7 Transformation of functions (inverse kinematics)

7.1 Input and output functions

As introduced in chapter 1.4 (timed motion) input motion can be any function of time. In the FEM-theory introduced so far attention has been given only to input motion with constant steps, thereby neglecting time, or to input motion with constant velocity (see chapter 5.6.4). For the majority of cases this might be sufficient.

In some cases however the use of a specific input function is essential, for instance when the input motion includes start and stop behaviour of the mechanism. The function of the type $\alpha(t)$, as to be given by the user, must be made known to the computer program RUNMEC. At present the way to organize this is by means of an input file, in which the numerical values of the subsequent mechanism positions can be listed on beforehand. The layout of such files concerning Periodic and Non-periodic functions (PION) has been standardized for use within the computer programs of the CADOM-working group. For use with the Runmec program the characteristic layout has been depicted in figs 7.1.1 and 7.1.2. This layout allows specifying various types of input values, like:

- A list of input values (arguments),
- A list of timed input values (time and argument)
- A list of timed input values and derivatives (input velocity and acceleration)

To calculate the mechanism motion, implicit differentiation will be applied, see chapter 5.3.6. When no derivatives of input motion have been specified, the timed motion will be calculated with default input velocity and acceleration (the values one and zero respectively).
In case that the degree of freedom of a mechanism is higher than one, each input motion must have its own input file ("Movement"-block, keyword MOV_FUNC). These files must:

- Have the same number of points, and
- Have identical time values at comparable positions (if the time values are present).

The aspect of periodicity of the input functions deserves attention. In many cases input motion concerns just a complete revolution of a crank, which can be understood as periodic motion. As was argued in chapter 2.7 this does in general not guarantee that output motion is periodic as well. The property of periodicity can thus not be pasted to output motion automatically. To avoid misunderstanding the Runmec program assumes all input and output motion as non-periodic, unless the user specifies otherwise.

With respect to the design purpose of a mechanism a special calculation task, frequently indicated with "inverse kinematics", requires the use of input functions. Typical for inverse kinematics is the idea that a mechanism is to be designed and how to drive it needs some calculation. A certain output motion is given then, while the input motion is to be calculated. For this calculation the role of input and output needs to be reversed (inverted). Actually it is better to say that the mechanism (model) is inverted and that it concerns just kinematics of that inverted mechanism.

Note: version 4.0 of the Runmec program makes fully use of the PION functionality. The major differences regarding the previous versions are:

- Continuous descriptions (other than lists of points) can be used for input motion.
- Interpolation of input function values will be performed when necessary. It means that, for multi DOF mechanisms, input functions are not required to have the same amount of points (instead, the argument range must cover the working range of the input).
- All desired properties (attributes) of the PION function object, to be created by Runmec for output, can be specified in the mechanism input file.

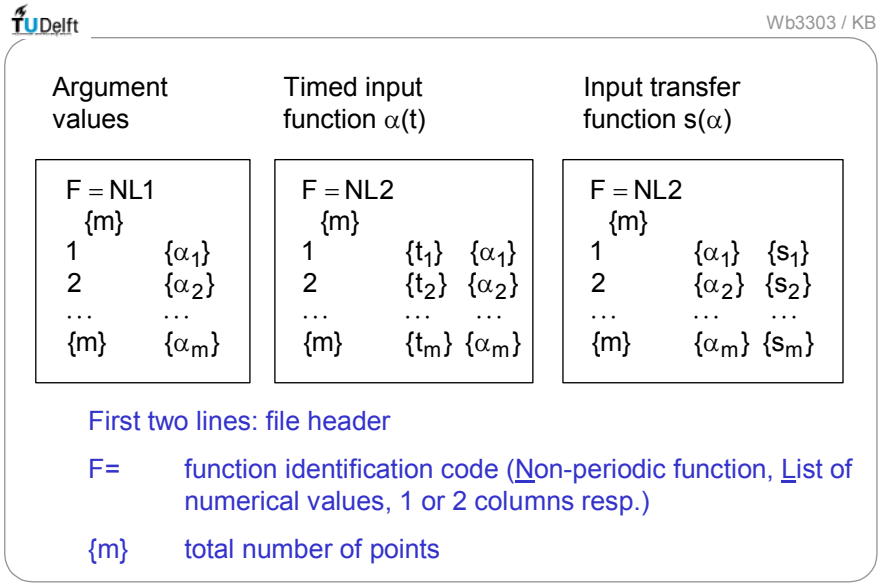

Fig. 7.1.1 Typical PION file structure, without derivatives.
{ } indicates a numerical value

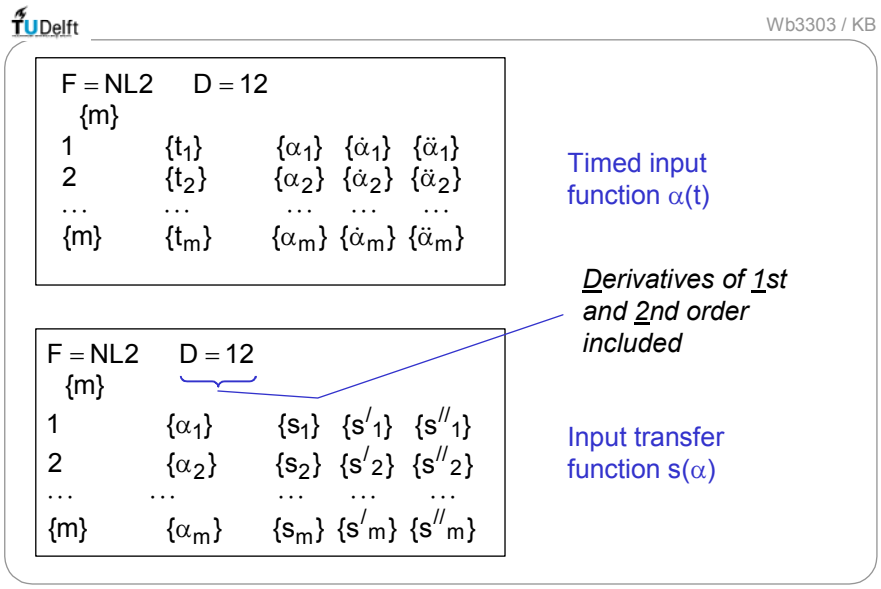

Fig. 7.1.2 Typical PION file structure, including derivatives.
{ } indicates a numerical value

