2021-12-08

CTSimU

CTSimU Scenario Descriptions

A data format specification in JSON

Version 0.9

WIPANO CTSimU Project

Durchstrahlungssimulation für die Messunsicherheitsbestimmung beim Messen geometrischer Merkmale mittels Röntgen-Computertomographie

Summary: This is the specification for a JSON file format to handle the parameters of a full industrial CT scan scenario, intended for describing virtual CT scenarios for simulations, as well as for documenting real CT scan geometries and acquisition parameters and their measurement uncertainties.

You can find the **online version** of this document at
https://bamresearch.github.io/ctsimu-scenarios

Contents

About the WIPANO CTSimU Project
Full project name: *Durchstrahlungssimulation für die Messunsicherheitsbestimmung beim Messen geometrischer Merkmale mittels Röntgen-Computertomographie*

CTSimU

Das Projekt CTSimU wurde durch die Förderrichtlinie WIPANO gefördert und durch das Budesministerium für Wirtschaft und Energie finanziert.

WIPANO

Gefördert durch:

Bundesministerium
für Wirtschaft
und Energie

aufgrund eines Beschlusses
des Deutschen Bundestages

1 Introduction

The WIPANO CTSimU project aims to define a qualification framework for radiographic simulation software. Such software is intended to be used for determining measurement uncertainties for geometrical properties of scanned samples, which arise from the overall procedure of a computed tomography (CT) scan.

On the following pages, a JSON structure (JavaScript Object Notation) is introduced. Its purpose is to contain all relevant parameters of a CT scan, with the intention to

1. document the parameters of any real CT set-up with their respective uncertainties, and
2. completely describe CT scenarios for simulations within the project.

The JSON format was chosen because it is both *human-readable* and *human-writeable*, as well as *machine-readable*, which enables reproducible simulations based on the scenario description files.

A listing of a *full example* can be found at the end of this document. The line numbers in the code snippets in the following description refer to the line numbers of the full example. The example describes a step cylinder made of a glass ceramic, placed 20 mm to the “left” of the rotation axis at a slight angle. The rotation stage is symmetrically surrounded by a fixed aluminium frame that does not follow the rotation of the stage (Fig. 1.1). The detector is tilted around its planar normal by 0.023 rad in clockwise direction (as seen from the source).

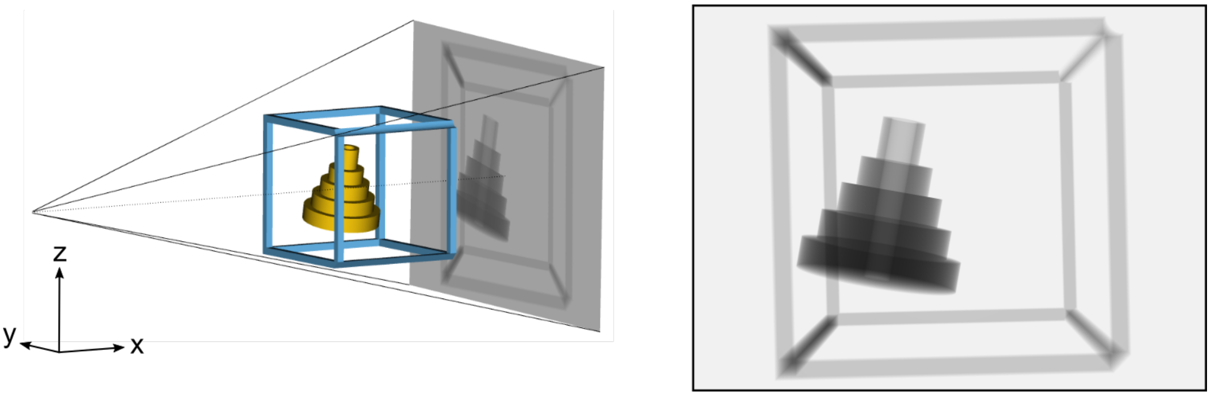


Fig. 1.1: Left: CT setup that is described by the example code snippets in the following sections. **Right:** Resulting projection on the detector.

2 General information

2.1 File structure

The JSON file should be encoded and decoded using the UTF-8 character set, even though all characters in this specification are from the ASCII set to ensure compatibility with other decoding interpretations.

The scenario description consists of the following main sections:

`"file"`: General file information: scenario name, file format version.

`"environment"`: Description of the environment: temperature, atmosphere.

`"geometry"`: Location and orientation of source, detector and sample stage (rotation axis).

`"detector"`: Parameters and characteristics of the detector.

`"source"`: Parameters and characteristics of the X-ray source.

`"samples"`: Definitions of all samples: surface models, location and orientation, size.

