

INTERNATIONAL STANDARD

ISO
10303-105

First edition
1996-12-15

Industrial automation systems and integration — Product data representation and exchange —

Part 105:

Integrated application resource: Kinematics

*Systèmes d'automatisation industrielle et intégration — Représentation et
échange de données de produits —*

Partie 105: Ressource d'application intégrée: Cinématique



Reference number
ISO 10303-105:1996(E)
AVA Documents
Document Sharing Hub

Contents	Page
1 Scope	1
2 Normative references	1
3 Definitions	2
3.1 Terms defined in ISO 10303-1	2
3.2 Terms defined in ISO 10303-42	2
3.3 Terms defined in ISO 8855	3
3.4 Other definitions	3
3.4.1 base	3
3.4.2 frame	3
3.4.3 ground	3
3.4.4 joint	3
3.4.5 kinematics	3
3.4.6 kinematic chain	3
3.4.7 link	3
3.4.8 link frame	3
3.4.9 loop	3
3.4.10 mechanism	3
3.4.11 motion	3
3.4.12 pair	3
3.4.13 pair actuation	4
3.4.14 placement	4
3.4.15 transform	4
3.4.16 SU-parameters	4
3.4.17 world coordinate system	4
4 Symbols and abbreviations	4
4.1 Mathematical symbols	4
4.2 Graphic convention for matrix representation	4
4.3 Abbreviations	4
5 Kinematic structure	6

5.1	Introduction	7
5.2	Fundamental concepts and assumptions	8
5.2.1	Kinematic model structure	8
5.2.2	Positioning and placement	9
5.2.3	Positive sense of rotations	10
5.2.4	Shape representations	10
5.2.5	Consistent use of units	11
5.3	kinematic_structure_schema type definitions	11
5.3.1	rigid_placement	11
5.3.2	rotational_range_measure	11
5.3.3	translational_range_measure	12
5.3.4	unlimited_range	12
5.3.5	spatial_rotation	12
5.3.6	ypr_enumeration	12
5.3.7	ypr_rotation	13
5.3.8	kinematic_frame_background	14
5.4	kinematic_structure_schema entity definitions	14
5.4.1	rotation_about_direction	14
5.4.2	kinematic_property_definition	14
5.4.3	kinematic_property_representation_relation	15
5.4.4	kinematic_ground_representation	15
5.4.5	mechanism	16
5.4.6	mechanism_base_placement	17
5.4.7	initial_state	18
5.4.8	kinematic_structure	19
5.4.9	kinematic_joint	19
5.4.10	kinematic_link	20
5.4.11	kinematic_link_representation_relation	21
5.4.12	kinematic_link_representation	21
5.4.13	kinematic_link_representation_association	22
5.4.14	kinematic_frame_background_representation	23
5.4.15	kinematic_frame_based_transformation	24
5.4.16	kinematic_frame_background_representation_association	24
5.4.17	su_parameters	25
5.4.18	kinematic_pair	27
5.4.19	pair_actuator	29
5.4.20	pair_value	31
5.4.21	simple_pair_range	31
5.4.22	revolute_pair	31
5.4.23	revolute_pair_value	32
5.4.24	revolute_pair_range	32
5.4.25	prismatic_pair	33
5.4.26	prismatic_pair_value	33
5.4.27	prismatic_pair_range	34
5.4.28	screw_pair	35
5.4.29	screw_pair_value	36

5.4.30	screw_pair_range	37
5.4.31	cylindrical_pair	37
5.4.32	cylindrical_pair_value	38
5.4.33	cylindrical_pair_range	39
5.4.34	spherical_pair	40
5.4.35	spherical_pair_value	40
5.4.36	spherical_pair_range	41
5.4.37	universal_pair	43
5.4.38	universal_pair_value	43
5.4.39	universal_pair_range	44
5.4.40	planar_pair	46
5.4.41	planar_pair_value	46
5.4.42	planar_pair_range	46
5.4.43	unconstrained_pair	48
5.4.44	unconstrained_pair_value	49
5.4.45	fully_constrained_pair	49
5.4.46	point_on_surface_pair	50
5.4.47	point_on_surface_pair_value	52
5.4.48	point_on_surface_pair_range	53
5.4.49	surface_pair	55
5.4.50	surface_pair_range	59
5.4.51	sliding_surface_pair	60
5.4.52	sliding_surface_pair_value	60
5.4.53	rolling_surface_pair	61
5.4.54	rolling_surface_pair_value	61
5.4.55	point_on_planar_curve_pair	62
5.4.56	point_on_planar_curve_pair_value	65
5.4.57	point_on_planar_curve_pair_range	66
5.4.58	planar_curve_pair	68
5.4.59	planar_curve_pair_range	72
5.4.60	sliding_curve_pair	72
5.4.61	sliding_curve_pair_value	73
5.4.62	rolling_curve_pair	73
5.4.63	rolling_curve_pair_value	74
5.4.64	gear_pair	74
5.4.65	gear_pair_value	77
5.4.66	gear_pair_range	78
5.4.67	rack_and_pinion_pair	78
5.4.68	rack_and_pinion_pair_value	79
5.4.69	rack_and_pinion_pair_range	80
5.4.70	kinematic_substructure	81
5.4.71	kinematic_network_structure	82
5.4.72	kinematic_tree_structure	82
5.4.73	kinematic_loop	82
5.4.74	joint_logical_relationship	84
5.4.75	oriented_joint	85

5.4.76	oriented_joint_in_tree	86
5.4.77	advent_oriented_joint	86
5.5	kinematic_structure_schema function definitions	86
5.5.1	ypr_index	86
5.5.2	representation_of_link	87
5.5.3	suitably_based_mechanism	87
5.5.4	unique_link_usage	89
5.5.5	coordinated_pair_link_representation	90
5.5.6	frame_associated_to_background	90
5.5.7	plane_angle_for_pair_in_radian	92
5.5.8	convert_plane_angle_for_pair_from_radian	94
5.5.9	convert_spatial_to_ypr_rotation	96
5.5.10	assign_directed_link	100
5.5.11	connected_in_simple_loop	101
6	Kinematic motion representation	102
6.1	Introduction	102
6.2	Fundamental concepts and assumptions	102
6.3	kinematic_motion_representation_schema type definition: motion_parameter- measure	103
6.4	kinematic_motion_representation_schema entity definitions	103
6.4.1	translation	103
6.4.2	transform	103
6.4.3	path_node	104
6.4.4	kinematic_path	104
6.4.5	path_element_connection	104
6.4.6	composite_path	105
6.4.7	path_element	106
6.4.8	point_to_point_path	107
6.4.9	circular_path	107
6.4.10	linear_path	107
6.4.11	curve_based_path	108
6.5	kinematic_motion_representation_schema function definitions	108
6.5.1	connected_in_simple_path	108
6.5.2	compare_unit_components	110
6.5.3	increasing_measure_value	110
6.5.4	non_coincident_coordinates	111
7	Kinematic analysis control and result	112
7.1	Introduction	112
7.2	Fundamental concepts and assumptions	113
7.3	kinematic_analysis_control_and_result_schema type definitions	113
7.3.1	interpolation_type	113
7.3.2	kinematic_analysis_definition	114
7.3.3	kinematic_result	114
7.4	kinematic_analysis_control_and_result_schema entity definitions	114

7.4.1	configuration_interpolation	114
7.4.2	kinematic_control	115
7.4.3	founded_kinematic_path	115
7.4.4	motion_link_relationship	116
7.4.5	configuration_definition	117
7.4.6	interpolated_configuration_sequence	117
7.4.7	prescribed_path	117
7.4.8	kinematic_analysis_result	118
7.4.9	kinematic_analysis_consistency	119
7.4.10	resulting_path	119

Annexes

A	Short names of entities	120
B	Information object registration	123
B.1	Document identification	123
B.2	Schema identification	123
B.2.1	kinematic_structure_schema identification	123
B.2.2	kinematic_motion_representation_schema identification	123
B.2.3	kinematic_analysis_control_and_result_schema identification	123
C	<i>EXPRESS</i> listing	124
D	<i>EXPRESS-G</i> figures	125
E	Rotation matrix corresponding to ypr_rotation	135
E.1	Matrix representation of ypr_rotation	135
E.2	<i>EXPRESS</i> function compute_rotation_matrix_from_ypr_rotation	136
F	Replacement of DH-Parameters by SU-Parameters	138
G	Bibliography	140
	Index	141

Figures

1	The relationships between the three schemas of ISO 10303-105: Kinematics	xv
2	Definition of the su-parameters	26
3	Placement relationships of the pair frames relative to the mating links	30
4	Example of a joint representing a revolute pair	32
5	Example of a joint representing a prismatic pair	34
6	Example of a joint representing a screw pair	36
7	Example of a joint representing a cylindrical pair	38
8	Example of a joint representing a spherical pair	41
9	Example of a joint representing a universal pair	44
10	Example of a joint representing a planar pair	47

11	Example of a point on surface pair	51
12	Example of a surface pair.	56
13	Example of a point on planar curve pair	63
14	Example of a planar curve pair.	69
15	Examples of gear pairs in different situations.	75
16	Example of a rack and pinion pair in the reference situation	79
17	Example of a kinematic network structure	83
18	Example of a kinematic tree structure	83
D.1	Graphical notation of the major aspects of the kinematic_structure_schema – diagram	
1 of 7		126
D.2	Graphical notation of the major aspects of the kinematic_structure_schema – diagram	
2 of 7		127
D.3	Graphical notation of the major aspects of the kinematic_structure_schema – diagram	
3 of 7		128
D.4	Graphical notation of the major aspects of the kinematic_structure_schema – diagram	
4 of 7		129
D.5	Graphical notation of the major aspects of the kinematic_structure_schema – diagram	
5 of 7		130
D.6	Graphical notation of the major aspects of the kinematic_structure_schema – diagram	
6 of 7		131
D.7	Graphical notation of the major aspects of the kinematic_structure_schema – diagram	
7 of 7		132
D.8	Graphical notation of the major aspects of the kinematic_motion_representation_- schema	133
D.9	Graphical notation of the major aspects of the kinematic_analysis_control_and_result_- schema	134

Tables

1	Mathematical symbology	5
A.1	Short names of entities	120

Foreword

The International Organization for Standardization (ISO) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

Draft International Standards adopted by technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75% of the member bodies casting a vote.