In practice two kinds of inverse kinematics can be recognized:

- Calculate a timed input motion like $\alpha(t)$, possibly with its timed derivatives. This kind is typical for robotics: the input motion is to be generated by some controlled actuator. In chapter 7.2 an example will be presented.
- Calculate a transfer function like $\beta(\alpha)$, with the intention to generate the non-uniform input motion with a mechanism in a series connection. Such a mechanism is typically a cam mechanism. The cam design algorithm requires also derivatives of the design function $\beta(\alpha)$. They can be obtained using the derivatives of the input function. The implicit differentiation algorithm is precisely the same as for time derivatives. An example will be presented in chapter 7.3.

One may ask how to create PION-files. At present there are various options.

- Manually, using a plain text editor program (but this is only for simple cases or simple modifications).
- With the assistance of spreadsheet computer programs or the like, creating all kinds of mathematical functions in tabular form. The two header lines can be added later with a text editor.
- By the built-in function edit facilities of some other CADOM programs (especially the cam designing program TADSOC).
- By output of RUNMEC, using the keywords PIONOUTX or PIONOUTE (for transfer functions of co-ordinates or form parameters respectively, version 3.x).

7.2 Inverse kinematics (example: a planar robot)

The planar robot of fig. 5.3.5 will be used in this chapter for an exercise in inverse kinematics. In this robot the two cylinders are the actuators. Their lengths ℓ_1 and ℓ_2 respectively are input motion quantities, which are a function of time. Their values and time derivatives must be calculated when the output motion (point M) is prescribed.

To perform the exercise manually a simple motion for point M will be proposed: a straight line, which will be passed with constant speed. The timed functions can be specified, using c_1 and c_2 as constants, as:

$$x_M = c_1 \cdot t \quad ; \dot{x}_M = c_1 \quad ; \ddot{x}_M = 0$$

$$y_M = c_2 \cdot t \quad ; \dot{y}_M = c_2 \quad ; \ddot{y}_M = 0$$

This motion is taken as input motion for the inverse mechanism. In RUNMEC however the option to define a co-ordinate as input has not been adopted. Input motion needs a specification as a form parameter. This can be solved easily by taking an extra element in the mechanism model. A ternary element (type I), see fig. 5.4.5, can be applied such that the form parameters u and v are identical with x_M and y_M respectively. A normal kinematic analysis of the inverse mechanism provides the values of all co-ordinates and their (timed) derivatives up to order two. These co-ordinates include the ones of the input links. In this example however the co-ordinates of the input links are already given, so the first part of the kinematic analysis can be skipped. Just the timed functions of the form parameters ℓ_1 and ℓ_2 (assume binary elements here) are to be calculated. The eqs (5.31a and 5.32a) can be used for this purpose, and for this occasion they can be written out for ℓ_1 as:

$$\dot{\ell}_1 = \frac{d\ell_1}{dt} = \frac{\partial \ell_1}{\partial x^k} \cdot \frac{dx^k}{dt} = |X \;\; X \;\; \cos\beta_1 \;\; \sin\beta_1| \cdot \begin{vmatrix} 0 \\ 0 \\ c_1 \\ c_2 \end{vmatrix} = c_1 \cos\beta_1 + c_2 \sin\beta_1$$

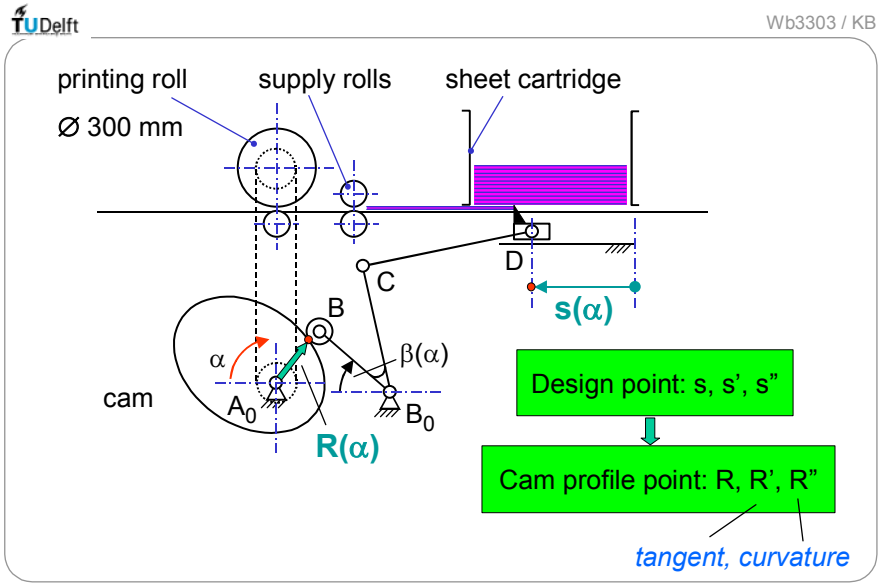

Fig. 7.3.1 Cam profile information as available from the design: discrete points and derivatives