`"acquisition"`: Parameters for the CT acquisition: angular steps, flat fields, etc.

`"materials"`: Density and chemical composition of all materials relevant to the scenario.

`"simulation"`: Proprietary parameters specific to the simulation software.

2.2 Parameters

Most values with physical representation come with an associated physical unit and a measurement uncertainty. Additionally, many parameters may drift during a CT scan. In general, parameters are defined using a JSON object which may contain the following properties. Depending on the situation, only the ones of relevance must be specified. Irrelevant ones may be omitted or set to `null` at any level.

```
"parameter": {
  "value": 10.0,
  "unit": "mm",
  "uncertainty": {
```

(continues on next page)

(continued from previous page)

```
    "value": 0.1,
    "unit": "mm"
  }
  "drift": [
    {
      "value": [-100, 100],
      "file":  null,
      "unit":  "mm",
      "known_to_reconstruction": true
    },
    {
      "value": null,
      "file":  "vertical_motion_deviations.csv",
      "unit":  "mm",
      "known_to_reconstruction": false
    }
  ]
}
```

2.3 Values & Units

`"value"`: gives the measured value or the value that should be used by the simulation software.

`"unit"`: gives the physical unit of the value.

The following units are allowed for length, angle, time, voltage, current, density, temperature, angular velocity and the spatial frequency, and should be interpreted correctly by any parser.

"nm"	"deg"	"ms"	"MV"	"uA"	"g/cm^3"	"C"	"deg/s"	"lp/mm"	"px"	"relative"	**null**
"um"	"rad"	"s"	"kV"	"mA"	"kg/m^3"	"K"	"deg/min"	"lp/cm"			
"mm"		"min"	"V"	"A"		"F"	"deg/h"	"lp/dm"			
"cm"		"h"					"rad/s"	"lp/m"			
"dm"							"rad/min"				
"m"							"rad/h"				

The prefix `u` represents the SI prefix μ (10^{-6}). `"relative"` can be used for relative uncertainties or any values that express a fraction of a related measure. For properties without a unit, the keyword `null` is used.

2.4 Uncertainty

`"uncertainty"`: gives a `"value":` for the standard measurement uncertainty and its physical `"unit":`. The intention is to use this to document (or model) a real, physical CT machine.

2.5 Drifts

`"drift":` provides an array that may contain an arbitrary number of drift components. Typically, only one drift component is necessary, but in some cases it can be useful to provide more than one drift component for a parameter, especially if some drift contributions shall be unknown to the reconstruction software, whereas others shall be considered during the reconstruction. A typical example would be a helix scan: the vertical movement of the stage along the rotation axis can be modelled as a drift that must be known during the reconstruction. However, inhomogeneities in the vertical motion can be modelled as a second drift component unknown to the reconstruction software.

Drifts are applied for each frame individually once the stage has reached its intended position as described in the *acquisiton section* of the scenario file. Because any drift value describes an absolute deviation from the initial condition at frame 0, they are not accumulated over time. If multiple drift components are defined, they are applied in an additive, sequential manner in the given order.

Each drift component must provide a range of drift values (at least one). These drift values represent absolute deviations from the initial values at the start position (frame 0). They can be provided in the component's `"value":` array or through a single-column CSV `"file":`. If a drift component provides only a single `"value":` different from `null`, this drift deviation will stay constant throughout the scan. For example, a constant stage drift value for its tilt around one of the stage's plane vectors (see *geometry section*) leads to a simple axis wobble.

For dynamic, non-constant drifts, more than one drift value can be provided in the `"value":` array or CSV `"file":`. Ideally, the number of rows in the CSV file would match the number of frames of the scan. If the number of provided drift values does not match the number of frames, the values are assumed to be spread in equidistant steps between start projection (first value) and last projection (last value), and a linear interpolation between neighbouring values is assumed to calculate each frame's deviation value for the parameter.

The physical `"unit":` of the deviation values should be specified; otherwise, the main parameter unit is assumed. If the drift refers to a parameter that expects a string (e.g. file name of a spectrum file), the `"values":` array or CSV `"file":` should contain a string for each frame; otherwise, the same equidistant behaviour is assumed as for numerical parameters with the exception that no interpolation takes place. Instead, a string remains valid until the next key frame is reached.

The parameter `"known_to_reconstruction":` (either `true` or `false`) can be used to specify whether the drift should be considered during the reconstruction of the CT scan (e.g. when calculating projection matrices).

Drift keywords are prepared throughout this document for any parameters where they are assumed to be possible. In most cases of the example, they are set `null`, rendering them inactive.

2.6 Formats of referred data files

A scenario description may refer to other data files.

