
```

$$T := (\mathit{t1},\, \mathit{t2}) \to \begin{bmatrix} 1 & 0 & 0 \\ \mathit{t1} & 1 & 0 \\ \mathit{t2} & 0 & 1 \end{bmatrix}$$

```
> A:=vector([1,d,0]);
```

$$A := [1,\, d,\, 0]$$

One potential result of the project - the epicycloid corresponding to the parameter values $R = 10$, $r = 5$ and $d = 8$ is presented in Figure 6.

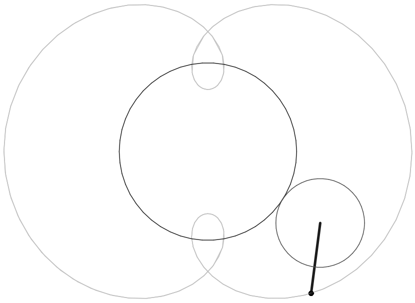


Figure 6: Epicycloid

The Maple code of the solution is here, or see [4].

1.4.2 Spatial motion - Möbius strip

Another project to prove the knowledge of spatial transformations and the use of Maple deals with the kinematic creation of the Möbius strip (Figure 7). It can be understood as a surface which is formed by the motion of a line segment. Then the task of the project is to find such a rigid body motion which by the changing of its parameter *t* moves a line segment of the $2d$ length from its initial position along the Möbius strip with the interior radius *R*.

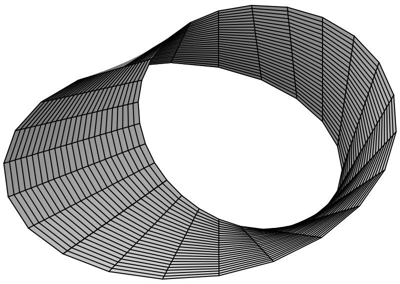


Figure 7: Möbius strip

The Maple code of the solution (see [5]):

```
> restart:
```

First we define the general transformation matrices `Rev_x`, `Rev_y` and `Rev_z` of revolution about axes x, y and z, respectively, and the matrix `Tr` of translation by the vector $(t1, t2, t3)$.

```
> Rev_x:=t->matrix([[1,0,0,0],[0,1,0,0],[0,0,cos(t),-sin(t)],
  [0,0,sin(t),cos(t)]]);
> Rev_y:=t->matrix([[1,0,0,0],[0,cos(t),0,-sin(t)],[0,0,1,0],
  [0,sin(t),0,cos(t)]]);
> Rev_z:=t->matrix([[1,0,0,0],[0,cos(t),-sin(t),0],
  [0,sin(t),cos(t),0],[0,0,0,1]]);
> Tr:=(t1,t2,t3)->matrix([[1,0,0,0],[t1,1,0,0],[t2,0,1,0],
  [t3,0,0,1]]);
```

Then, we derive the matrix M of the transformation which corresponds to the Möbius strip.

```
> M:=Rev_z(t).Tr(R,0,0).Rev_y(t/2);
> M:=evalm(M);
```

Initial position of the strip is assigned to the variable `Strip0`.

```
> Strip0:=expand([1,-d,0,0]+s*[0,2*d,0,0]);
```

Then the parametric expression `Striph` of the strip in homogenous coordinates is derived by the command

```
> Striph:=evalm(M.Strip0);
```

and its parametric expression in the cartesian coordinates is assigned to the variable `Strip` by the following command.

```
> Strip:=LinearAlgebra[SubVector](Vector(Striph),[2,3,4]);
```

Assigning values to the parameters *R* (interior radius of strip) and *d* (half width of the strip)

```
> R:=10: d:=5:
```

gives us the parametric expression (vector-valued function) of the concrete strip.

```
> Strip;
```

$$[10cos(t) + cos(t)cos(t/2)(10s - 5), 10sin(t) + sin(t)cos(t/2)(10s - 5), sin(t/2)(10s - 5)]$$

The last command plots the given strip. The result is shown in Figure 7.