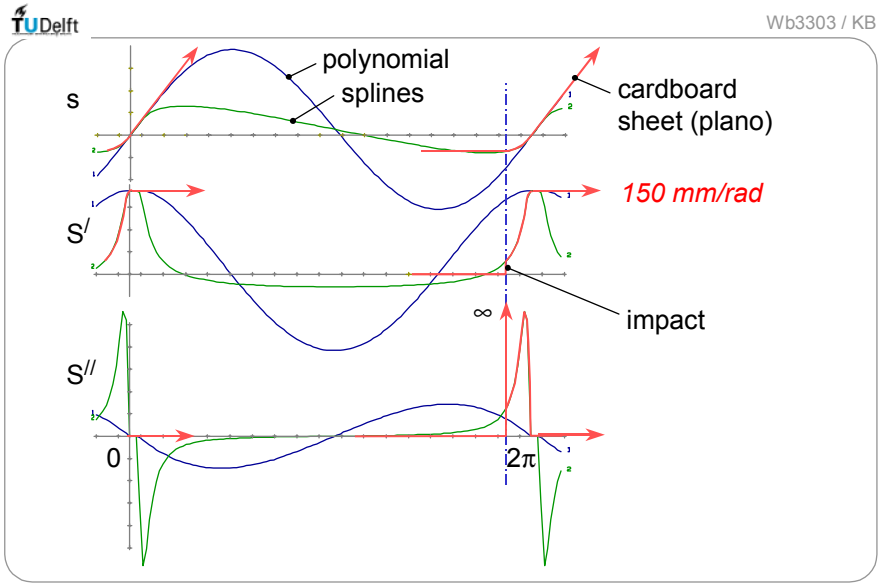

Fig. 7.3.2 Two possible design functions for supply of sheets in a printing machine

$$\ddot{\ell}_1 = \frac{d^2\ell_1}{dt^2} = \frac{1}{\ell_1}|0\ 0\ c_1\ c_2|\cdot\begin{vmatrix} X & X & X & X \\ X & X & X & X \\ X & X & \sin^2\beta_1 & -\sin\beta_1\cos\beta_1 \\ X & X & -\sin\beta_1\cos\beta_1 & \cos^2\beta_1 \end{vmatrix}\cdot\begin{vmatrix} 0 \\ 0 \\ c_1 \\ c_2 \end{vmatrix} = \frac{(c_1\sin\beta_1 - c_2\cos\beta_1)^2}{\ell_1}$$

The matrix coefficients denoted as X are not relevant because they will be multiplied with zero. The terms $\sin\beta_1$, $\cos\beta_1$ and ℓ_1 are considered here as help quantities which can be calculated from the given position (co-ordinates) of the binary element 1, see eqs. (5.2) and (5.3). For element 2 a comparable set of equations can be written out.

Applying RUNMEC it would be possible to read input files for u and v (inputs of the inverse mechanism) which contain a list of the values $(t, u, \dot{u}, \ddot{u})$ and $(t, v, \dot{v}, \ddot{v})$ respectively. Using the output keyword PIONOUTE twice, two files can be created for the motion of the two cylinders, containing a list of the values $(t, \ell_1, \dot{\ell}_1, \ddot{\ell}_1)$ and $(t, \ell_2, \dot{\ell}_2, \ddot{\ell}_2)$ respectively. To verify that the inverse calculations have been done correctly, these files can be used as input files in the original planar robot mechanism. The straight line with constant speed should be the resulting motion of point M.

7.3 Inverse kinematics (example: supply mechanism in a printing machine)

Figure 7.3.1 shows the kinematic principle of the machine, which has to print an image on a plane sheet. Typically such a sheet is a collapsed cardboard box (plano), on which the text "fragile" has to be printed. The negative of the image resides on a printing roll and will be kept moistened with ink. The sheets must be supplied with the circumferential speed and in phase with the printing roll. All motions – printing roll, supply rolls and slider element – will be derived from a central driving shaft, located near the bottom of the machine. The driving shaft in operation has constant angular velocity $\dot{\alpha}$. A cam mechanism is to be designed to generate the oscillating motion of the slider. A rocker-slider mechanism B_0CD was chosen to convert the motion of the cam follower to the slider motion.
In this design case the supply motion of the slider $s(\alpha)$ will typically be chosen by the designer and the cam profile should be calculated such that the required motion of the slider will be generated. With a given radius of the printing roll of 150 mm it is obvious that the slider must deliver the sheet with a "speed" of 150 mm/rad at the printing roll. The rest of the supply motion can be chosen more or less freely, as long as an acceptable amplitude will occur. Two characteristic proposals for the slider motion have been depicted in fig. 7.3.2. For both proposals the motion cycle starts with a short interval having constant first order motion. One proposal connects two successive cycles with just one intermediate function: a polynomial of the fifth degree. Now the result is a rather big amplitude.
The other proposal specifies that – on return half way – the "speed" is -30 mm/rad. The two gaps have been filled with certain spline functions. Based on this proposal the motions of two successive sheets have been drawn also in the figure.

Once the designer is satisfied with his design motion for the slider, the motion $s(\alpha)$, including derivatives, is known. To obtain the design motion $\beta(\alpha)$ for the cam, including the derivatives, it can be stated that