For **one-dimensional** data such as response curves, characteristics files or spectra, the CSV format shall be used, with its columns in the order specified in the corresponding sections of this guide. The columns shall be separated by commas or white-space.

For **two-dimensional** data such as intensity profiles or bad pixel maps, image files shall be used. The minimum set of supported image file formats should be TIFF and headerless RAW. For the RAW format, please follow the details for three-dimensional data.

For **three-dimensional** data, such as a 3D spot intensity profile, a headerless RAW file shall be used. Its dimensions are specified at the respective place in the JSON scenario. The data type can be one of the following: `uint8`, `int8`, `uint16`, `int16`, `float` (32 bit). Data shall be written row-first, column-second, slice-third. For an image with n_x columns, n_y rows and n_z slices, this results in an array with the following one-dimensional index representation, with coordinates (x, y, z) starting at $(0, 0, 0)$:

$$\mathrm{Index}(x, y, z) = (n_x \cdot n_y \cdot z) + (n_x \cdot y) + x. \tag{2.1}$$

3D raw data is expected to be written in little-endian byte order, and most-significant to least-significant bit (MSB_0).

3 File

The `"file":` section leaves room for specifying a scenario `"name":` and a short `"description":`. Further meta data such as a `"contact":` person, dates of creation and modification, and a file `"version":` can be added here.

The `"file_type":` is a constant string to identify this as a CTSimU scenario and must not be changed. The `"file_format_version":` states which issue of the file format specification is used.

```
"file":
{
  "name": "Example Scenario",
  "description": "Step cylinder in a rigid frame.",

  "contact": "David Plotzki",
  "date_created": "2020-04-23",
  "date_changed": "2020-05-15",
  "version": {"major": 1, "minor": 4},

  "file_type": "CTSimU Scenario",
  "file_format_version": {"major": 0, "minor": 9}
}
```

4 Environment

In the `"environment":` section, the composition of the atmosphere can be described, as well as the environment temperature.

```
"environment":
{
  "material_id": "Air",
  "temperature": {
    "value": 20, "unit": "C",
    "uncertainty": {"value": 0.5, "unit": "C"},
    "drift": [
      {"file": "temperate_drift.csv", "unit": "C"}
    ]
  }
}
```