International Standard ISO 10303-105 was prepared by Technical Committee ISO/TC 184, *Industrial automation systems and integration*, Subcommittee SC4, *Industrial data*.

ISO 10303 consists of the following parts under the general title *Industrial automation systems and integration – Product data representation and exchange*:

- Part 1, Overview and fundamental principles;
- Part 11, Description methods: The *EXPRESS* language reference manual;
- Part 12, Description method: The *EXPRESS-I* language reference manual;
- Part 21, Implementation methods: Clear text encoding of the exchange structure;
- Part 22, Implementation method: Standard data access interface specification;
- Part 23, Implementation method: C++ language binding to the standard data access interface;
- Part 24, Implementation method: C language binding to the standard data access interface;
- Part 26, Implementation method: Interface definition language binding to the standard data access interface;
- Part 31, Conformance testing methodology and framework: General concepts;
- Part 32, Conformance testing methodology and framework: Requirements on testing laboratories and clients;
- Part 33, Conformance testing methodology and framework: Structure and use of abstract test suites;
- Part 34, Conformance testing methodology and framework: Abstract test methods;

- Part 35, Conformance testing methodology and framework: Abstract test methods for SDAI implementations;
- Part 41, Integrated generic resources: Fundamentals of product description and support;
- Part 42, Integrated generic resources: Geometric and topological representation;
- Part 43, Integrated generic resources: Representation structures;
- Part 44, Integrated generic resources: Product structure configuration;
- Part 45, Integrated generic resource: Materials;
- Part 46, Integrated generic resource: Visual presentation;
- Part 47, Integrated generic resource: Shape variation tolerances;
- Part 49, Integrated generic resource: Process structure and properties;
- Part 101, Integrated application resource: Draughting;
- Part 104, Integrated application resource: Finite element analysis;
- Part 105, Integrated application resource: Kinematics;
- Part 106, Integrated application resource: Building construction core model;
- Part 201, Application protocol: Explicit draughting;
- Part 202, Application protocol: Associative draughting;
- Part 203, Application protocol: Configuration controlled design;
- Part 204, Application protocol: Mechanical design using boundary representation;
- Part 205, Application protocol: Mechanical design using surface representation;
- Part 207, Application protocol: Sheet metal die planning and design;
- Part 208, Application protocol: Life cycle management - Change process;
- Part 209, Application protocol: Composite and metallic structural analysis and related design;
- Part 210, Application protocol: Design of layered electronic products;

- Part 211, Application protocol: Electronics test diagnostics and remanufacture;
- Part 212, Application protocol: Electrotechnical design and installation;
- Part 213, Application protocol: Numerical control process plans for machined parts;
- Part 214, Application protocol: Core data for automotive mechanical design;
- Part 215, Application protocol: Ship arrangement;
- Part 216, Application protocol: Ship moulded forms;
- Part 217, Application protocol: Ship piping;
- Part 218, Application protocol: Ship structures;
- Part 220, Application protocol: Process planning, manufacture, and assembly of layered electronic products;
- Part 221, Application protocol: Functional data and their schematic representation for process plant;
- Part 222, Application protocol: Exchange of product data for composite structures;
- Part 223, Application protocol: Exchange of design and manufacturing product information for cast parts;
- Part 224, Application protocol: Mechanical product definition for process plans using mechanical feature;
- Part 225, Application protocol: Building elements using explicit shape representation;
- Part 226, Application protocol: Ship mechanical systems;
- Part 227, Application protocol: Plant spatial configuration;
- Part 228, Application protocol: Building services: Heating, ventilation, and air conditioning;
- Part 229, Application protocol: Exchange of design and manufacturing product information for forged parts;
- Part 230, Application protocol: Building structural frame: Steelwork;
- Part 231, Application protocol: Process engineering data: Process design and process specification of major equipment;

- Part 232, Application protocol: Technical data package;
- Part 301, Abstract test suite: Explicit draughting;
- Part 302, Abstract test suite: Associative draughting;
- Part 303, Abstract test suite: Configuration controlled design;
- Part 304, Abstract test suite: Mechanical design using boundary representation;
- Part 305, Abstract test suite: Mechanical design using surface representation;
- Part 307, Abstract test suite: Sheet metal die planning and design;
- Part 308, Abstract test suite: Life cycle management - Change process;
- Part 309, Abstract test suite: Composite and metallic structural analysis and related design;
- Part 310, Abstract test suite: Design of layered electronic products;
- Part 311, Abstract test suite: Electronics test diagnostics and remanufacture;
- Part 312, Abstract test suite: Electrotechnical design and installation;
- Part 313, Abstract test suite: Numerical control process plans for machined parts;
- Part 314, Abstract test suite: Core data for automotive mechanical design;
- Part 315, Abstract test suite: Ship arrangement;
- Part 316, Abstract test suite: Ship moulded forms;
- Part 317, Abstract test suite: Ship piping;
- Part 318, Abstract test suite: Ship structures;
- Part 320, Abstract test suite: Process planning, manufacture, and assembly of layered electronic products;
- Part 321, Abstract test suite: Functional data and their schematic representation for process plant;
- Part 322, Abstract test suite: Exchange of product data for composite structures;
- Part 323, Abstract test suite: Exchange of design and manufacturing product information for cast parts;

- Part 324, Abstract test suite: Mechanical product definition for process plans using mechanical features;
- Part 325, Abstract test suite: Building elements using explicit shape representation;
- Part 326, Abstract test suite: Ship mechanical systems;
- Part 327, Abstract test suite: Plant spatial configuration;
- Part 328, Abstract test suite: Building services: Heating, ventilation, and air conditioning;
- Part 329, Abstract test suite: Exchange of design and manufacturing product information for forged parts;
- Part 330, Abstract test suite: Building structural frame: Steelwork;
- Part 331, Abstract test suite: Process engineering data: Process design and process specification of major equipment;
- Part 332, Abstract test suite: Technical data package;
- Part 501, Application interpreted construct: Edge-based wireframe;
- Part 502, Application interpreted construct: Shell-based wireframe;
- Part 503, Application interpreted construct: Geometrically bounded 2D wireframe;
- Part 504, Application interpreted construct: Draughting annotation;
- Part 505, Application interpreted construct: Drawing structure and administration;
- Part 506, Application interpreted construct: Draughting elements;
- Part 507, Application interpreted construct: Geometrically bounded surface;
- Part 508, Application interpreted construct: Non-manifold surface;
- Part 509, Application interpreted construct: Manifold surface;
- Part 510, Application interpreted construct: Geometrically bounded wireframe;
- Part 511, Application interpreted construct: Topologically bounded surface;
- Part 512, Application interpreted construct: Faceted boundary representation;
- Part 513, Application interpreted construct: Elementary boundary representation;

- Part 514, Application interpreted construct: Advanced boundary representation;
- Part 515, Application interpreted construct: Constructive solid geometry;
- Part 517, Application interpreted construct: Mechanical design geometric presentation;
- Part 518, Application interpreted construct: Mechanical design shaded representation.

The structure of this International Standard is described in ISO 10303-1. The numbering of the parts of this International Standard reflects its structure:

- Parts 11 to 13 specify the description methods,
- Parts 21 to 26 specify the implementation methods,
- Parts 31 to 35 specify the conformance testing methodology and framework,
- Parts 41 to 49 specify the integrated generic resources,
- Parts 101 to 106 specify the integrated application resources,
- Parts 201 to 232 specify the application protocols,
- Parts 301 to 332 specify the abstract test suites, and
- Parts 501 to 518 specify the application interpreted constructs.

Should further parts be published, they will follow the same numbering pattern.

Annexes A and B are an integral part of this part of ISO 10303. Annexes C, D, E, F, and G are for information only.

Introduction

ISO 10303 is an International Standard for the computer-interpretable representation and exchange of product data. The objective is to provide a neutral mechanism capable of describing product data throughout the life cycle of a product independent from any particular system. The nature of this description makes it suitable not only for neutral file exchange, but also as a basis for implementing and sharing product databases and archiving.

This International Standard is organized as a series of parts, each published separately. The parts of ISO 10303 fall into one of the following series: description methods, integrated resources, application interpreted constructs, application protocols, abstract test suites, implementation methods, and conformance testing. The series are described in ISO 10303-1. This part of ISO 10303 is a member of the integrated resources series. Major subdivisions of this International Standard are:

- kinematic structure;
- kinematic motion representation;
- kinematic analysis control and results.

This part of ISO 10303 specifies an information model for the kinematic aspects of a mechanical product as required for the communication between CAD systems and kinematic analysis systems, and among dissimilar kinematic analysis systems. Kinematic information in the context of this part of ISO 10303 may be used in:

- early design stages, where detailed component shape is yet undetermined. The purpose of the kinematic description in these stages is to develop the conceptual model of the mechanical product to understand its motion characteristics;
- detail design stage where detailed shapes of components are determined. The purpose of the kinematic description in that stage is to verify the performance of the kinematic characteristics of a mechanical product using the final shape of its components, e.g., by means of collision checking.

The kinematic structure in this part of ISO 10303 is composed of rigid components related by kinematic pairs along a surface, on a curve, or at a point.

The kinematic structure schema defines the kinematic structure of rigid objects in terms of links, pairs, and joints. A link is the rigid part of a kinematic object. A pair represents the geometric aspects of the kinematic constraints of motion of these objects, and a joint represents the topological aspects of these constraints. The kinematic structures are represented by graphs in which the vertices of the graph correspond to the links, and the edges of the graph correspond to the joints.

The kinematic motion representation schema specifies the motion of a mechanical product by using a parametric path definition.

The kinematic analysis control and result schema specifies configurations of kinematic structures and the interpolation between configurations. This schema describes prescribed paths for a kinematic analysis and paths resulting from such a kinematic analysis.

Figure 1 illustrates these three schemas and the types and entities that are defined in one schema and referenced by another.