$$\beta(\alpha) = \beta(s(\alpha)) \tag{7.1}$$

$$\frac{d\beta}{d\alpha} = \frac{d\beta}{ds}\cdot\frac{ds}{d\alpha} \tag{7.2}$$

```
; case printing machine, inverted mechanism (for Runmec version 3.x)
KIN ANA
TOPOLOGY
 ELEM 1 TR1  3  4  5  6   7  ; Ternary element for input motion
      2 BIN  6  7  8  9
      3 BIN  1  2  8  9
      4 BIN  1  2  10  11    ; The cam follower link

 FORM 1 lenv 1 1  1.0
      2 lenl 2 2  1.0
      3 lenl 3 3  1.0
      4 lenl 4 4  1.0
      5 angb 3 5 -1.0
      6 angb 4 5  1.0
      7 lenu 1 6  1.0
      8 angb 4 -1 1.0
      9 angb 3 -2 1.0
 NRDOF 1

GEOMETRY
 XFIX 1  0.0
      2  0.0
      3  0.4
      4  0.25
      5  3.14159265
 XMOV 6  0.4
      7  0.25
      8  0.1
      9  0.2
     10 -0.1
     11  0.2
 PARA 1  0.0
      2  0.4
      3  0.2
      5  0.2

STORAGE
 BUFTIME 99
 BUFE0    -1    100
 BUFDE1   -1    101
 BUFDE2   -1    102
 BUFE0    -2    200 ; as test
 BUFDE1   -2    201
 BUFDE2   -2    202
 PRINTBUF

 PIONOUTE -1 9 2 camfunction2.dat  ; design function for the cam (PION-format)
;     form.par  DOF.nr   nr.of.derivatives filename
;     DOF.nr > NRDOF specifies derivatives with respect to time
 BEGINPIC
  plotfunc 99 100
 ENDPIC

 BEGINPIC
  plotfunc 99 101
  plotfunc 99 102
 ENDPIC

MOVEMENT
 MOV_FUNC doelf2.dat  ; file with required slider motion (PION-format)
END
```

Table 7.1

$$\frac{d^2\beta}{d\alpha^2} = \frac{d^2\beta}{ds^2} \cdot \left(\frac{ds}{d\alpha}\right)^2 + \frac{d\beta}{ds} \cdot \frac{d^2s}{d\alpha^2} \tag{7.3}$$

It can be assumed that the derivatives of s with respect to α are known from the design function $s(\alpha)$. The function $\beta(s)$ concerns a transfer function of the inverse mechanism. This function and the derivatives (with respect to s) can be calculated by investigating just the intermediate rocker-slider mechanism. For this occasion the program RUNMEC will be used, so a FEM model of this inverse mechanism must be defined. For a drawing of this mechanism model see fig. 7.3.3. Note that a ternary element can be used to define the slider motion $u(\alpha)$ as input motion of a form parameter, which will then be in the positive direction as meant in fig. 7.3.1.

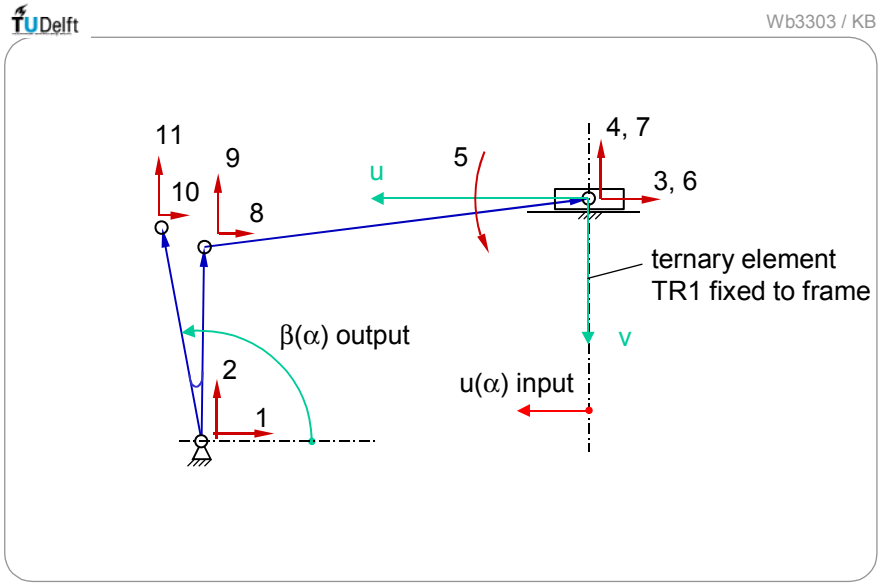

Fig. 7.3.3 FEM-model of inverse transformation mechanism

Table 7.1 shows the complete RUNMEC input file of the case. The input function in the file named doelf2.dat has been created using the function editor from the program TADSOC. The contents are partially listed in table 7.2 (columns α, s, $\frac{ds}{d\alpha}$, $\frac{d^2s}{d\alpha^2}$). The result, the cam design function, has been drawn in figures 7.3.4 and 7.3.5, while the numerical data can be found in table 7.3 (columns α, β, $\frac{d\beta}{d\alpha}$, $\frac{d^2\beta}{d\alpha^2}$). The file named camfunction2.dat can be imported directly in TADSOC to have the cam mechanism calculated. The cam design will not be worked out in this course.