```
> plot3d(Strip,s=0..1,t=0..2*Pi,scaling=constrained);
```

2 Kinematics of serial robot-manipulators

The graphical and programming tools of Maple enable us to build both planar and spatial simple movable models of respective manipulators.

The forward and the inverse kinematic problems refer to the relationship between joint variables (angles or lengths) and the position and orientation of the end-effector of a robot-manipulator (see Figure 8). The velocity problem deals with the relationship between the instantaneous velocity of the end-effector and the joint velocities of a manipulator. Examples related to these very basic problems in robotics represent real and attractive problems the solving of which requires mostly basic knowledge of linear algebra, geometry, differential geometry and methods of solving systems of nonlinear algebraic equations.

The following examples are introduced in courses of Geometry and Algebra and in the optional subject Solving problems using Maple, which is one of those preparing students of teaching to use CAS programmes in education.

2.1 A serial robot-manipulator

Kinematics of robot-manipulators deals with the study of their motions, independent of the forces which cause them. Serial robot-manipulator (RM) can be introduced as an open kinematic chain of rigid links connected together by movable joints. The mutual motions of the links are mostly elementary (rotational or translational) and are independent of one another. More complicated mutual motions of links can be produced from these elementary motions. Therefore we can distinguish only two basic types of joints (or corresponding axes), *rotational*, denoted by R, and *prismatic*, denoted by P and also called translational or sliding. Each joint represents the interconnection between two links. Thus, the relative configurations (relative displacement between adjacent links) of each pair of adjacent links of the robot are determined by values of joint variables θ_i - angles for revolution joints and linear displacements for prismatic ones (although, the variable for the prismatic joint is often denoted by a special character, for example d). Then, the configuration (position and orientation) of the end-effector in $SE(3)$ (or $SE(2)$ in the case of a planar manipulator) of a serial robot with n-joints can be expressed by a vector of θ_i, with $i = 1, 2, ..., n$.

To describe the configuration of types of successive n joints in a RM we use a n-tuple of components R or P with the corresponding order. For example a serial robot-manipulator with all three axes rotational can be assign in brief by the 3-tuple RRR and a serial manipulator with four axes, first and third rotational, second and fourth prismatic, can be given by the 4-tuple RPRP.

2.2 The forward kinematic problem