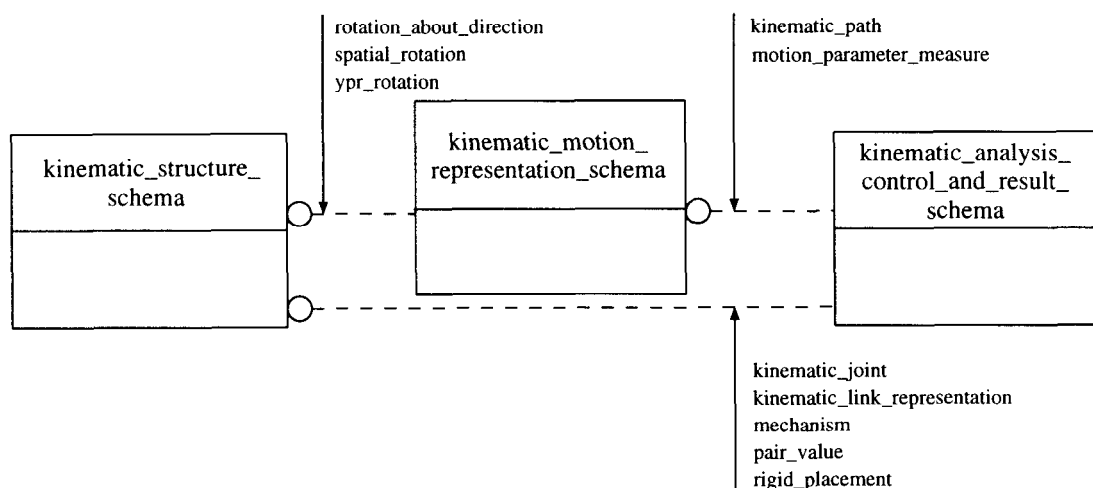


**Figure 1 – The relationships between the three schemas of
ISO 10303-105: Kinematics**

Industrial automation systems and integration — Product data representation and exchange — Part 105 : Integrated application resource: Kinematics

1 Scope

This part of ISO 10303 specifies the resource constructs for the representation of the kinematic aspects of a mechanical product.

The following are within the scope of this part of ISO 10303:

- definition of kinematic relationships between rigid objects called links;
- representation of the topology for a kinematic structure;
- definition of a kinematic motion as a sequence of discretised positions and orientations;
- representation of the input to and the result of a kinematic analysis.

The following are outside the scope of this part of ISO 10303:

- description of tolerances and clearances in kinematic structures;
- description of motion parameters in terms of a continuous time variable;
- representation of intermittent joints with variable constraints on motion;
- representation of dynamic mechanical assemblies;
- representation of force, mass, and friction.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO 10303. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this part of ISO 10303 are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO/IEC 8824-1:1995, *Information technology - Abstract Syntax Notation One (ASN.1): Specification of basic notation.*

ISO 8855:1991, *Road vehicles - Vehicle dynamics and road-holding ability - Vocabulary.*

ISO 10303-1:1994, *Industrial automation systems and integration - Product data representation and exchange - Part 1: Overview and fundamental principles.*

ISO 10303-11:1994, *Industrial automation systems and integration - Product data representation and exchange - Part 11: Description methods: The EXPRESS language reference manual.*

ISO 10303-21:1994, *Industrial automation systems and integration - Product data representation and exchange - Part 21: Implementation methods: Clear text encoding of the exchange structure.*

ISO 10303-41:1994, *Industrial automation systems and integration - Product data representation and exchange - Part 41: Integrated generic resources: Fundamentals of product description and support.*

ISO 10303-42:1994, *Industrial automation systems and integration - Product data representation and exchange - Part 42: Integrated generic resources: Geometric and topological representation.*

ISO 10303-43:1994, *Industrial automation systems and integration - Product data representation and exchange - Part 43: Integrated generic resources: Representation structures.*

3 Definitions

3.1 Terms defined in ISO 10303-1

This part of ISO 10303 makes use of the following terms defined in ISO 10303-1:

- application resource;
- data;
- generic resource;
- information;
- integrated resource;
- product.

3.2 Terms defined in ISO 10303-42

This part of ISO 10303 makes use of the following terms defined in ISO 10303-42:

- geometric coordinate system;

- geometrically founded;
- graph.

3.3 Terms defined in ISO 8855

This part of ISO 10303 makes use of the following terms defined in ISO 8855:

- pitch angle;
- roll angle;
- yaw angle.

3.4 Other definitions

For the purposes of this part of ISO 10303, the following definitions apply.

3.4.1 base: a link which is fixed to the ground or whose motion is prescribed along a defined path.

3.4.2 frame: a coordinate system used in kinematic representations.

3.4.3 ground: the immobile part of a mechanical product in the world coordinate system.

3.4.4 joint: the topological aspect of an ordered connection or motion constraint between two and only two links.

3.4.5 kinematics: the description of movable mechanical structures consisting of joints and links including the number, location, and orientation of the joints.

3.4.6 kinematic chain: a sequence of kinematic links connected by joints.

3.4.7 link: a rigid kinematic object whose motion is constrained by one or more joints.

3.4.8 link frame: the local coordinate system of a link, implicitly defined as the geometric context of the link, with respect to which all geometric definitions of the link are defined.

3.4.9 loop: a part of a mechanical structure whose joints and links together form a closed kinematic chain.

3.4.10 mechanism: a mechanical product whose motion is constrained by joints.

3.4.11 motion: change of position and orientation of a rigid body.

3.4.12 pair: the geometric aspect of either an ordered connection or motion constraint between two and only two links.

3.4.13 pair actuation: the assignment of a pair value to a kinematic pair by an application.

3.4.14 placement: the location and orientation of a frame.

3.4.15 transform: a representation of a placement that is specified with respect to an independent reference frame.

3.4.16 SU-parameters: the parameters that specify the placement between a pair of frames.

3.4.17 world coordinate system: the initial coordinate system from which all other frames are defined.

4 Symbols and abbreviations

For the purposes of this part of ISO 10303, the following symbols and abbreviations apply.

4.1 Mathematical symbols

The mathematical symbol convention used in this part of ISO 10303 is given in table 1.

4.2 Graphic convention for matrix representation

In mathematical formulas, a transform representation (three position values and three rotation values) is written as **BOLD** capital letters.

4.3 Abbreviations

SU	Sheth-Uicker
ypr	rotation angles called yaw, pitch, and roll.

Table 1 – Mathematical symbology

Symbol	Definition
a	Scalar quantity a
$\bar{a}$	Vector quantity $\bar{a}$
x, y, z	Coordinate axes
$\bar{x}, \bar{y}, \bar{z}$	Unit vectors along coordinate axes
$\mathbf{F}$	Matrix $\mathbf{F}$
$\mathbf{F}^{-1}$	The inverse of the matrix $\mathbf{F}$
$\times$	Vector (cross) product
$\cdot$	Scalar product
$f(u)$	Parametric curve f
$f(u) \Big _{u=u_p} = f(u_p)$	Value of $f(u)$ at the point denoted by p
$\frac{df(u)}{du}$	Derivative of $f(u)$ with respect to u
$f(u, v)$	Parametric surface f
$f(u, v) \Big _{u=u_p, v=v_p}$	Value of $f(u, v)$ at the point denoted by p
$\frac{\partial f(u, v)}{\partial u}$	Partial differential of f with respect to u
$x \in]a, b]$	x is element of the range between a (excluded) and b (included)
$sign(x)$	Function returns the sign of the argument x
$ $	Absolute value, or magnitude, or determinant
R^m	m -dimensional real space

5 Kinematic structure

The following *EXPRESS* declaration begins the **kinematic_structure_schema** and identifies the necessary references.

EXPRESS specification:

```
*)
SCHEMA kinematic_structure_schema;

REFERENCE FROM geometry_schema
  (axis2_placement_3d,
   cartesian_transformation_operator_3d,
   curve,
   direction,
   geometric_representation_context,
   normalise,
   point,
   point_on_curve,
   point_on_surface,
   surface,
   rectangular_trimmed_surface,
   trimmed_curve);

REFERENCE FROM measure_schema
  (conversion_based_unit,
   global_unit_assigned_context,
   length_measure,
   plane_angle_measure,
   si_prefix,
   si_unit,
   si_unit_name,
   unit);

REFERENCE FROM product_property_definition_schema
  (characterized_definition,
   property_definition);

REFERENCE FROM product_property_representation_schema
  (property_definition_representation);

REFERENCE FROM representation_schema
  (functionally_defined_transformation,
   item_defined_transformation,
   representation,
   representation_context,
   representation_item,
   representation_relationship,
   representation_relationship_with_transformation);

REFERENCE FROM support_resource_schema
  (bag_to_set,
```

```
label);  
(*
```

NOTES

1 – The schemas referenced above can be found in the following parts of ISO 10303:

geometry_schema	ISO 10303-42
measure_schema	ISO 10303-41
product_property_definition_schema	ISO 10303-41
product_property_representation_schema	ISO 10303-41
representation_schema	ISO 10303-43
support_resource_schema	ISO 10303-41

2 – See annex D for a graphical representation of this schema using the *EXPRESS-G* notation.

5.1 Introduction

The subject of the **kinematic_structure_schema** is the description of the kinematic information of a mechanical product defining its motion capabilities.

Motion in this part of ISO 10303 comprises both free, unconstrained motion of objects in a Cartesian coordinate system and motion constrained by kinematic joints and pairs.

This part of ISO 10303 covers the following types of pairs.

- 1) revolute pair;
- 2) prismatic pair;
- 3) screw pair;
- 4) cylindrical pair;
- 5) spherical pair;
- 6) universal pair;
- 7) planar pair;
- 8) gear pair;
- 9) rack and pinion pair;
- 10) unconstrained pair;
- 11) fully constrained pair;
- 12) point on surface pair;

- 13) sliding surface pair;
- 14) rolling surface pair;
- 15) point on planar curve pair;
- 16) sliding curve pair;
- 17) rolling curve pair.

Pair types 1 through 11 are called low order pairs and the remaining pair types are called high order pairs.

NOTE – Pair types 1 through 11 do not require explicit geometric information to describe their kinematic behaviour. Pair types 12 through 17 require a reference to a surface or to a curve in order to be defined kinematically.

For surface pairs and curve pairs, two specialisations are distinguished: the sliding and rolling subtypes. A rolling pair is constrained in the relative motion of the two surfaces or curves such that they cannot slide on each other.

EXAMPLE 1 – High levels of friction between two surfaces or curves may be the cause for relative motion to be constrained to rolling.

For sliding pairs such a constraint does not exist.