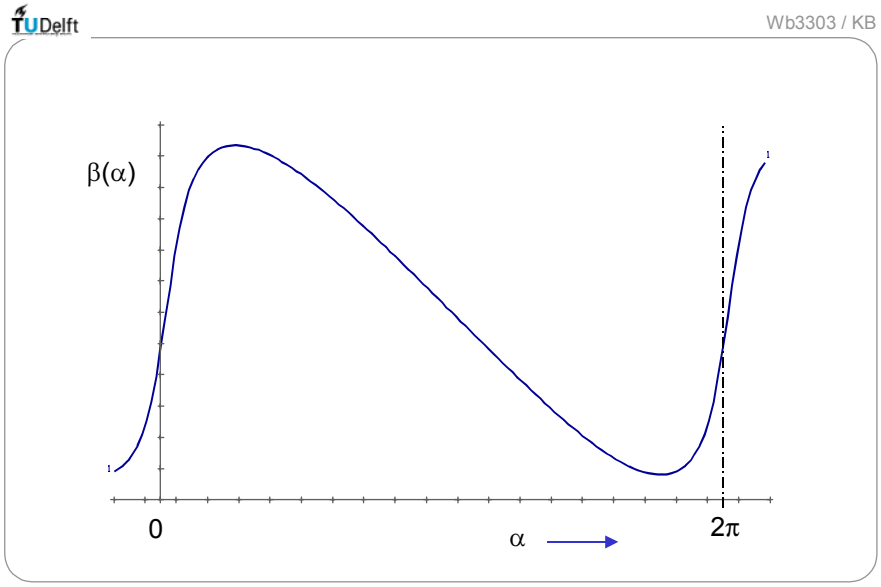

Fig. 7.3.4 Result of (inverse) transformation, order zero

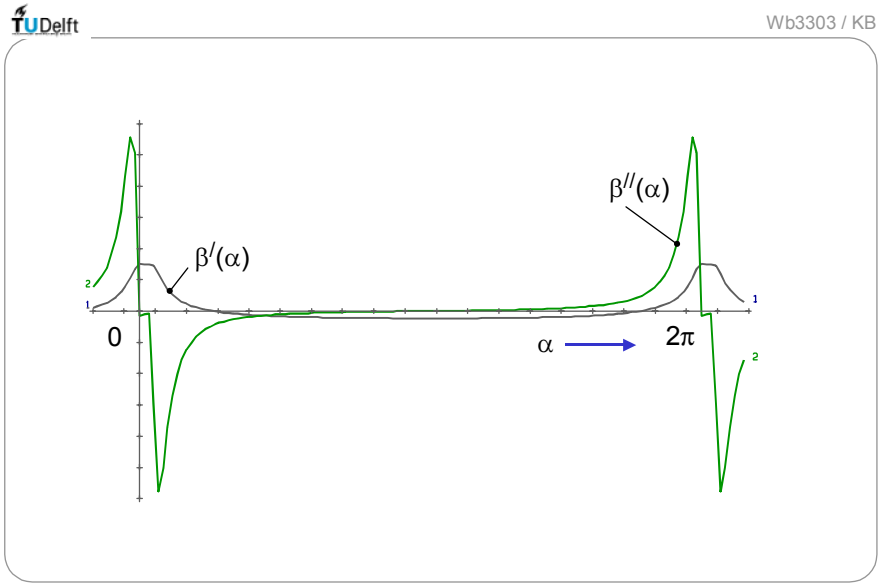

Fig. 7.3.5 Result of (inverse) transformation, order one and two

```
F=PL2 D=12 UA=RAD UY=M P=6.2831852 B=0.0 N=doelf2.dat
 120
   1   .0000000E+00   .0000000E+00   .1500007E+00   .0000000E+00
   2   .5235988E-01   .7854020E-02   .1500007E+00   .0000000E+00
   3   .1047198E+00   .1570804E-01   .1500007E+00   .0000000E+00
   4   .1570796E+00   .2352964E-01   .1464122E+00  -.2932325E+00
   5   .2094395E+00   .3059802E-01   .1210965E+00  -.5818399E+00
  ...
 116   .6021386E+01  -.2486619E-01   .4487730E-01   .2326652E+00
 117   .6073746E+01  -.2216330E-01   .5908119E-01   .3151193E+00
 118   .6126106E+01  -.1858881E-01   .7848784E-01   .4319968E+00
 119   .6178466E+01  -.1382514E-01   .1045915E+00   .5597535E+00
 120   .6230825E+01  -.7561895E-02   .1343680E+00   .5195267E+00
```

Table 7.2

```
F=NL2 D=12 N=CAMFUNCTION2.DAT
 120
   1   .0000000E+00   .1755094E+01   .7515837E+00  -.8030472E-01
   2   .5235988E-01   .1794347E+01   .7479893E+00  -.5737788E-01
   3   .1047198E+00   .1833443E+01   .7455386E+00  -.3654652E-01
   4   .1570796E+00   .1872277E+01   .7263334E+00  -.1471396E+01
   5   .2094395E+00   .1907329E+01   .6003846E+00  -.2885753E+01
  ...
 116   .6021386E+01   .1629014E+01   .2309843E+00   .1182079E+01
 117   .6073746E+01   .1642899E+01   .3029094E+00   .1590685E+01
 118   .6126106E+01   .1661181E+01   .4004975E+00   .2164426E+01
 119   .6178466E+01   .1685418E+01   .5306763E+00   .2778239E+01
 120   .6230825E+01   .1717092E+01   .6774248E+00   .2535152E+01
```