The forward kinematic problem deals with the finding of the position and orientation of the end-effector if values of the joint variables (angles or lengths) are given.

Example 3 Position of the end-effector *Consider a planar robot-manipulator with three revolute joints and two movable links of lengths l_1 and l_2 (Figure 8). Find an explicit formula which relates values of the joint variables to the position $E = [x_E, y_E]$ of the end-effector with respect to the system of coordinates x, y connected to the base of the end-effector.*

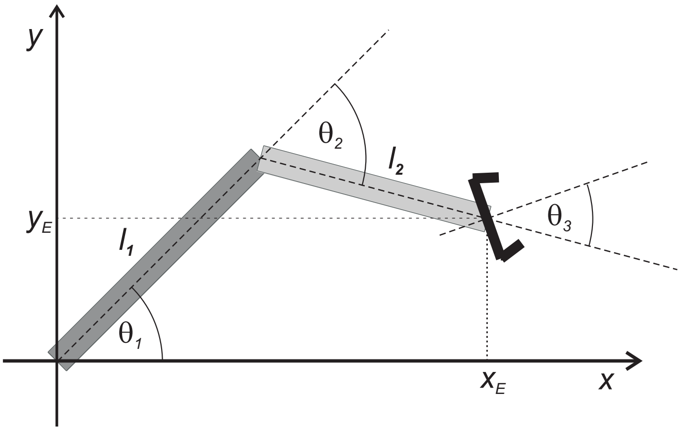


Figure 8: Planar RRR manipulator

To solve the example we can apply two approaches, both representing attractive application of mathematics subject matter. First, we can simply use trigonometric functions. Second, considering the kinematic features of the problem, we can use the composition of the successive link transformations. It is a nice illustration of a utilization of matrices in geometry, application of homogeneous coordinates, homogeneous transformation matrices and matrix algebra (Many similar exercises can be found in [Cox, 1998], [Craig, 1986], [Tsai, 1999]).

Solution 1 - The trigonometric approach
To derive relations between joint variables θ_1, θ_2, θ_3 and the coordinates $[x_E, y_E]$ of the end-effector we simply apply definitions of trigonometric functions sine and cosine (see Figure 9)

$$E(\theta_1, \theta_2) = [l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2), l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2)].$$

Orientation of the end-effector is given by the sum of joint variables - angles θ_1, θ_2, θ_3 of mutual rotation of successive joints

$$\theta = \theta_1 + \theta_2 + \theta_3.$$

The introduced trigonometric approach to the solution of the forward kinematic problem fails in case of more complex spatial mechanisms. Moreover it does not provide us with information about positions of individual links and about their mutual configurations that are crucial to the full control of a manipulator motion.

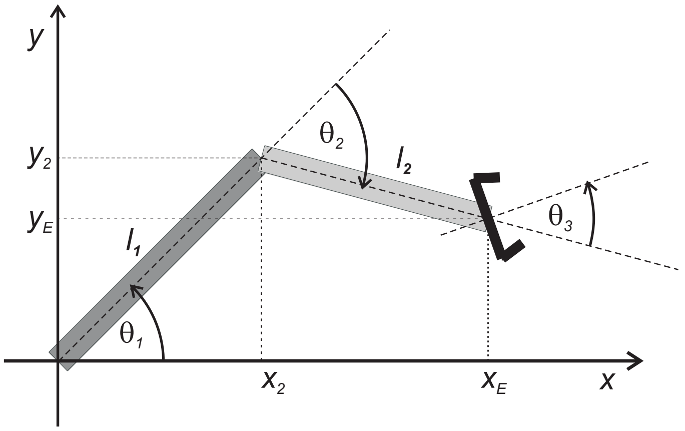


Figure 9: Planar RRR manipulator - joint variables

Solution 2 - The kinematic approach
We will derive transformation of the frame firmly connected to the end-effector by way of a composition of the successive link transformations (Figure 10). The Maple code of the solution is here, or see [6].

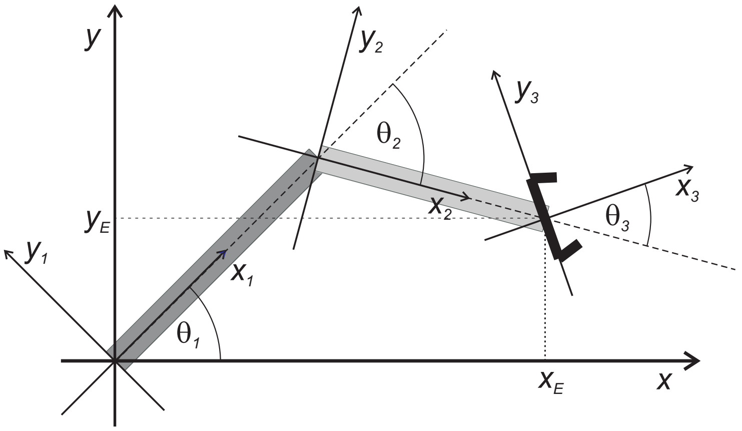


Figure 10: Planar RRR manipulator - frames attached to links

Consider the homogeneous transformation matrices of revolution $R(\theta)$ and translation $T(\vec{t})$

$$R := \theta \to \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix}, \qquad T := t \to \begin{bmatrix} 1 & 0 & 0 \\ t_1 & 1 & 0 \\ t_2 & 0 & 1 \end{bmatrix}.$$

We shall compute the individual transformation matrices A01, A12, A23 which relates the neighboring links (the coordinate systems attached to these links - see Figure 10).

