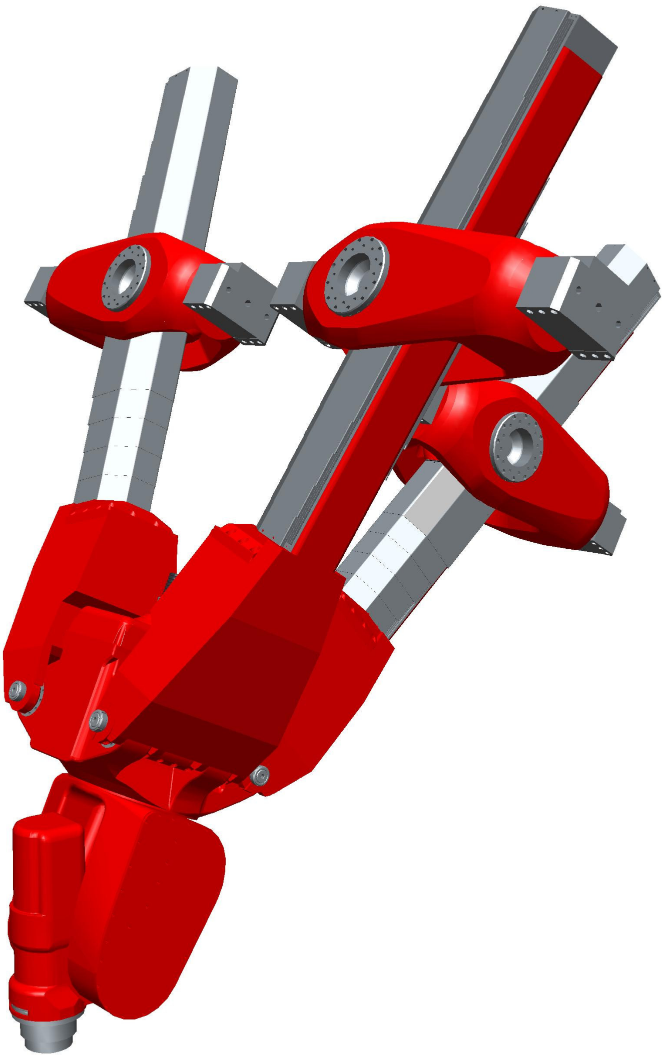
Description of functions Execon kinematics DLL

2007/03/12 Edition



Version: 1.4
Creation date: 2006-06-27
Latest modification: 2007-03-12

1 Brief Description

The Exechon kinematics Dynamic Link Library, DLL is a Windows C/C++ DLL that contains two functions for calculating the direct and inverse kinematics of the Exechon parallel kinematics machine. On initialization the DLL file reads Siemens 840D parameters that define the basic geometric dimensions of the machine. This file is typically used as a component when creating a simulation model of the Exechon PKM machine.

2 Supported Siemens Parameters

The following Siemens parameters are supported in the DLL and read on initialization:
A text file with these parameters should be stored in the same directory as the DLL file.

Parameter	Notation
• THYK_BASE_DIST_1_3:	λ
• THYK_BASE_JOINT_2:	A_2
• THYK_PLAT_JOINT_1:	B_1
• THYK_PLAT_JOINT_2:	B_2
• THYK_PLAT_JOINT_3:	B_3
• THYK_HAND_OFFSET:	c
• THYK_BASE_TOOL:	t_2 and t_3
• THYK_BCS_ICS_TRANS:	Zero offset of the ICS with respect to the BCS
• THYK_BCS_ICS_ROT:	Rotation of the ICS with respect to the BCS

Note: There are some additional parameters that are used to correct small geometric errors for the real machine, these parameters are not supported in the DLL.