Low order pairs can be represented as sliding high order pairs. If a pair can be represented as a low order pair, it should be so represented and not as a high order pair.

5.2 Fundamental concepts and assumptions

5.2.1 Kinematic model structure

Mechanical products with kinematic representations may have arbitrary topological structures. The topological structures are not constrained to simple chains (i.e., open kinematic chains without branches) or to tree-type structures; they may include loops or a network structure.

NOTE – Every kinematic structure can be described in terms of its joints and links. Joints constrain the motion between two rigid objects called links. Joints and links are sufficient to describe the topology of kinematic structures, but for computation it is meaningful to introduce additional levels of structure based on graph theory. Information related to the graph representation of a kinematic structure is given in 5.4.70.

EXAMPLES

2 – Simple chains are suitable kinematic structures for many industrial robots or cranes. Tree-type structures are used to represent multi-arm manipulators.

3 – Network structures occur in industrial products. One example is the mechanism used for wind-screen wipers.

In this part of ISO 10303, kinematic structures are represented by graphs where the links represent the vertices of the graph, and the joints represent the edges. The rigid objects with kinematic representations are defined kinematically in terms of links and geometrically in terms of their associated shape representation. All coordinate systems related to a kinematic link are founded in the link frame, which is the geometric context of the related link representation. These coordinate systems are called frames in this part of ISO 10303. For the purpose of representing the kinematic aspect of a mechanism, the shape of the link is represented by the relative location and orientation of all its pair frames with respect to its link frame.

5.2.2 Positioning and placement

Whenever constraints are imposed on the relative motion of two kinematic links, this fact is topologically represented by a joint. The geometric aspect of this motion constraint is represented by the corresponding pair.

A placement describes a relative position and orientation of a frame with respect to another frame. There are different ways of representing a placement. Eight different representations of placement are used in this part of ISO 10303:

- rigid homogenous matrix;
- **axis2_placement_3d**;
- **transform** (see 6.4.2);
- **su_parameters**;
- **point_on_curve**;
- **point_on_surface**;
- a pair type with its **initial_state**;
- **cartesian_transformation_operator_3d**.

The rigid homogeneous matrix is used only as a notation in mathematical formulas. It defines the relative position and orientation of one frame with respect to another. The terms of this matrix are as follows:

$${}^k P_i = \begin{bmatrix} 3 \times 3 & \delta x \\ \text{rotational} & \delta y \\ \text{matrix} & \delta z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where the notation used for a frame **P** shall be ${}^k P_i$ with:

- k = identification of the reference frame in which the position of frame i is described;
- i = identification of the frame whose position is described with respect to frame k ;
- $\delta x, \delta y, \delta z$ = the components of the translation vector.

${}^0P_i$ represents the placement relative to the world coordinate system.

The transform includes three rotational and three translational components. The transform is the only form of placement representation which is suitable for interpolation or approximation via weighting functions.

The remaining representations are described in the schema.

5.2.3 Positive sense of rotations

For any rotation defined in the **kinematic_structure_schema**, a positive value of the rotation angle indicates a counter-clockwise rotation when looking in the negative direction with respect to the axis of that rotation.

5.2.4 Shape representations

Kinematic structure representations are frequently associated with shape representations for single parts of the structure at various levels of detail. In this part of ISO 10303, three kinds of shape representations are distinguished:

- shape representations of parts that belong to the immobile environment of a kinematic structure, referred to as the *ground shape*;
- shape representations describing the shape of a link of a kinematic structure, referred to as the *link shape*;
- shape representations describing specific aspects of the shape of a kinematic pair, referred to as the *pair shape*.

The **geometric_representation_items** which contribute to the ground shape representation may be in the **items** set either of the **kinematic_ground_representation** of the kinematic structure which, in this case, should be simultaneously a **shape_representation**, or of a **shape_representation** which is related to the **kinematic_ground_representation** of the kinematic structure by any means. This part of ISO 10303 does not specify any specific constraints on the representation of ground shape.

The **geometric_representation_items** contributing to a link shape representation shall not be in the **items** set of the corresponding **kinematic_link_representation**; rather they shall be in the **items** set of a **shape_representation** which is related to this **kinematic_link_representation** by means of a **kinematic_link_representation_association**. Any number of **shape_representations** may be related in this way to a specific **kinematic_link_representation**.

For the purpose of a pair shape representation, only **geometric_representation_items** of type **kinematic_frame_background**, i.e., **point**, **curve**, or **surface** are allowed. They shall be in the **items** set of a **kinematic_frame_background_representation** which, in turn, is related to one of the **kinematic_link_representations** of the pair by means of a **kinematic_frame_background_representation_association**. The **items** set of a **kinematic_frame_background_representation** may contain more than one item, and in general any number of

kinematic_frame_background_representations may be related to a **kinematic_link_representation** and to any frame in its **items** set. In the case of the higher pair types, however, which need explicit geometric information to describe their kinematic behaviour, at least one **kinematic_frame_background_representation** is required for each **curve** or **surface** that is referenced in its specification.

5.2.5 Consistent use of units

It is required that units are used consistently throughout a kinematic structure, although such a kinematic structure will include a variety of **representations** and related **representation-contexts**. To ensure this consistency, it is further required that all units which are to be used in a kinematic structure are assigned globally to the items of the respective **representations** by means of the **global_unit_assigned_context** subtype of the **representation_context** (see ISO 10303-41). Units for **measure_values** of the same kind shall be identical throughout all the **representations** in a kinematic structure.

EXAMPLE 4 – The pair placements of a **kinematic_pair** are founded in two different **kinematic_link_representations**. Consistent use of units requires that these two **kinematic_link_representations** share the same units via their **global_unit_assigned_contexts**.

5.3 kinematic_structure_schema type definitions

5.3.1 rigid_placement

The **rigid_placement** type is a means for referencing either an **axis2_placement_3d** or **su_parameters**.

EXPRESS specification:

```
*)
TYPE rigid_placement = SELECT
    (axis2_placement_3d,
     su_parameters);
END_TYPE;
(*
```

5.3.2 rotational_range_measure

The **rotational_range_measure** type is a means for referencing either a **plane_angle_measure** or an **unlimited_range**.

EXPRESS specification:

```
*)
TYPE rotational_range_measure = SELECT
    (plane_angle_measure,
     unlimited_range);
END_TYPE;
(*
```

5.3.3 translational_range_measure

The **translational_range_measure** type is a means for referencing either a **length_measure** or an **unlimited_range**.

EXPRESS specification:

```
*)
TYPE translational_range_measure = SELECT
    (length_measure,
     unlimited_range);
END_TYPE;
(*
```

5.3.4 unlimited_range

The **unlimited_range** type is a means by which an unlimited value can be assigned to a range bound of a pair.

EXPRESS specification:

```
*)
TYPE unlimited_range = ENUMERATION OF
    (unlimited);
END_TYPE;
(*
```

Enumerated item definitions:

unlimited: an indication that the range bound is unlimited.

5.3.5 spatial_rotation

The **spatial_rotation** type enables an arbitrary rotation in 3D space (R^3) to be specified either as a **ypr_rotation** or as a **rotation_about_direction**.

EXPRESS specification:

```
*)
TYPE spatial_rotation = SELECT
    (ypr_rotation,
     rotation_about_direction);
END_TYPE;
(*
```

5.3.6 ypr_enumeration

The **ypr_enumeration** is a means by which the angles of rotation about the origin in Cartesian coordinate space R^3 are distinguished. **Yaw** represents the angle by which the original frame (x_0, y_0, z_0) is rotated first about the z-axis to yield an intermediate (x', y', z') frame. **Pitch** represents the angle by which the intermediate frame (x', y', z') then is rotated about the y' -axis to yield another intermediate (x'', y'', z'') frame. **Roll** represents the angle by which the (x'', y'', z'') frame is rotated about the x'' -axis to yield the desired final (x, y, z) frame.

NOTE – For more detailed information on yaw, pitch, and roll convention see ISO 8855:1991 (clause 2.2).

EXPRESS specification:

```
*)
TYPE ypr_enumeration = ENUMERATION OF
    (yaw,
     pitch,
     roll);
END_TYPE;
(*)
```

Enumerated item definitions:

yaw: the angle of rotation about the z-axis.

pitch: the angle of rotation about the y' -axis.

roll: the angle of rotation about the x'' -axis.

5.3.7 ypr_rotation

The **ypr_rotation** type is a means for specifying an ordered sequence of the angles of rotation. The **yaw** value is the first, the **pitch** value the second, and the **roll** value the last element in the array. The yaw rotation shall be performed first, followed by the pitch rotation. The roll rotation shall be performed as the last action of a **ypr_rotation**.

NOTE – An *EXPRESS* function which returns the rotation matrix corresponding to an input **ypr_rotation** is given in annex E. This function has been provided for applications that use this part of ISO 10303, but need the rotational transformation to be expressed in terms of a rotation matrix rather than in terms of a sequence of rotation angles.

EXPRESS specification:

```
*)
TYPE ypr_rotation = ARRAY [ypr_index(yaw) : ypr_index(roll)]
    OF plane_angle_measure;
END_TYPE;
(*)
```

Informal propositions:

angle bounds: The values of the yaw, pitch, and roll angle shall be in the ranges equivalent to the following ranges, where the angles are measured in radians:

$$\begin{aligned} \text{yaw angle} &\in] - \pi, \pi] \\ \text{pitch angle} &\in [-\pi/2, \pi/2] \\ \text{roll angle} &\in] - \pi, \pi] \end{aligned}$$

rectangular pitch angle: If the absolute value of the pitch angle is equal to the equivalent value of $\pi/2$, the roll angle shall be zero.

5.3.8 kinematic_frame_background

The **kinematic_frame_background** type is a means for referencing either a **point**, a **curve**, or a **surface**.

EXPRESS specification:

```
*)
TYPE kinematic_frame_background = SELECT
  (point,
   curve,
   surface);
END_TYPE;
(*
```

5.4 kinematic_structure_schema entity definitions

5.4.1 rotation_about_direction

A **rotation_about_direction** specifies a rotation in Cartesian coordinate space R^3 . The axis of rotation passes the origin of the related coordinate system; its direction is given explicitly as **direction_of_axis**. Furthermore, the amount of rotation is specified by **rotation_angle** which may have any finite value; i.e., its absolute value in radians is not restricted to be less than π . A positive value of **rotation_angle** indicates a counter-clockwise rotation when looking in the negative direction with respect to the **direction_of_axis**.