Table 7.3

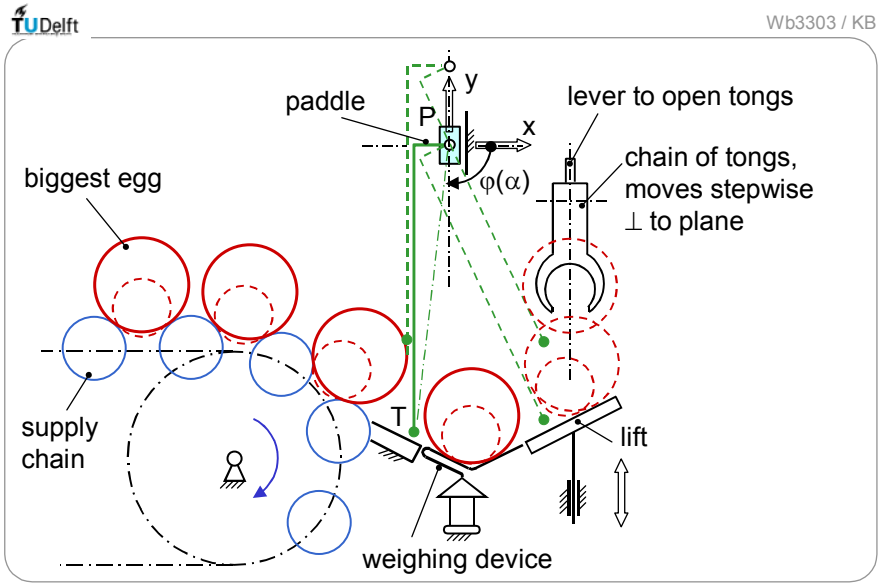

Fig.7.4.1 Egg weighing station

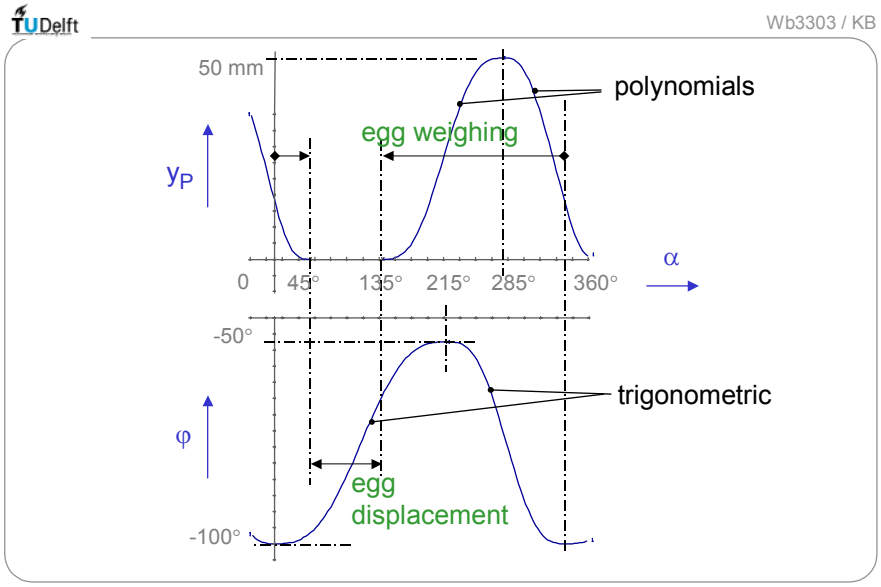

Fig.7.4.2 Design of paddle motion

7.4 Inverse kinematics (case: weighing station of eggs)

Figure 7.4.1 depicts a kinematic drawing of the weighing station. The eggs, yet all sizes together, lie in rows of six on a supply chain, which moves with constant speed. The so-called paddle has two functions:

- It guides the egg to the weighing device after falling over the edge of the supply chain.
- It pushes the previous egg off the weighing device to the lift and guides the egg while moving upwards until a pair of tongs takes over the egg.

The return path of the paddle is over the egg at the weighing device, giving it some time to be measured (the vibration time of the mass-spring system will be measured, at least one period of time is needed).
The pairs of tongs form a chain that moves stepwise perpendicular to the drawing plane. The row of six eggs will be treated parallel. The weighing information will be used to divide the eggs in 7 weight classes.
The case study concentrates on the motion and the mechanism to move the paddle. Actually it concerns a so-called plane motion (the motion of a point and the orientation of the plane, represented here by the two points P and T, need to be prescribed). Intuitively it was decided to simplify the motion: the point P can be guided along a vertical line, so that only two motion components need a prescription: y_p and angle of the paddle φ. These motion should be generated from a central driving shaft with angle α and will be indicated further as

- Lift function $y(\alpha)$ and
- Tilt function $\varphi(\alpha)$.

For this case the precise form of the paddle could be subject of discussion. It will be assumed here that the paddle is a line (a plane perpendicular to the drawing) with negligible thickness.
The design procedure is based on the assumption that some characteristic positions of the paddle are given, that means some discrete points of the lift function and the tilt function are given. These functions will be completed to be a continuous function, using standard functions like polynomials and trigonometric functions on the intervals (available in the function editor of the cam design program TADSOC), in order to construct a continuous path for point T, the tip of the paddle. Mainly this path will be used to judge the motion of the paddle. After some trial-and-error the motion components were defined as depicted in fig. 7.4.2. These functions are then input of a mechanism, which consists just of the paddle. The path and velocity along the path of point T will be calculated then with a simple RUNMEC model, which will not be explained here further. For the result see fig. 7.4.3. It can be verified that this paddle motion treats all sizes of eggs carefully, because velocity jumps (impacts) have been kept low.
To apply the desired tilt function to the paddle, the actuator could be mounted on the slider of point P. This is however not a favourable solution, since the mass of the tilt drive causes a great fluctuation in potential energy. It is worthwhile to consider the application of an intermediate mechanism, which has degree of freedom two, and which can be driven relative to the frame for both inputs. Such mechanism can be found in figure 2.6.2 (five bar chain). In particular the configurations with one slider are of interest here. The configuration as drawn in fig. 7.4.4, left hand part, is typically suited. The two inputs could for instance be:

- The slider motion $s(\alpha)$, which can be taken identical with the lift motion, and
- The angular motion of rocker B0B, indicated with $\beta(\alpha)$.