```
> A01:=evalm(R(theta1)):
> A12:=evalm(T([l1,0]).R(theta2)):
```

```
> A23:=evalm(T([l2,0]).R(theta3)):
```

Then the transformation matrix `A` that performs the end-effector motion is the result of the multiplication of the above matrices `A01`, `A12` and `A23`. Entries of this resulting matrix have the form of products of trigonometric expressions comprised of the terms $\sin\theta_i$ and $\cos\theta_i$ for $i = 1, 2, 3$. To simplify them we use the function `combine` that converts these expressions into a sum of trigonometric terms the arguments of which are linear combinations of the original arguments. In our simple example the function `combine` uses this way of transformation automatically, but in cases of some more complex trigonometric expressions we could appreciate the possibility of specifying this transformation directly, using a second optional argument `trig` as shown in the following command.

```
> A:=map(x->combine(x,trig),evalm(A01&*A12&*A23));
```

$$A := \begin{bmatrix} 1 & 0 & 0 \\ \cos(\theta1)\,\mathit{l1} + \cos(\theta1+\theta2)\,\mathit{l2} & \cos(\theta1+\theta2+\theta3) & -\sin(\theta1+\theta2+\theta3) \\ \sin(\theta1)\,\mathit{l1} + \sin(\theta1+\theta2)\,\mathit{l2} & \sin(\theta1+\theta2+\theta3) & \cos(\theta1+\theta2+\theta3) \end{bmatrix}$$

To get coordinates of a reference point of the end-effector (the origin of the corresponding frame) relative to the base frame we multiply the coordinates of the base frame origin by the transformation matrix A.

```
> Xh:=Vector[column]([1,0,0]):
> X1h:=Vector[column](evalm(A.Xh));
```

$$\mathit{X1h} := \begin{bmatrix} 1 \\ \cos(\theta1)\,\mathit{l1} + \cos(\theta1+\theta2)\,\mathit{l2} \\ \sin(\theta1)\,\mathit{l1} + \sin(\theta1+\theta2)\,\mathit{l2} \end{bmatrix} \tag{1}$$

```
> E:=unapply([X1h[2],X1h[3]],[theta1,theta2]);
```

$$E := (\theta1, \theta2) \to [\cos(\theta1)\,\mathit{l1} + \cos(\theta1+\theta2)\,\mathit{l2},\ \sin(\theta1)\,\mathit{l1} + \sin(\theta1+\theta2)\,\mathit{l2}]$$

2.3 Building of live models of planar manipulators

Previous computation of the individual transformations between neighboring links implies the possibility to build live movable models of manipulators (see Figure 11).

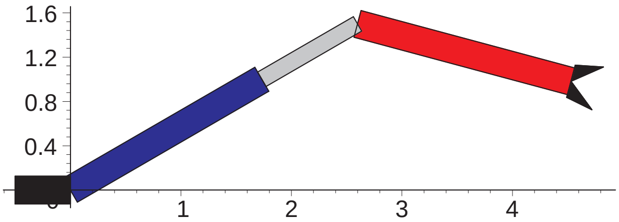


Figure 11: Model of a planar RPR manipulator

The basic building blocks of such models are the following simple pre-programmed procedures and functions (Their Maple codes are here, or see [7].):

`tool(h)`
- representation of the end-effector, parameter `h` means the height of the tool.

```
> tool:=h->plottools[polygon]([[0,0],[0,-0.6*h],[h,-0.8*h],[0,0],
  [h,0.8*h],[0,0.6*h]],color=black):
```

`link(l,w,col)`
- model of a link, parameters `l`, `w` and `col` mean the length, width and color of the link, respectively.

```
> link:=(l,w,col)->plottools[rectangle]([0,w/2],[l,-w/2],color=col):
```

`base(w)`
- imitation of the manipulator's base of the width `w`.

```
> base:=w->plottools[rectangle]([-2*w,-w/2],[0,w/2],color=black):
```

`P_joint(d,w,col)`
- model of the prismatic joint, parameters `d`, `w` and `col` mean the variable length, width and color of the joint, respectively.

```
> P_joint:=(d,w,col)->plottools[rectangle]([0,0.3*w],[d,-0.3*w],
  color=col):
```

`g(A,points)`
- function that apply a transformation given by the matrix `A` to a set of points `points`:

```
> g:=proc(A,points)
  local G01,g01,x01,y01:
  G01:=map(x->evalf(x),evalm(A&*linalg[matrix](3,1,[1,x,y]))):
  x01:=unapply(G01[2,1],x,y):
  y01:=unapply(G01[3,1],x,y):
  g01:=plottools[transform]((x,y)->[x01(x,y),y01(x,y)]):
  g01(points);
  end:
```

Example 4 Model of a planar manipulator. *Use the above procedures to build a model of the planar RRR manipulator.*

The Maple code of the solution (see [8]):

See Figure 12. Both links and the end-effector are constructed in the same manner. First we define the element (a link or the end-effector) in its starting position with a reference point in the origin of the base frame - using the `link` and `tool` procedures. Then we apply a corresponding transformation to each of these elements via the `g` function.