EXPRESS specification:

```
*)
ENTITY rotation_about_direction;
  direction_of_axis : direction;
  rotation_angle    : plane_angle_measure;
WHERE
  WR1: SIZEOF (direction_of_axis.direction_ratios) = 3;
END_ENTITY;
(*
```

Attribute definitions:

direction_of_axis: the three-dimensional direction of the axis of rotation.

rotation_angle: the angle of rotation about the axis of rotation.

Formal propositions:

WR1: The **direction_of_axis** shall be a three-dimensional direction.

5.4.2 kinematic_property_definition

The **kinematic_property_definition** specifies the kinematic property of a product.

EXPRESS specification:

```

*)
ENTITY kinematic_property_definition
  SUBTYPE OF (property_definition);
  ground_definition : characterized_definition;
END_ENTITY;
(*

```

Attribute definitions:

ground_definition: the **characterized_object**, **characterized_product_definition**, or **shape_definition** related to the **kinematic_ground_representation** which is associated with the **kinematic_property_definition**.

NOTE – The **ground_definition** may be a **characterized_object** if it does not need to be associated with a product definition.

5.4.3 kinematic_property_representation_relation

A **kinematic_property_representation_relation** is a **property_definition_representation** that relates a **kinematic_ground_representation** to a **kinematic_property_definition**.

EXPRESS specification:

```

*)
ENTITY kinematic_property_representation_relation
  SUBTYPE OF (property_definition_representation);
UNIQUE
  UR1: SELF\property_definition_representation.definition;
WHERE
  WR1: 'KINEMATIC_STRUCTURE_SCHEMA.KINEMATIC_PROPERTY_DEFINITION'
      IN TYPEOF (SELF\property_definition_representation.definition);
  WR2: 'KINEMATIC_STRUCTURE_SCHEMA.KINEMATIC_GROUND_REPRESENTATION'
      IN TYPEOF (SELF\property_definition_representation.used_representation);
END_ENTITY;
(*

```

Formal propositions:

UR1: A **kinematic_property_definition** shall be related to a **kinematic_ground_representation** by at most one **kinematic_property_representation_relation**.

WR1: The **definition** of a **kinematic_property_representation_relation** shall be a **kinematic_property_definition**.

WR2: The **used_representation** of a **kinematic_property_representation_relation** shall be a **kinematic_ground_representation**.

5.4.4 kinematic_ground_representation

A **kinematic_ground_representation** specifies the world coordinate system.

NOTE – A **shape_representation** may be given together with a **kinematic_ground_representation** to specify the shape of those parts of a **kinematic_property_definition** that are immobile.

EXPRESS specification:

```
*)
ENTITY kinematic_ground_representation
  SUBTYPE OF (representation);
INVERSE
  property : kinematic_property_representation_relation FOR used_representation;
WHERE
  WR1: 'GEOMETRY_SCHEMA.GEOMETRIC_REPRESENTATION_CONTEXT' IN
    TYPEOF (SELF\representation.context_of_items);
END_ENTITY;
(*
```

Attribute definitions:

SELF\representation.items: a set of **representation_items** that are related in the **context_of_items**.

ground_context: the identification of a world coordinate system against which all placements in the **kinematic_property_definition** are measured, either directly or indirectly.

property: the **kinematic_property_representation_relation** that relates the **kinematic_ground_representation** to the **kinematic_property_definition** to which it belongs.

Formal propositions:

WR1: The **context_of_items** of a **kinematic_ground_representation** shall always be a **geometric_representation_context**.

Informal propositions:

ground definition: The **kinematic_ground_representation** shall be associated to **property\property_definition_representation.definition.ground_definition**.

5.4.5 mechanism

A **mechanism** specifies the capabilities of relative motion of its links under the constraints imposed by its joints. It is a kinematic structure with a single link fixed to the ground or to some other **mechanism**. The entire **mechanism** may be subject to the rigid body motion imposed to its **base**.

NOTE – A mechanism may have different bases at different times. In space applications, worm-like mechanisms have been designed which grab to a space-structure with one end, move on with the other end, fix themselves with an end-effector establishing a new base, and then loosen the original grip.

EXPRESS specification:

```
*)
ENTITY mechanism;
  structure_definition : kinematic_structure;
  base                 : kinematic_link;
```

```

    containing_property : kinematic_property_definition;
WHERE
    WR1: SIZEOF (QUERY (joint <* structure_definition.joints |
                        (base :=: joint.first_link) OR
                        (base :=: joint.second_link) )) > 0;
END_ENTITY;
(*

```

Attribute definitions:

structure_definition: the topological structure of the **mechanism**.

base: the initial link of the **mechanism**.

containing_property: the **kinematic_property_definition** to which the **mechanism** belongs.

Formal propositions:

WR1: The **base** shall be either the first link or the second link in at least one joint in the set **structure_definition.joints**. This ensures that the **base** is within the kinematic structure.

5.4.6 mechanism_base_placement

A **mechanism_base_placement** is a means to define the placement of the link frame of the **base** of a mechanism either relative to the world coordinate system or relative to the local coordinate system of a link that belongs to another **mechanism**.

NOTE 1 – If the placement is defined relative to the link of another **mechanism**, the **mechanism** whose **base** is being placed is said to be mounted on the other **mechanism**.

If the placement is defined relative to the world coordinate system, **mechanism_base_placement** relates the link frame of the **base** in the **context_of_items** of a **kinematic_ground_representation**. If the placement is defined relative to the local coordinate system of another link, **mechanism_base_placement** relates the link frame of the **base** in the **link_frame** of the **kinematic_link_representation** which is associated with this other link.

EXPRESS specification:

```

*)
ENTITY mechanism_base_placement
  SUBTYPE OF (representation_relationship_with_transformation);
  base_of_mechanism      : mechanism;
  SELF\representation_relationship_with_transformation.
    transformation_operator : cartesian_transformation_operator_3d;
DERIVE
  SELF\representation_relationship.rep_2
                                : kinematic_link_representation
                                := representation_of_link (base_of_mechanism.base);
UNIQUE
  UR1: base_of_mechanism;
WHERE
  WR1: ('KINEMATIC_STRUCTURE_SCHEMA.KINEMATIC_GROUND_REPRESENTATION' IN
        TYPEOF (SELF\representation_relationship.rep_1))

```

```

OR
('KINEMATIC_STRUCTURE_SCHEMA.KINEMATIC_LINK_REPRESENTATION' IN
  TYPEOF (SELF\representation_relationship.rep_1));
WR2: suitably_based_mechanism (SELF, base_of_mechanism);
WR3: SELF\representation_relationship_with_transformation.
      transformation_operator\representation_item IN
      SELF\representation_relationship.rep_1.items;
END_ENTITY;
(*)

```

Attribute definitions:

base_of_mechanism: the **mechanism** whose **base** is being placed.

SELF\representation_relationship_with_transformation.transformation_operator:
the transformation between the context of **rep_1** and the context of **rep_2**.

NOTE 2 – Legal values for the attributes of the **cartesian_transformation_operator_3d** that is used as the **transformation_operator**, are specified in application protocols which use this entity.

SELF\representation_relationship.rep_2: the **kinematic_link_representation** that is related to the **base** of **base_of_mechanism**. This is derived from **base_of_mechanism.base** by the function **representation_of_link**.

Formal propositions:

UR1: A **mechanism** shall be the **base_of_mechanism** for at most one **mechanism_base_placement**.

WR1: The **rep_1** of a **mechanism_base_placement** shall be a **kinematic_ground_representation** or a **kinematic_link_representation**.

WR2: The **mechanism_base_placement** shall place the **base** of **base_of_mechanism** either with respect to the **kinematic_ground_representation** which is associated with its **containing_property**, or with respect to a **kinematic_link_representation** that belongs to another **mechanism**. This other **mechanism** shall also be in the **mechanisms** set of **containing_property**. In this way, the **base** of **base_of_mechanism** shall be placed with respect to the **kinematic_ground_representation**, directly or indirectly. If the placement is done indirectly, a **mechanism** shall not participate in the placement of its own **base**. These conditions are checked by the function **suitably_based_mechanism**.

WR3: The **cartesian_transformation_operator_3d** that is used as the **transformation_operator** in a **mechanism_base_placement** shall be in the **items** set of **rep_1**.

5.4.7 initial_state

The **initial_state** specifies initial values of the pair parameters of a **mechanism**.

NOTE – Some types of kinematic structures, such as four bar linkages, can only be assembled uniquely if configuration information is provided in addition to shape parameters, such as link

lengths. In those cases the **initial_state** provides enough pair parameter information to make the configuration unique.

EXPRESS specification:

```
*)
ENTITY initial_state;
    applies_to_mechanism : mechanism;
    pair_values           : SET [ 1 : ?] OF pair_value;
WHERE
    WR1: SIZEOF (QUERY (joint <* applies_to_mechanism.structure_definition.joints |
        SIZEOF (QUERY (init_val <* pair_values |
            init_val.applies_to_pair.joint :=: joint)) <> 1)) = 0;
END_ENTITY;
(*
```

Attribute definitions:

applies_to_mechanism: the **mechanism** to which the **initial_state** applies.

pair_values: the pair parameter values of each pair in the **mechanism**.

Formal propositions:

WR1: An **initial_state** shall provide in its set of **pair_values** exactly one **pair_value** for each **kinematic_pair** in the **mechanism** to which it applies.

5.4.8 kinematic_structure

The **kinematic_structure** specifies the topological aspects of a kinematic representation. The topology of the **kinematic_structure** is represented by means of **kinematic_joints** which, in turn, establish a relationship between pairs of **kinematic_links**.

NOTE – Every kinematic structure can be described in terms of its joints and links. Joints constrain the motion between two rigid objects called links. Joints and links are sufficient to describe the topology of kinematic structures, but for computation it is meaningful to introduce additional levels of structure based on graph theory. Information related to the graph representation of a kinematic structure is given in 5.4.70.