These new input functions can for instance be generated by cam mechanisms, as suggested in fig. 7.4.4, right hand part. The intermediate mechanism $PQBB_0$ transforms thus the motion as generated by the cam (here for the tilt motion). To design the cam –for the tilt- the reverse of the intermediate mechanism needs to be considered:

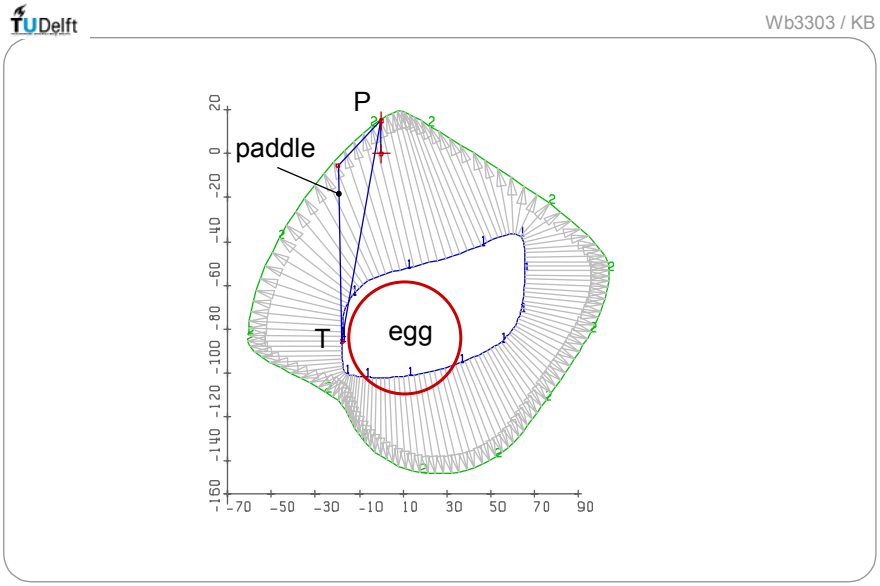

Fig. 7.4.3 Design path of paddle with velocity hodograph

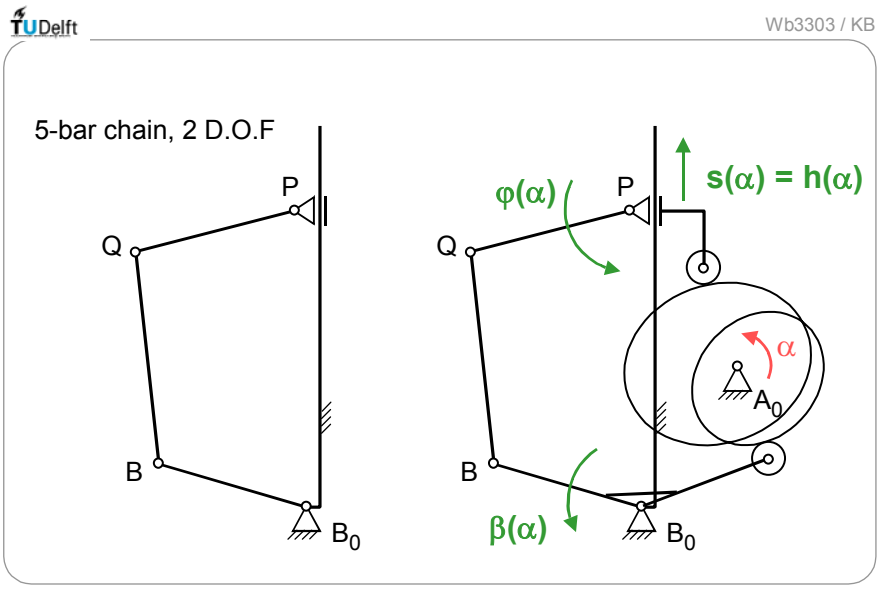

Fig. 7.4.4 Intermediate mechanism, driven with central shaft

$$\beta = \beta(y, \varphi) \quad \text{with} \quad y(\alpha) \quad \text{and} \quad \varphi(\alpha) \tag{7.4}$$

$$\frac{d\beta}{d\alpha} = \frac{\partial\beta}{\partial y} \cdot \frac{dy}{d\alpha} + \frac{\partial\beta}{\partial\varphi} \cdot \frac{d\varphi}{d\alpha} \tag{7.5}$$

$$\frac{d^2\beta}{d\alpha^2} = \frac{\partial^2\beta}{\partial y^2} \cdot \left(\frac{dy}{d\alpha}\right)^2 + \frac{\partial^2\beta}{\partial\varphi^2} \cdot \left(\frac{d\varphi}{d\alpha}\right)^2 + \frac{\partial\beta}{\partial y} \cdot \frac{d^2\beta}{d\alpha^2} + \frac{\partial\beta}{\partial\varphi} \cdot \frac{d^2\varphi}{d\alpha^2} + \frac{2\partial^2\beta}{\partial\varphi\partial y} \cdot \frac{d\varphi}{d\alpha} \cdot \frac{dy}{d\alpha} \tag{7.6}$$

The functions $y(\alpha)$ and $\varphi(\alpha)$, including their derivatives with respect to α, are assumed to be known by the specification of the paddle function (fig. 7.4.2). The other derivatives can be obtained from the inverse mechanism. Instead of doing the multiplication of equations (7.5) and (7.6) by hand or by a specific programming code, the result can also be obtained in the generalized implicit differentiation algorithm (5.31a) and (5.32a). The output function like $\beta(\alpha)$ including its (timed) derivatives can be stored in a PION-file as a design function for the cam. The cam design itself will be left out the case here.
To demonstrate the method a model of the inverse mechanism is depicted in fig. 7.4.5. The Runmec input file is listed in table 7.4. The transformed tilt function $\beta(\alpha)$ is drawn in fig. 7.4.6.

7.5 References

[1] VDI-Richtlinie 2126: Konstruktion übertragungsgünstigster Schubkurbel-Umwandlung von Schwing- in Schubbewegungen.

[2] Veen, T.D van der: Inhangmechanisme eiersoorteermechanisme (in Dutch). Stageverslag sectie Bedrijfsmechanisatie, TU Delft, 1981.

[3] Bos, J.M.: Ontwerp nokken bewegingsmechanisme van de opvangplaat van een eiersoorteermachine (in Dutch). Stageverslag sectie Bedrijfsmechanisatie, TU Delft, 1983.