```
> link1:=g(R(-Pi/5),link(2,0.2,blue)):
> link2:=g(R(-Pi/5).T([2,0]).R(0.9),link(1,0.2,red)):
> end_effector:=g(R(-Pi/5).T([2,0]).R(0.9).T([1,0]),tool(1/10)):
```

Finally we display all the elements of the manipulator together.

```
> plots[display](base(0.2),link1,link2,end_effector,
  scaling=constrained);
```

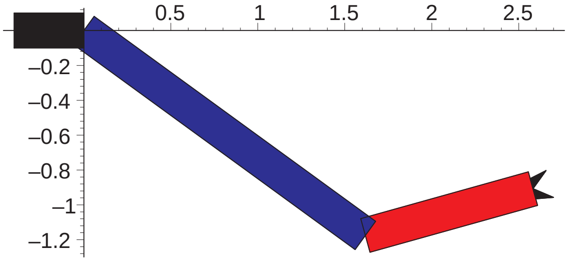


Figure 12: Model of a planar RRR manipulator

Using the foregoing procedures we can define a single command to represent a model of a planar manipulator. As we can see in the next example of the RPR planar manipulator, the parameters of such a command will be lengths of links l_i and angles θ_i between directions of successive ones.

Example 5 *Define a single command to represent the planar RPR manipulator.*

The Maple code of the solution (see [9]):

We will produce the procedure `RM_RPR(l1,l2,l3,theta1,theta2)` where the parameters `l1` and `l2` are constant and the remaining `l2`, `theta1` and `theta2` are variable. To emphasize the variable length of the second link we label it `d` instead of `l2`.

```
RM_RPR:=proc(l1,d,l2,theta1,theta2)
local link1,link2,Pjoint,end_effector,w:
w:=1/8*max(l1,l2):
link1:=g(R(theta1),link(l1,w,blue)):
Pjoint:=g(R(theta1).T([l1,0]),P_joint(d,w,grey)):
link2:=g(R(theta1).T([l1,0]).T([d,0]).R(theta2),link(l2,w,red)):
end_effector:=g(R(theta1).T([l1,0]).T([d,0]).R(theta2).T([l2,0]),
  tool(w)):
plots[display](base(w),link1,Pjoint,link2,end_effector,
  scaling=constrained):
end:
```

To plot the model of the manipulator in some specific configuration we invoke the command with relevant values of parameters. The result is shown in Figure 11.

```
> RM_RPR(2,1,2,Pi/6,-Pi/4);
```

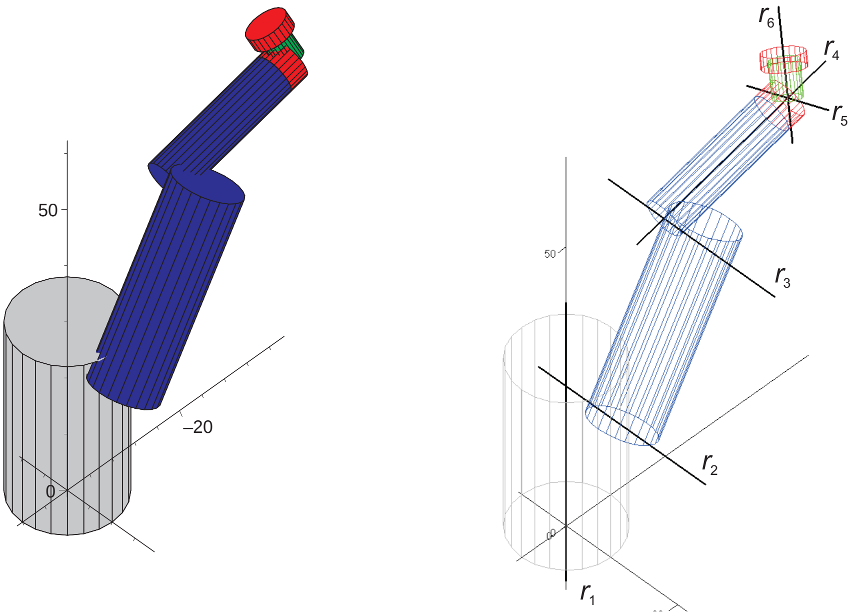
2.4 Movable 3D model of a serial robot-manipulator

Similarly to the planar model we can build a more realistic spatial model of a robot-manipulator. For example the `Robot6R` procedure enables us to represent any configuration $\theta = (\theta_1, \theta_2, ..., \theta_6)$ of a serial 6-axes robot-manipulator of the so called Puma type [Craig, 1986] as is shown in Figure 13 (a).

Such a type of a serial robot-manipulator has six rotational axes. Its architecture is given by the standard *Denavit-Hartenberg parameters* ([Angeles, 1997], [Craig, 1986]): a_i is the distance from the ith axis to the $i+1$th axis, α_i is the angle from the ith axis to the $i+1$th axis, d_i means the offset on the

ith axis and θ_i denotes the angle of rotation around the ith axis r_i (see Figure 13 (b)). In the case of the Puma type manipulator, the parameters a_i, α_i and d_i are given constants whereas the parameters θ_i are variables. Then, when the architecture of a robot-manipulator is known, the configuration (position and orientation) of the end-effector in $SE(3)$ of a serial robot with 6-joints can be expressed by the vector $\theta = (\theta_1, \theta_2, ..., \theta_6)$ of angles of revolution around the joints' axes $r_1, r_2, ..., r_6$.