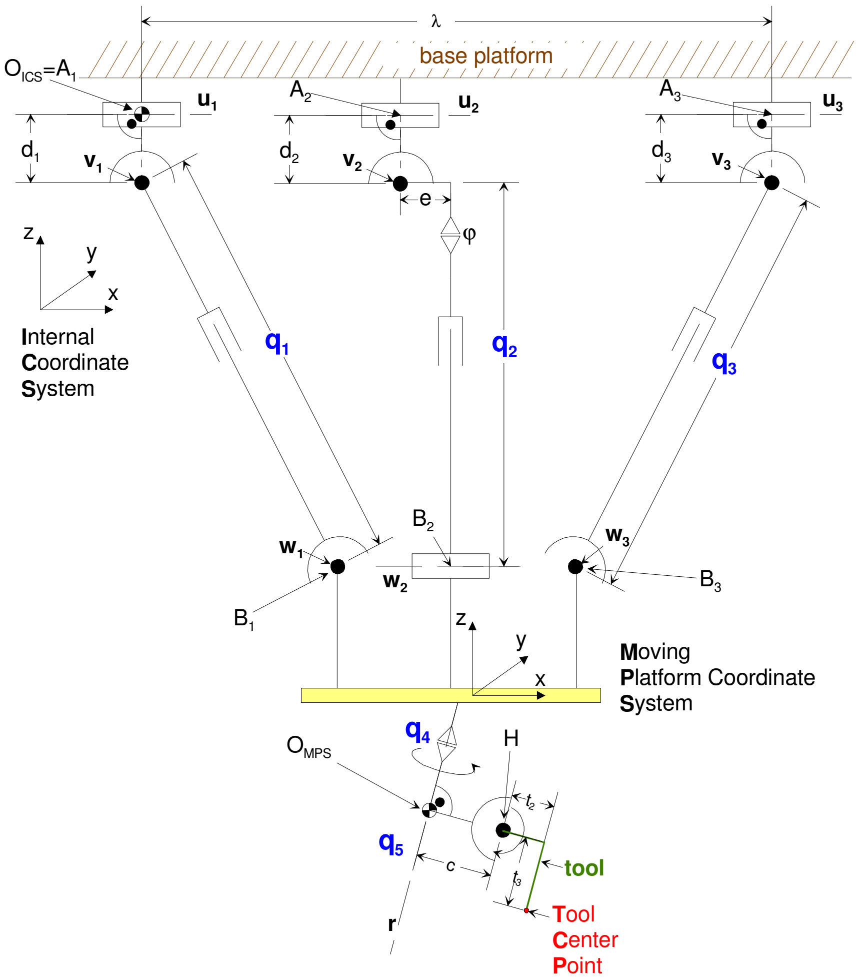


Figure 1: Equivalent kinematic diagram of the kinematics module in XZ plane

2.1 Notations

In the following the index i runs from 1 to 3.

BCS	basic coordinate system
ICS	internal coordinate system
MPS	moving platform coordinate system
O_{ICS}	origin of the base platform
O_{MPS}	origin of the moving platform, i.e. intersection of axis 4 with the perpendicular of axis 5 to axis 4
u_i	rotary axis of the universal joint fixed at the base platform ("outer joint")
v_i	rotary axis of the universal joint fixed at the linear axis ("inner joint")
A_i	intersection of the perpendicular between the axes of a universal joint at the outer joint
λ	distance between A_1 and A_3 along the X direction of the ICS
φ	Angle between the second linear axis about its longitudinal direction, i.e. the angle between $\mathbf{v_2}$ and $\mathbf{w_2}$
q_i	length of the perpendicular between $\mathbf{v_i}$ and $\mathbf{w_i}$. This matches the value of the corresponding linear axis.
q_4	Angle of the fourth rotary axis
q_5	Angle of the fifth rotary axis
w_i	rotary axis at the moving platform
B_i	intersection of the perpendicular between $\mathbf{v_i}$ and $\mathbf{w_i}$ at the rotary near the moving platform. This point is unique by the requirement that the perpendicular point from the outer to the inner joint of the universal joint is on that perpendicular.
H	hand point, i.e. intersection between axis five and the perpendicular from O_{MPS}
c	distance between O_{MPS} and H
tool	tool direction
t_1	tool dimension along axis 5
t_2	tool dimension perpendicular to the tool direction and axis 5
t_3	tool dimension along the tool direction

3 Kinematics

3.1 Definitions

- Primary axes refer to the driven axes 1 to 5, noted as q_1-q_5 in Figure 1.
- Secondary axes refer to the "passive" axes, such as the gimbals and the moving platform.
- Rotation angles are given in **degrees**

The 4x4 matrix represent the location and orientation of the Tool Center Point and is defined like this:

$$\text{Matrix:=}\begin{bmatrix} Nx & Ny & Nz & 0 \\ Ox & Oy & Oz & 0 \\ Ax & Ay & Az & 0 \\ Px & Py & Pz & 1 \end{bmatrix}$$

- **N** defines "Normal" vector that is unit vector in X-direction.
- **O** defines "Orientation" vector that is unit vector in Y-direction.
- **A** defines "Approach" vector that is unit vector in Z-direction.
- **P** defines the position.