EXPRESS specification:

```
*)
ENTITY kinematic_structure;
    joints : SET [1 : ?] OF kinematic_joint;
END_ENTITY;
(*
```

Attribute definitions:

joints: the set of **kinematic_joints** that compose the **kinematic_structure** of a **mechanism**.

5.4.9 kinematic_joint

The **kinematic_joint** specifies the topological aspect of an ordered connection between two and only two links.

A topological orientation is always implicitly associated with a joint. This orientation is defined as being from the first link to the second link.

For a **kinematic_pair** referencing a **kinematic_joint**, the **pair_placement_in_first_link_context** shall be an item of the **kinematic_link_representation** of the **first_link** of the **kinematic_joint**.

Similarly, the **pair_placement_in_second_context** shall be an item of the **kinematic_link_representation** of the **second_link** of the **kinematic_joint**.

NOTES

1 – The type of constraint on the relative motion is defined by the **kinematic_pair** that has the joint as its **joint** attribute. In a graph representation of the topology, the **kinematic_joint** represents an oriented edge connecting two vertices in the graph. The orientation of the **kinematic_joint** reflects an implicit orientation which is arbitrary, except for the rack-and-pinion pair, the point-on-surface pair, and the point-on-planar-curve pair.

2 – If the sequence of **first_link** and **second_link** is interchanged, both, pair values and pair range values, have to be used with the opposite signs, and in the subtypes of **simple_pair_range** additionally the attributes giving a lower limit are to be interchanged with those giving the corresponding upper limit.

EXPRESS specification:

```
*)
ENTITY kinematic_joint ;
    first_link : kinematic_link;
    second_link : kinematic_link;
INVERSE
    structure : kinematic_structure FOR joints;
WHERE
    WR1: first_link :<>: second_link;
END_ENTITY;
(*
```

Attribute definitions:

first_link: the link through which the joint is entered.

second_link: the link through which the joint is left.

structure: the **kinematic_structure** containing the **kinematic_joint** in its set of **joints**.

Formal propositions:

WR1: **first_link** and **second_link** shall not be identical.

5.4.10 kinematic_link

A **kinematic_link** represents the topological aspects associated with a rigid part of a **mechanism**.

EXPRESS specification:

```

*)
ENTITY kinematic_link;
WHERE
  WR1: SIZEOF (USEDIN (SELF,
    'KINEMATIC_STRUCTURE_SCHEMA.KINEMATIC_JOINT.FIRST_LINK') +
    USEDIN (SELF,
    'KINEMATIC_STRUCTURE_SCHEMA.KINEMATIC_JOINT.SECOND_LINK')) > 0;
  WR2: unique_link_usage (SELF);
END_ENTITY;
(*)

```

Formal propositions:

WR1: A **kinematic_link** shall be the **first_link** or the **second_link** of at least one **kinematic_joint**.

WR2: A **kinematic_link** shall be a constituent of one mechanism only and shall occur only once in the kinematic structure of that mechanism.

5.4.11 kinematic_link_representation_relation

A **kinematic_link_representation_relation** relates the geometric aspects of a **kinematic_link** with its topological aspects.

EXPRESS specification:

```

*)
ENTITY kinematic_link_representation_relation;
  topological_aspects : kinematic_link;
  geometric_aspects   : kinematic_link_representation;
UNIQUE
  UR1: topological_aspects;
END_ENTITY;
(*)

```

Attribute definitions:

topological_aspects: the **kinematic_link** whose geometric aspects are represented by the **kinematic_link_representation**.

geometric_aspects: the representation of the context of the rigid part of a mechanical product and of the frames related to it for a **kinematic_link**.

Formal propositions:

UR1: A **kinematic_link** shall be related to a **kinematic_link_representation** by at most one **kinematic_link_representation_relation**.

5.4.12 kinematic_link_representation

A **kinematic_link_representation** specifies the geometric aspects of a **kinematic_link**. The context of the representation is established by the **link_frame** attribute.

EXPRESS specification:

```

*)
ENTITY kinematic_link_representation
  SUBTYPE OF (representation);
  SELF\representation.context_of_items : geometric_representation_context;
DERIVE
  link_frame
    : geometric_representation_context
    := SELF\representation.context_of_items;
INVERSE
  link_representation_relation : kinematic_link_representation_relation FOR
    geometric_aspects;
WHERE
  WR1: SIZEOF (QUERY (item <* SELF\representation.items |
    NOT (('KINEMATIC_STRUCTURE_SCHEMA.RIGID_PLACEMENT' IN
      TYPEOF (item))
    OR
    ('GEOMETRY_SCHEMA.CARTESIAN_TRANSFORMATION_OPERATOR_3D' IN
      TYPEOF (item)))) = 0;
END_ENTITY;
(*

```

Attribute definitions:

SELF\representation.items: a set of **representation_items** that are related in the **link_frame**. It contains at least one pair frame that is used by a **kinematic_pair** as **pair_placement_in_first_link_context** or as **pair_placement_in_second_link_context**. It may contain any number of **cartesian_transformation_operator_3ds** which serve as the **transformation_operator** of a **mechanism_base_placement**. Furthermore, it may contain additional frames that are attached to the link.

NOTE – The **items** set of a **kinematic_link_representation** does not contain **geometric_representation_items** that define the shape of the link.

SELF\representation.context_of_items: the **geometric_representation_context** providing the context for the **items**.

link_frame: the link frame of the related **kinematic_link**.

link_representation_relation: the **kinematic_link_representation_relation** that relates a **kinematic_link** to the **kinematic_link_representation**.

Formal propositions:

WR1: All elements in the **items** set of a **kinematic_link_representation** shall be of type **rigid_placement** or of type **cartesian_transformation_operator_3d**.

5.4.13 kinematic_link_representation_association

A **kinematic_link_representation_association** is a **representation_relationship** that associates a **representation** with a **kinematic_link_representation**.

NOTE – A **kinematic_link_representation_association** is used to define the shape of a link by associating a **shape_representation** to the corresponding **kinematic_link_representation**.

EXPRESS specification:

```
*)
ENTITY kinematic_link_representation_association
  SUBTYPE OF (representation_relationship);
  SELF\representation_relationship.rep_1 : kinematic_link_representation;
WHERE
  WR1: SELF\representation_relationship.rep_2.context_of_items :=:
    SELF\representation_relationship.rep_1\representation.context_of_items;
  WR2: SIZEOF ([ 'KINEMATIC_STRUCTURE_SCHEMA.KINEMATIC_GROUND_REPRESENTATION',
    'KINEMATIC_STRUCTURE_SCHEMA.KINEMATIC_LINK_REPRESENTATION' ] *
    TYPEOF (SELF\representation_relationship.rep_2)) = 0;
END_ENTITY;
(*
```

Attribute definitions:

SELF\representation_relationship.rep_1: the **kinematic_link_representation** with which **rep_2** is associated.

Formal propositions:

WR1: The **context_of_items** of **rep_2** shall be identical to the **link_frame** of the **kinematic_link_representation** with which **rep_2** is associated.

WR2: Neither a **kinematic_ground_representation** nor a **kinematic_link_representation** shall be associated with a **kinematic_link_representation** by this entity.

5.4.14 kinematic_frame_background_representation

A **kinematic_frame_background_representation** specifies the shape aspects which are associated with a **kinematic_pair**.

EXPRESS specification:

```
*)
ENTITY kinematic_frame_background_representation
  SUBTYPE OF (representation);
  SELF\representation.items : SET [1 : ?] OF
    kinematic_frame_background;
  SELF\representation.context_of_items : geometric_representation_context;
WHERE
  WR1: SELF\representation.context_of_items\
    geometric_representation_context.coordinate_space_dimension = 3;
END_ENTITY;
(*
```

Attribute definitions:

SELF\representation.items: a set of instances of **kinematic_frame_background** which specify shapes that are to be associated with a **kinematic_pair**.

SELF\representation.context_of_items: the context in which the **items** are founded.

Formal propositions:

WR1: The **context_of_items** of a **kinematic_frame_background_representation** shall be three-dimensional.

5.4.15 kinematic_frame_based_transformation

The **kinematic_frame_based_transformation** is a **functionally_defined_transformation** which uses a **rigid_placement** as the transformation function.

EXPRESS specification:

```
*)
ENTITY kinematic_frame_based_transformation
  SUBTYPE OF (functionally_defined_transformation);
  transformer_frame : rigid_placement;
END_ENTITY;
(*
```

Attribute definitions:

transformer_frame: the **rigid_placement** which is used as the transformation function for the **kinematic_frame_based_transformation**.

5.4.16 kinematic_frame_background_representation_- association

A **kinematic_frame_background_representation_association** is a **representation_relationship_with_transformation** which uses a **kinematic_frame_based_transformation** as the **transformation_operator**. The **transformer_frame** of this operator is founded in the **kinematic_link_representation** used as **rep_1**; the representation being transformed is a **kinematic_frame_background_representation** which is referenced by **rep_2**.

EXPRESS specification:

```
*)
ENTITY kinematic_frame_background_representation_association
  SUBTYPE OF (representation_relationship_with_transformation);
  SELF\representation_relationship_with_transformation.
    transformation_operator : kinematic_frame_based_transformation;
WHERE
  WR1: 'KINEMATIC_STRUCTURE_SCHEMA.KINEMATIC_LINK_REPRESENTATION' IN
    TYPEOF (SELF\representation_relationship.rep_1);
  WR2: 'KINEMATIC_STRUCTURE_SCHEMA.KINEMATIC_FRAME_BACKGROUND_REPRESENTATION'
    IN TYPEOF (SELF\representation_relationship.rep_2);
  WR3: SELF\representation_relationship_with_transformation.
    transformation_operator\kinematic_frame_based_transformation.
    transformer_frame\representation_item IN
    SELF\representation_relationship.rep_1.items;
END_ENTITY;
(*
```

Attribute definitions:

SELF\representation_relationship_with_transformation.transformation_operator:
the transformation operator for the **kinematic_frame_background_representation_association**.

Formal propositions:

WR1: The first **representation** related in a **kinematic_frame_background_representation_association** shall be a **kinematic_link_representation**.

WR2: The second **representation** related in a **kinematic_frame_background_representation_association** shall be a **kinematic_frame_background_representation**.

WR3: The **rigid_placement** which serves as the **transformer_frame** in the **transformation_operator** of a **kinematic_frame_background_representation_association** shall be in the **items** set of the **kinematic_link_representation** used as **rep_1**.