```
> Robot6R(Pi/2,-Pi/6,-Pi/4,-Pi/6,Pi/2,0);
```



(a) Resulting model

(b) Its axes of revolution

Figure 13: Procedure `Robot6R`

The Maple code of the procedure Robot6R is here, or see [10].

2.5 The inverse kinematic problem

The inverse kinematic problem deals with the finding of all possible values of the joint variables if the position and orientation of the end-effector are given. Examples corresponding to the inverse kinematic problem can be used as a means of introduction to the solving of systems of nonlinear equations.

Example 6 *Find values of the joint variables θ_1, θ_2 that correspond to given position $[x_E, y_E] = [-1, 0.5]$ of the end-effector of the planar RRR manipulator with link lengths $l_1 = 1$, $l_2 = 1$.*

Solution 1 - Trigonometric equations
A corresponding manipulator is shown in Figure 8. To solve the task we can take advantage of our solution to the forward kinematic problem, for example (1). Then we can see that it is sufficient to solve the two corresponding trigonometric equations. The problem has two solutions, they are shown in Figure 14.

```
> X1:=([X1h[2],X1h[3]]);
```

$$X1 := [\cos(\theta 1)l1 + \cos(\theta 1 + \theta 2)l2,\ \sin(\theta 1)l1 + \sin(\theta 1 + theta2)l2]$$

```
> E:=[-1,1/2]; l1:=1; l2:=1;
> op(X1-E);
```

$$1 + \cos(\theta 1) + \cos(\theta 1 + \theta 2),\ -\frac{1}{2} + \sin(\theta 1) + \sin(\theta 1 + \theta 2)$$