3.2 Direct Kinematics

Input:

- One dimensional array with axes 1 to 5, axes 1 to 3 are linear and 4 to 5 rotational
- Tool offset in Y and Z direction

Output:

- 4x4 position matrix of the Tool Center Point, TCP
- One dimensional array with secondary axes (7 values + moving platform position, 2 rotations and XYZ position, total 12 values)

Function prototype:

```
Int DirektKin(
double *Ax1To5,
double *ToolLength,
double *ToolMatrix,
double *SecondaryJointValues);
```

Function return 0 if calculation is successful, 1 if not successful

Parameters

Ax1To5 Pointer to a 5x1 array with the 5 axes values starting from index 0

[0] Axis 1 linear
[1] Axis 2 linear
[2] Axis 3 linear
[3] Axis 4 rotational, value in degrees
[4] Axis 5 rotational, value in degrees

ToolLength Pointer to a 2x1 array with Tool offset value
ToolLength[0]: Offset in Y
ToolLength[1]: Offset in Z

ToolMatrix Pointer to a 4x4 matrix with Tool Center Point position and orientation

SecondaryJointValues Pointer to a 12x1 array with secondary axes starting with index 0

[0] Angle of outer gimbal 1 rotation
[1] Angle of inner gimbal 1 rotation
[2] Angle of outer gimbal 2 rotation
[3] Angle of inner gimbal 2 rotation
[4] Angle of actuator 2 torsion (φ)
[5] Angle of outer gimbal 3 rotation
[6] Angle of inner gimbal 3 rotation
[7] Position of moving platform (O_{MPS}) in X relative ICS
[8] Position of moving platform (O_{MPS}) in Y relative ICS
[9] Position of moving platform (O_{MPS}) in Z relative ICS
[10] First rotation of moving platform around X in O_{MPS}
[11] Second rotation of moving platform rotation around the rotated Y in O_{MPS}

3.3 Inverse Kinematics

Input:
- Array 6x1 xyzijk with tool position and orientation
- Tool offset in Y and Z direction

Output:
- Array 5x2 with primary axes values and for two solutions
- Array 12x2 with secondary axes values and for two solutions

Function prototype:

```
Int InverseKinematics(
double *xyzijk,
double *ToolLength,
double *Ax1To5,
double *SecondaryAxes);
```

Function return 0 if calculation is successful, 1 if not successful and 2 if singular solution is found.

xyzijk Pointer to a 6x1 array with Tool Center Point position and orientation

[0] Tool position in X
[1] Tool position in Y
[2] Tool position in Z
[3] Tool vector component i
[4] Tool vector component j
[5] Tool vector component k

ToolLength Pointer to a 2x1 array with Tool offset value

ToolLength[0]: Offset in Y
ToolLength[1]: Offset in Z

PrimaryAxes Pointer to a 5x2 array with the 5 axes values and for two solutions

[0][0/1] Axis 1 linear
[1][0/1] Axis 2 linear
[2][0/1] Axis 3 linear
[3][0/1] Axis 4 rotational, value in degrees
[4][0/1] Axis 5 rotational, value in degrees

SecondaryAxes Pointer to a 12x2 array with secondary axes and for two solutions

[0][0/1] Angle of outer gimbal 1 rotation
[1][0/1] Angle of inner gimbal 1 rotation
[2][0/1] Angle of outer gimbal 2 rotation
[3][0/1] Angle of inner gimbal 2 rotation
[4][0/1] Angle of actuator 2 torsion (φ)
[5][0/1] Angle of outer gimbal 3 rotation
[6][0/1] Angle of inner gimbal 3 rotation
[7][0/1] Position of moving platform (O_{MPS}) in X relative ICS
[8][0/1] Position of moving platform (O_{MPS}) in Y relative ICS
[9][0/1] Position of moving platform (O_{MPS}) in Z relative ICS
[10][0/1] First rotation of moving platform around X
[11][0/1] Second rotation of moving platform rotation around the rotated Y