5.4.17 su_parameters

su_parameters are an alternative method to **axis2_placement_3d** for specifying the placement of pairs on a link.

To describe the kinematic behaviour, the full shape representation of a link is not needed; rather it is sufficient to characterize the link by the transition from the link frame to all the pair frames. This shape information is captured by a set of characteristic parameters called Sheth-Uicker-Parameters, henceforth abbreviated “SU-parameters”.

NOTE 1 – See figure 2. For a **kinematic_link** with two pairs and for the link frame coinciding with the frame of one of the pairs, this may be interpreted as the transition from the frame at the beginning of the link to the frame at the following end.

The following notation is used for frame definitions:

x_k^r, y_k^r, z_k^r	coordinate axes of a link frame;
$\bar{o}_k^r$	origin of that link frame;
x_k, y_k, z_k	axes of the frame of a kinematic pair on that link;
$\bar{o}_k$	origin of that pair frame.

The following auxiliary vectors are introduced:

$\bar{t}_k$	unit vector along the common perpendicular between the two axes z_k^r and z_k , directed from z_k^r towards z_k . The common perpendicular itself is denoted by t_k ;
$\bar{s}_k^r$	intersection point of the common perpendicular t_k with z_k^r ;
$\bar{s}_k$	intersection point of the common perpendicular t_k with z_k .

Then the following notation applies to SU-parameters:

a_k	the (positive) distance from z_k^r to z_k , given by
-------	--

$$a_k = |\bar{s}_k - \bar{s}_k^r|$$

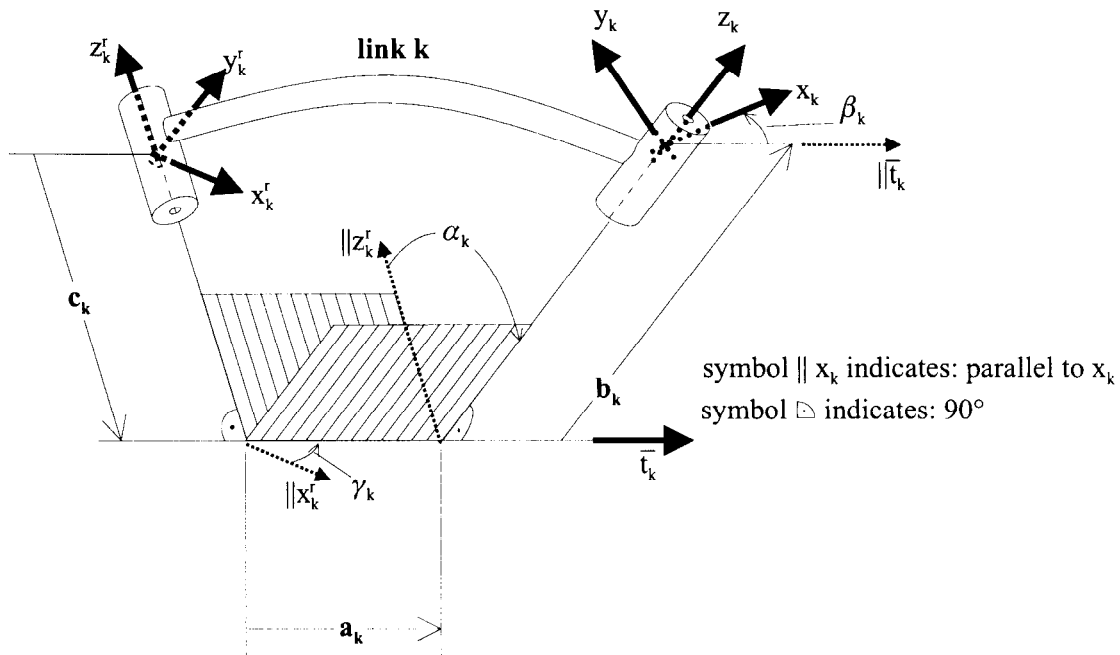


Figure 2 – Definition of the su_parameters

α_k the angle from positive z_k^r to positive z_k , measured in the mathematically positive sense about $\bar{t}_k$ and determined by

$$\sin \alpha_k = (\bar{z}_k^r \times \bar{z}_k) \cdot \bar{t}_k; \quad \cos \alpha_k = \bar{z}_k^r \cdot \bar{z}_k$$

b_k the deviation of $\bar{o}_k$ from $\bar{s}_k$, measured along positive z_k and given by

$$b_k = (\bar{o}_k - \bar{s}_k) \cdot \bar{z}_k$$

β_k the angle from positive $\bar{t}_k$ to positive x_k , measured in the mathematically positive sense about positive z_k and determined by

$$\sin \beta_k = (\bar{t}_k \times \bar{x}_k) \cdot \bar{z}_k; \quad \cos \beta_k = \bar{t}_k \cdot \bar{x}_k$$

c_k the deviation of $\bar{s}_k^r$ from $\bar{o}_k^r$, measured along positive z_k^r and given by

$$c_k = (\bar{s}_k^r - \bar{o}_k^r) \cdot \bar{z}_k^r$$

γ_k the angle from positive x_k^r to positive $\bar{t}_k$, measured in the mathematically positive sense about positive z_k^r and determined by

$$\sin \gamma_k = (\bar{x}_k^r \times \bar{t}_k) \cdot \bar{z}_k^r; \quad \cos \gamma_k = \bar{x}_k^r \cdot \bar{t}_k$$

NOTES

2 – For a detailed description of the Sheth-Uicker parameters see [1].

3 – In industrial robot technology the Denavit-Hartenberg parameters (DH-parameters) are widely

used. See [2]. For kinematic structures with loops it is important to represent the information about the relative placement of the link axes separately from the present state of the pair. In contrast to DH-parameters, SU-parameters provide for this separation.

4 – See annex F for information on the replacement of DH-parameters by SU-parameters.

5 – An instance of **su_parameters** with SU-parameters as defined above is equivalent to an instance of **axis2_placement_3d** that has the following attribute values:

SELF\placement.location:	cartesian_point $((a_k \cos \gamma_k + b_k \sin \gamma_k \sin \alpha_k), (a_k \sin \gamma_k - b_k \cos \gamma_k \sin \alpha_k), (c_k + b_k \cos \alpha_k))$;
axis:	direction $((\sin \gamma_k \sin \alpha_k), (-\cos \gamma_k \sin \alpha_k), \cos \alpha_k)$;
ref_direction:	direction $((\cos \gamma_k \cos \beta_k - \sin \gamma_k \cos \alpha_k \sin \beta_k), (\sin \gamma_k \cos \beta_k + \cos \gamma_k \cos \alpha_k \sin \beta_k), (\sin \alpha_k \sin \beta_k))$.

EXPRESS specification:

```
*)
ENTITY su_parameters
  SUBTYPE OF (representation_item);
  a      : length_measure;
  alpha  : plane_angle_measure;
  b      : length_measure;
  beta   : plane_angle_measure;
  c      : length_measure;
  gamma  : plane_angle_measure;
END_ENTITY;
(*
```

Attribute definitions:

a: SU-parameter a_k .

alpha: SU-parameter α_k .

b: SU-parameter b_k .

beta: SU-parameter β_k .

c: SU-parameter c_k .

gamma: SU-parameter γ_k .

5.4.18 kinematic_pair

A **kinematic_pair** defines the kinematic constraints between two adjacent links coinciding at a joint.

EXPRESS specification:

```

*)
ENTITY kinematic_pair
  SUBTYPE OF (item_defined_transformation);
  joint : kinematic_joint;
DERIVE
  pair_placement_in_first_link_context
    : rigid_placement
    := SELF\item_defined_transformation.transform_item_1;
  pair_placement_in_second_link_context
    : rigid_placement
    := SELF\item_defined_transformation.transform_item_2;
UNIQUE
  UR1: joint;
WHERE
  WR1: coordinated_pair_link_representation
    (joint.first_link, pair_placement_in_first_link_context);
  WR2: coordinated_pair_link_representation
    (joint.second_link, pair_placement_in_second_link_context);
END_ENTITY;
(*)

```

Attribute definitions:

joint: the joint whose motion is constrained by the **kinematic_pair**.

pair_placement_in_first_link_context: the placement of the pair frame on the first link in the context of the first link, i.e., its link frame.

pair_placement_in_second_link_context: the placement of the pair frame on the second link in the context of the second link, i.e., its link frame.

Formal propositions:

UR1: Each **kinematic_pair** shall apply to a different **joint**.

WR1: There shall be a **kinematic_link_representation** related to **joint.first_link**, and it shall contain **pair_placement_in_first_link_context** in its **items** set. This condition is checked by the function **coordinated_pair_link_representation**.

WR2: There shall be a **kinematic_link_representation** related to **joint.second_link**, and it shall contain **pair_placement_in_second_link_context** in its **items** set. This condition is checked by the function **coordinated_pair_link_representation**.

NOTES

1 – The relative position of the second link with respect to the first link is defined on the basis of three frames:

- one frame, rigidly connected to the first link and placed in the context (i.e., with respect to the link frame) of the first link;

- another frame, rigidly connected to the second link and placed in the context (i.e., with respect to the link frame) of the second link;
- a contact frame which specifies the point and plane of contact between the two links. This contact frame may or may not be physically connected with one of the links.

The type of the pair, together with the pair parameters, defines the placement of the contact frame relative to the first pair frame as well as the placement of the contact frame relative to the second pair frame. Thus, indirectly the placement of the two pair frames relative to each other is defined. See figure 3.

2 – The goal of a kinematic analysis of a mechanism is to determine the link frames for all the links of that mechanism such that they are consistent with the actual pair parameters. The consistency condition is as follows:

Consider one pair which connects two links and let

- j indicate the first link of that pair;
- k indicate the second link of that pair;
- p indicate that pair as seen from the first link;
- q indicate that pair as seen from the second link;
- c indicate the common contact point (which may or may not be physically connected with both links);
- ${}^0\mathbf{LF}_j$ and ${}^0\mathbf{LF}_k$ be the rigid homogeneous matrix representation of the link frames of those links (both relative to the world coordinate system);
- ${}^j\mathbf{PF}_p$ and ${}^k\mathbf{PF}_q$ be the pair placements of pair p with respect to the first and second link of that pair, respectively,

then

${}^0\mathbf{PF}_p = {}^0\mathbf{LF}_j {}^j\mathbf{PF}_p$ is the rigid homogeneous matrix placing the first pair frame relative to the world coordinate system while

${}^p\mathbf{PF}_c$ is the placement for the contact frame relative to the first pair frame. This placement is determined by the type of the pair and its actual pair parameters.