```
> sol:=allvalues(solve({op(X1-E)},{theta1,theta2}));
```

$$sol := \left\{\theta 2 = -\arctan(\frac{\sqrt{55}}{3}) + \pi,\ \theta 1 = \arctan\left(\frac{\frac{1}{4} + \frac{\sqrt{55}}{10}}{-\frac{1}{2} + \frac{\sqrt{55}}{20}}\right) + \pi\right\},$$

$$\left\{\theta 1 = \arctan\left(\frac{\frac{1}{4} - \frac{\sqrt{55}}{10}}{-\frac{1}{2} - \frac{\sqrt{55}}{20}}\right) - \pi,\ \theta 2 = \arctan(\frac{\sqrt{55}}{3}) - \pi\right\}$$

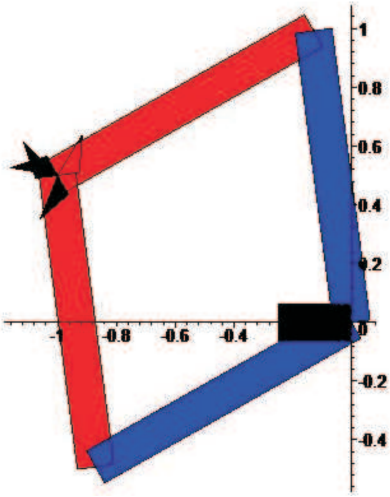


Figure 14: Two solutions to the given inverse problem

The Maple code of the solution is here, or see [11].

Although the given problem can be solved in such simple way we can use it to illustrate more general methods that are used to solve the inverse kinematic problem for more complex manipulators. To solve the system of nonlinear equations corresponding to the problem we can use Gröbner basis or the method of resultant. In the case of more general manipulators it can be necessary to solve them numerically [Tsai, 1999].

Solution 2 - Polynomial equations

Let $[a, b]$ be the coordinates of the end-effector and $s1$, $c1$, $s2$, and $c2$ be the shortened forms of the expressions $\sin\theta_1$, $\cos\theta_1$, $\sin\theta_2$ and $\cos\theta_2$, respectively. The inverse kinematic problem of the given manipulator can be expressed by the following system of four polynomial equations where $s1$, $c1$, $s2$, and $c2$ are variables and l_1, l_2, a, b are parameters.

```
> r1:=X2[1]-a;  r2:=X2[2]-b; r3:=c1^2+s1^2-1; r4:=c2^2+s2^2-1;
```

$$r1 := c1\ l1 + l2\ c1\ c2 - l2\ s1\ s2 - a$$

$$r2 := s1\ l1 + l2\ s1\ c2 + l2\ c1\ s2 - b$$

$$r3 := c1^2 + s1^2 - 1$$

$$r4 := c2^2 + s2^2 - 1$$

To find solutions of the system of polynomial equations we use the Groebner basis of the corresponding ideal. It can provide us with easily solvable equations in $s1$, $c1$, $s2$, and $c2$.

```
> GB:=Groebner[gbasis]([r1,r2,r3,r4],plex(c2,s2,c1,s1));
```

After substitution of lengths l_1, l_2 into resulting polynomials we get system of simple equations linking the joint variables with the end-effector coordinates $[a, b]$.