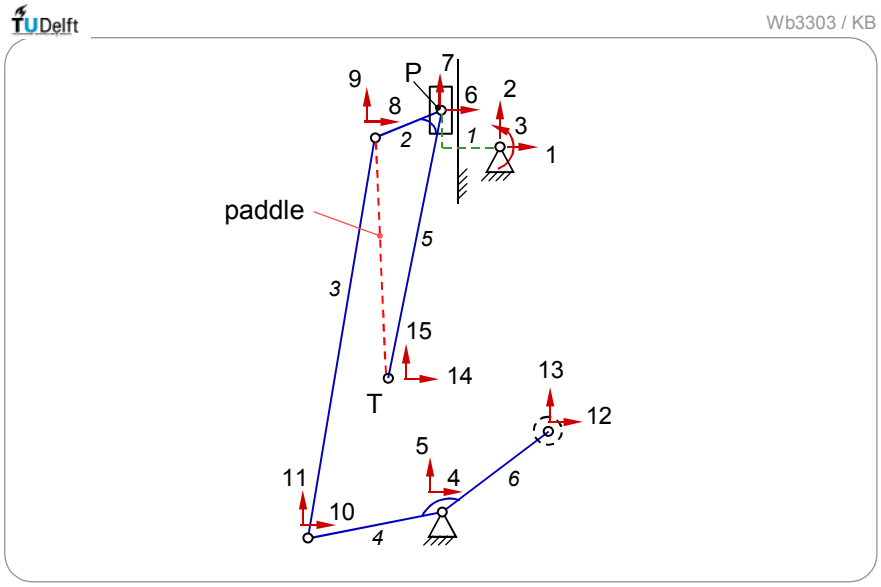

Fig. 7.4.5 FEM-model of inverse paddle mechanism

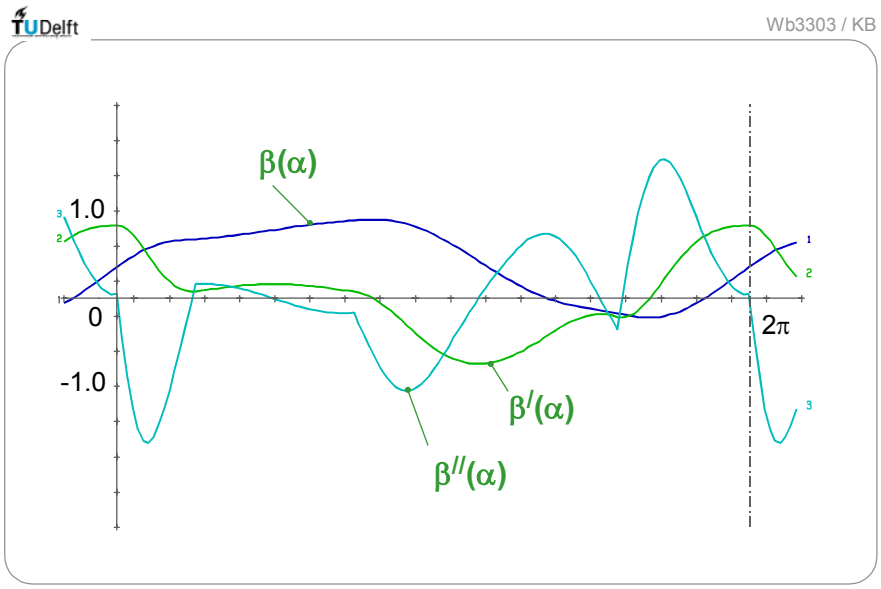

Fig. 7.4.6 Tilt function (design function for the cam)

```
; case egg weighing machine, two input functions at paddle, cam follower added
DYN ANA                                        (for Runmec version 3.x)
TOPOLOGY
 ELEM 1 TR1  1  2  3  6  7   ; Ternary element, vertical slider
      2 BIN  6  7  8  9      ; bar connected to spoon
      3 BIN  8  9 10 11      ; the coupler
      4 BIN  4  5 10 11      ; the rocker
      5 BIN  4  5 12 13      ; cam follower
      6 BIN  6  7 14 15      ; the paddle (connection line)
 FORM 1 lenl 2 1  1.0
      2 lenl 3 2  1.0
      3 lenl 4 3  1.0
      4 lenl 5 4  1.0
      5 lenl 6 5  1.0
      6 angb 2 6  1.0
      7 angb 6 6 -1.0 ; spoon has fixed angle with bar 2
      8 angb 4 7  1.0
      9 angb 5 7 -1.0 ; cam follower B0A has fixed angle with rocker QB
     10 lenv 1 8  1.0
     11 angb 6 9  1.0
     12 angb 5 -1 1.0 ; follower angle, for output
NRDOF 2

GEOMETRY
 XFIX 1  0.0
      2  0.0
      3  0.0
      4  0.0
      5 -150.0
      6  0.0
 XMOV 7  0.0
      8 -25.0
      9 -10.0
     10 -50.0
     11 -160.0
     12  40.0
     13 -120.0
     14 -20.0
     15 -100.0

DYNAMICS
STORAGE
 BUFTIME 99
 BUFX0    14    600
 BUFDX1   14    601
 BUFDX2   14    602
 BUFX0    15    700
 BUFDX1   15    701
 BUFDX2   15    702
 BUFE0    -1    800
 BUFDE1   -1    801
 BUFDE2   -1    802
 PRINTBUF
 PIONOUTE -1 9 2 follower.dat

 BEGINPIC
  drawmech 1
  plothodo 14  15 ; remark: use DYN mode to obtain velocity-hodograph
 ENDPIC

 BEGINPIC
  plotfunc 99 600
  plotfunc 99 700
 ENDPIC
 BEGINPIC
  plotfunc 99 800
  plotfunc 99 801
  plotfunc 99 802
 ENDPIC

MOVEMENT
 MOV_FUNC heffen.dat draaien.dat
END
```

Table 7.4