${}^q\mathbf{PF}_c$ is the placement for the contact frame relative to the second pair frame. This placement is determined by the type of the pair and its actual pair parameters.

Consistency now requires that

$${}^0\mathbf{LF}_j {}^j\mathbf{PF}_p {}^p\mathbf{PF}_c = {}^0\mathbf{LF}_k {}^k\mathbf{PF}_q {}^q\mathbf{PF}_c$$

Or else, for the second link placement relative to the world coordinate system is obtained

$${}^0\mathbf{LF}_k = {}^0\mathbf{LF}_j {}^j\mathbf{PF}_p {}^p\mathbf{PF}_c {}^q\mathbf{PF}_c^{-1} {}^k\mathbf{PF}_q^{-1}$$

3 – For lower pairs the contact frame and the second pair frame shall be treated as identical. Hence, for lower pairs the rigid homogeneous matrix representation of ${}^q\mathbf{PF}_c$ is a unit matrix.

5.4.19 pair_actuator

The **pair_actuator** identifies a **kinematic_pair** that is actuated. All degrees of freedom of a **kinematic_pair** that is identified by a **pair_actuator** are treated as being actuated.

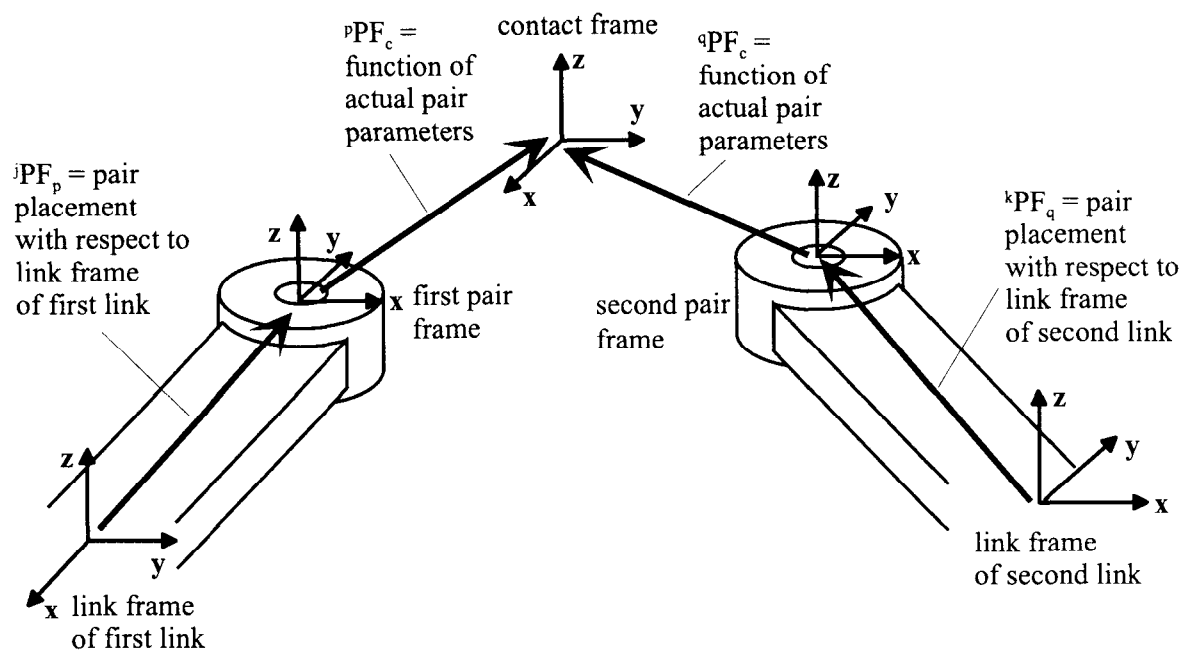


Figure 3 – Placement relationships of the pair frames relative to the mating links

NOTES

1 – For a backward kinematic analysis the actuated pairs are used to achieve a prescribed path or a prescribed position. For a forward kinematic analysis pair values may be prescribed only for actuated pairs.

2 – This part of ISO 10303 does not support actuation of individual degrees of freedom of a kinematic pair. Modelling of kinematic systems needs to be performed such that a kinematic pair is either fully actuated or not at all. Mixed situations such as, for example, a cylindrical pair can be avoided by using a revolute pair and a prismatic pair separately.

EXPRESS specification:

```
*)
ENTITY pair_actuator;
    actuated_pair : kinematic_pair;
    name          : label;
UNIQUE
    UR1: actuated_pair;
END_ENTITY;
(*)
```

Attribute definitions:

actuated_pair: the kinematic pair which is actuated.

name: the word or group of words by which the **pair_actuator** is referred to.

Formal propositions:

UR1: Each **pair_actuator** shall apply to a different **actuated_pair**.

5.4.20 pair_value

The **pair_value** specifies the configuration of the two links that join a **kinematic_pair**.

EXPRESS specification:

```
*)  
ENTITY pair_value;  
    applies_to_pair: kinematic_pair;  
END_ENTITY;  
(*
```

Attribute definitions:

applies_to_pair: the **kinematic_pair** to which the **pair_value** applies.

5.4.21 simple_pair_range

The **simple_pair_range** specifies the allowable range of configurations of the two links that join a **kinematic_pair**.

NOTE – The range specification allows for only simple range specifications of each individual pair parameter value in the form “*low bound – high bound*”.

EXPRESS specification:

```
*)  
ENTITY simple_pair_range;  
    applies_to_pair: kinematic_pair;  
END_ENTITY;  
(*
```

Attribute definitions:

applies_to_pair: the pair to which the range applies.

5.4.22 revolute_pair

The **revolute_pair** constrains the motion between two adjacent links to a rotation about a common axis. To measure the angle of rotation a frame is defined on each of the links such that local origins and the z-axes coincide and their positive directions agree. The motion of the second link with respect to the first link is defined as the angle required to rotate the x-axis of the first pair frame in positive direction around the common z-axis until it matches the x-axis of the second pair frame.

NOTE – See figure 4.

EXPRESS specification:

```
*)  
ENTITY revolute_pair
```

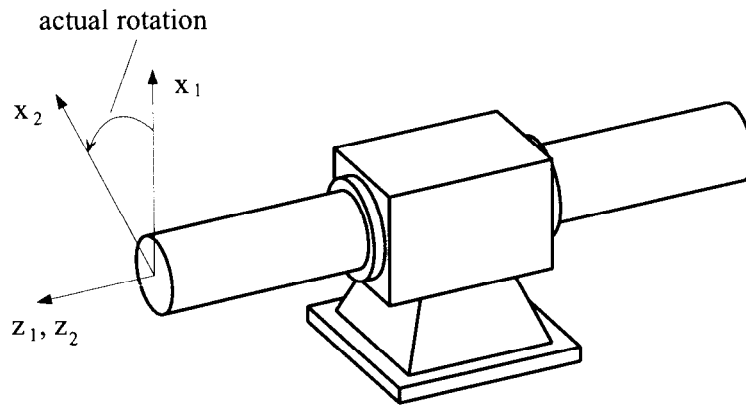


Figure 4 – Example of a joint representing a revolute pair

```
SUBTYPE OF (kinematic_pair);
END_ENTITY;
(*
```

5.4.23 revolute_pair_value

The **revolute_pair_value** specifies the value of the pair parameter for the **revolute_pair**.

EXPRESS specification:

```
*)
ENTITY revolute_pair_value
  SUBTYPE OF (pair_value);
  SELF\pair_value.applies_to_pair : revolute_pair;
  actual_rotation                  : plane_angle_measure ;
END_ENTITY;
(*
```

Attribute definitions:

SELF\pair_value.applies_to_pair: the **revolute_pair** to which the **revolute_pair_value** applies.

actual_rotation: the value of the pair parameter.

5.4.24 revolute_pair_range

The **revolute_pair_range** specifies the lower bound and the upper bound of the parameter range for the **revolute_pair**.

EXPRESS specification:

```
*)
ENTITY revolute_pair_range
  SUBTYPE OF (simple_pair_range);
  SELF\simple_pair_range.applies_to_pair : revolute_pair;
```

```

    lower_limit_actual_rotation      : rotational_range_measure;
    upper_limit_actual_rotation      : rotational_range_measure;
WHERE
    WR1: (('KINEMATIC_STRUCTURE_SCHEMA.UNLIMITED_RANGE' IN
          TYPEOF (lower_limit_actual_rotation))
        OR
        ('KINEMATIC_STRUCTURE_SCHEMA.UNLIMITED_RANGE' IN
          TYPEOF (upper_limit_actual_rotation)))
    XOR
    (lower_limit_actual_rotation < upper_limit_actual_rotation);
END_ENTITY;
(*)

```

Attribute definitions:

SELF\simple_pair_range.applies_to_pair: the **revolute_pair** to which the **revolute_pair_range** applies.

lower_limit_actual_rotation: the minimum value of the pair parameter.

upper_limit_actual_rotation: the maximum value of the pair parameter.

Formal propositions:

WR1: The range of the **revolute_pair** shall be positive when both bounds are not given as unlimited.

5.4.25 prismatic_pair

The **prismatic_pair** constrains the motion between two adjacent links to a translation along a common axis. To measure the distance of translation a frame is defined on each of the links such that their corresponding coordinate axes coincide and their positive directions agree. The motion of the second link with respect to the first link is defined as the distance required to translate the xy-plane of the first pair frame in positive direction of the common z-axis until it coincides with the xy-plane of the second pair frame.

NOTE – See figure 5.

EXPRESS specification:

```

*)
ENTITY prismatic_pair
    SUBTYPE OF ( kinematic_pair);
END_ENTITY;
(*)

```

5.4.26 prismatic_pair_value

The **prismatic_pair_value** is the value of the pair parameter for the **prismatic_pair**.

EXPRESS specification:

```

*)
ENTITY prismatic_pair_value

```