```
> b1; b2; b3; b4;
```

$$c2 + 1 - \frac{1}{2}\,a^2 - \frac{1}{2}\,b^2$$

$$s2 + \frac{1}{2}\,\frac{(2\,a^2 + 2\,b^2)\,s1}{a} + \frac{1}{2}\,\frac{-b\,a^2 - b^3}{a}$$

$$c1 + \frac{b\,s1}{a} + \frac{1}{2}\,\frac{-a^2 - b^2}{a}$$

$$s1^2 + \frac{(-4\,b^3 - 4\,b\,a^2)\,s1}{4\,a^2 + 4\,b^2} + \frac{a^4 + b^4 - 4\,a^2 + 2\,a^2\,b^2}{4\,a^2 + 4\,b^2}$$

The Maple code of the solution is here, or see [12].

Resolution of the inverse kinematic problem enables us to control the movement of the end-effector of a manipulator along some given trajectory.

Example 7 Movement of the end-effector. *Consider a planar RRR manipulator with link lengths $l_1 = 1, l_2 = 1$ and a curve which is given by parametric equations and lies inside the working area of the manipulator. Execute the motion of the end-effector of the manipulator along the curve.*

Solution to the example is shown in Figure 15. The Maple code of the solution is here, or see [13].

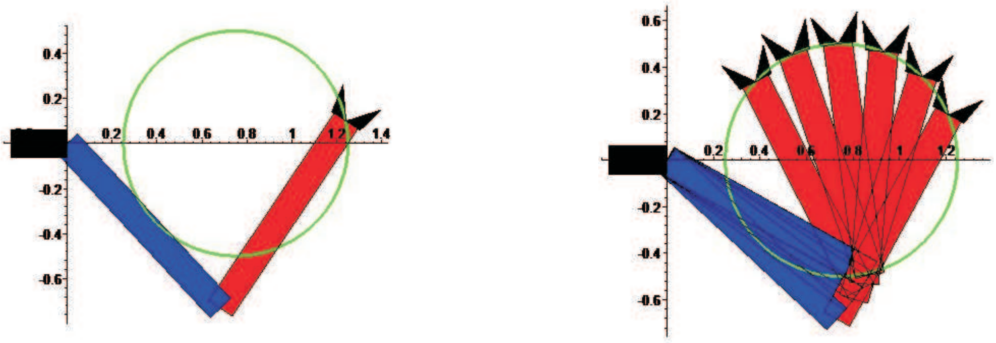


Figure 15: The end-effector movement

3 Conclusion

We have presented examples of various complexity that were all devoted to the matrix representation of geometric transformations. The greater part of the examples dealt with the basic problems of robotics. The aim of the paper was to show that the solving of such, at first glance engineering problems, can be beneficial to a wider audience of students at high schools and universities.

References

[Angeles, 1997] Angeles, J.: *Fundamentals of Robotic Mechanical Systems*, Springer-Verlag, New York, 1997.

[Cox, 1998] Cox, D., Little, J., O'Shea, D.: *Ideals, Varieties, and Algorithms - An Introduction to Computational Algebraic Geometry an Commutative Algebra*, Springer-Verlag, New York, 1998.

[Craig, 1986] Craig, John J.: *Introduction to Robotics*, Addison-Wesley Publishing Company, Inc., USA, 1986.

[Farin, 1998] Farin, G., Hansford, D.: *The Geometry toolbox for graphics and modeling*, A K Peters, Ltd., USA, 1998.

[Karger 1998] Karger, A.: *Classical Geometry and Computers*, Journal for Geometry and Graphics 2, 1998.

[Murray, 1994] Murray,R. M., Li, Z., Sastry, S. S.: *A mathematical introduction to robotic manipulation*, CRC Press LLC, USA, 1994.

[Pech, 2007] Pech, P.: *Selected Topics in Geometry with Classical vs. Computer Proving*, World Scientific Publishing Co., 2007.

[Selig, 1996] Selig, J. M.: *Geometrical Methods in Robotics*, Springer-Verlag New York, Inc., USA, 1996.

[Tsai, 1999] Tsai, Lung-Wen: *Robot analysis: the mechanics of serial and parallel manipulators*, John Wiley & Sons, Inc., New York, USA, 1999.

Supplementary Electronic Materials

[1] Hašek, R., *Transformation of a letter*, Maple 9.5 worksheet, 2007,
Transformation_F.mws

[2] Hašek, R., *Circle inversion*, Maple 9.5 worksheet, 2006,
CircleInversion_A.mws

[3] Hašek, R., *Motion of a figure along a given curve*, Maple 9.5 worksheet, 2007,
MotionAlongCurve.mws

[4] Hašek, R., *Cycloids*, Maple 9.5 worksheet, 2006,
Cycloids.mws

[5] Hašek, R., *Möbius strip*, Maple 9.5 worksheet, 2008,
MobiusStrip.mws

[6] Hašek, R., *The forward kinematic problem*, Maple 9.5 worksheet, 2006,
ForwardProblem_Kinematics.mws

[7] Hašek, R., *Building of live models of planar manipulators*, Maple 9.5 worksheet, 2006,
Procedures.mws

[8] Hašek, R., *Model of a planar manipulator with three rotational axes (RRR)*, Maple 9.5 worksheet, 2006,
PlanarRM_RRR_Model.mws

[9] Hašek, R., *Model of a planar manipulator with two rotational and one prismatic axis (RPR)*, Maple 9.5 worksheet, 2006,
PlanarRM_RPR_Model.mws

[10] Hašek, R., *3D model of a serial robot-manipulator*, Maple 9.5 worksheet, 2004,
Robot6R.mws

[11] Hašek, R., *The inverse kinematic problem - solution using trigonometric equations*, Maple 9.5 worksheet, 2007,
InverseProblem_Trig.mws

[12] Hašek, R., *The inverse kinematic problem - solution using polynomial equations*, Maple 9.5 worksheet, 2007,
InverseProblem_Polynom.mws

[13] Hašek, R., *Movement of the end-effector*, Maple 9.5 worksheet, 2007,
InverseProblem_Movement.mws