5 Geometry

5.1 Placement of objects in the world coordinate system

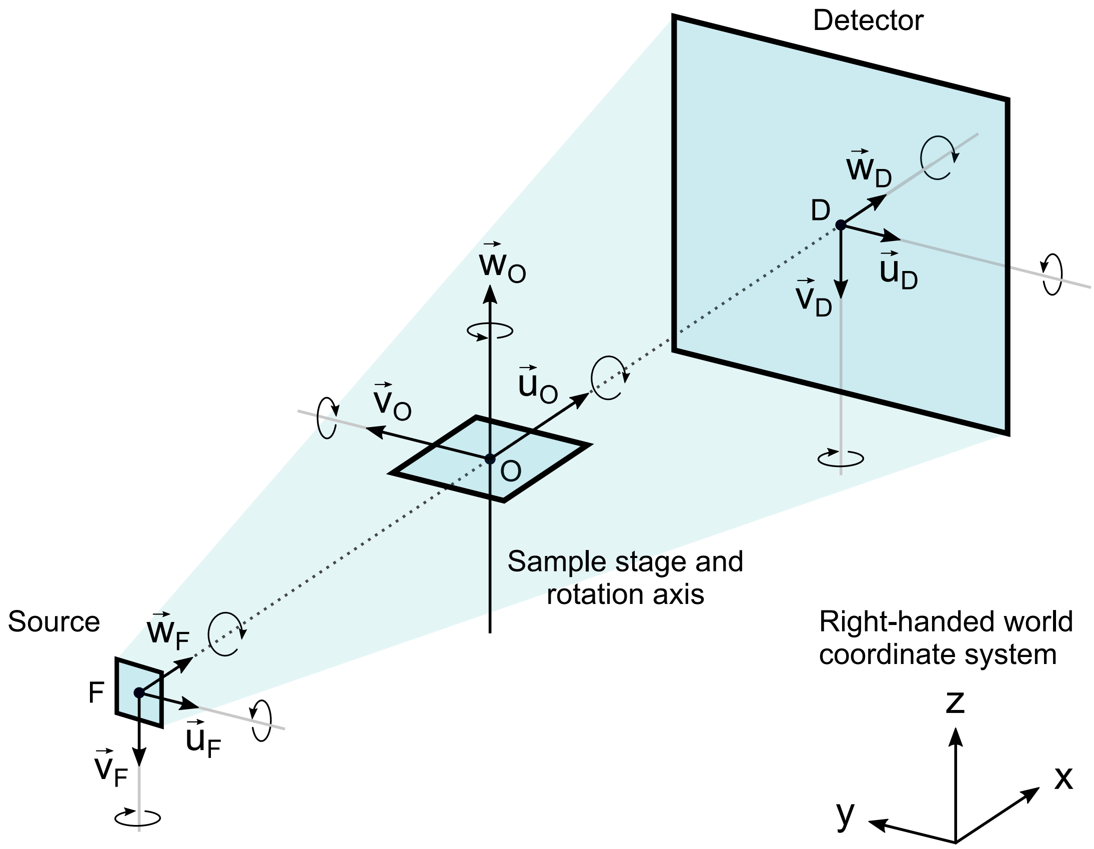


Fig. 5.1: World coordinate system {x, y, z} and local coordinate systems {u, v, w}.

This specification does not define a fixed coordinate system for the CT set-up. The only assumption is that a right-handed world coordinate system {x, y, z} is used, and the local coordinate systems {u, v, w} of source, detector and sample stage are to be placed in this world coordinate system. Fig. 5.1 illustrates the set-up that is used in the following examples.

The `"geometry":` section has three subsections to define the location and orientation of the principal CT scanner components: `"detector":`, `"source":` and `"stage":`. They all share a common set of JSON properties for positioning, and a very similar description is used later on to place samples into the scene (see *sample positioning*). The placement description generally consists of the following parts:

`"centre":` specifies the object's centre (x, y, z) in the world coordinate system, i.e. the geometric centre of its bounding box. The bounding box is defined as the smallest cuboid (or rectangle, for 2D objects) that completely encloses the object, under the condition that the cuboid's edges are each strictly parallel to one of the coordinate axes of the object's own Cartesian coordinate system. The `"centre":` is the origin of the local coordinate system {u, v, w} and also the pivot point for rotations.

`"vector_u":` and `"vector_w":` specify the object's orientation by defining the basis vectors $\vec{u}$ and $\vec{w}$ of the local coordinate system in terms of the world coordinate system {x, y, z}. They are not required to be unit vectors, but they must be **orthogonal.**

$\vec{w}$ is usually meant to be a normal vector, and $\vec{u}$ is one of the support vectors. $\vec{u}$ and $\vec{v}$ also serve a second meaning as row and column vector of the resulting projection (for the detector) or of a given spot intensity profile image (for the source). See the descriptions below for details.

`"rotation":` specifies small rotational deviations from the ideal geometry, such as tilts of the detector or the rotation axis. A simulation software should treat them as subsequent transformations of the initial local coordinate system. Whether these deviations are known to the reconstruction software or not depends on the purpose of the scenario and can be specified by setting the property `"known_to_reconstruction":` to either `true` or `false`:

```
"known_to_reconstruction": true
```

`"u":`, `"v":` and `"w":` specify small rotation angles around the axes of the **local** coordinate system. The convention here sticks to $wv'u''$ for subsequent rotations, which means that the rotation around the $\vec{w}$ vector is performed first, then a rotation around the new $\vec{v}'$ vector, and finally the rotation around the resulting $\vec{u}''$ vector. In the case of samples, their order of rotation takes place in the same manner: $ts'r''$ (see *sample positioning*).

Note that the angular deviations specified here always describe a **static** deviation of the initial coordinate system (at frame 0). They are not applied individually at each frame, which means that a stage tilt described here is meant to stay constant throughout the scan and does not describe a wobble. For dynamic deviations, the *drift property* of parameters can be used, as drifts are applied for each frame individually after the stage has reached its intended frame position.

5.2 Detector

For this specification, in general, vectors $\vec{u}$ and $\vec{v}$ designate an item's in-plane vectors, whereas $\vec{w}$ usually designates a normal vector. Following this convention, $\vec{u}_\mathrm{D}$ is the detector's **row vector,** pointing from left to right in the resulting projection image (as seen on a computer screen with a pixel coordinate system that has its origin in the upper left corner). $\vec{v}_\mathrm{D}$ is the detector's **column vector,** pointing from top to bottom in the resulting projection image. The orientation of these two vectors directly determines the orientation of the projection image.

Note that for any object only the normal vector $\vec{w}$ and the support vector $\vec{u}$ are given in the JSON file. The detector normal $\vec{w}_\mathrm{D}$ does not have any special meaning and should be arranged such that the detector's row and column vector point in the desired directions.

The size and further properties of the detector are defined later on in the *detector section* of the JSON file.

```
"detector":
{
  "centre": {
    "x": {"value": 400, "unit": "mm", "uncertainty": {"value": 0, "unit": "mm"}, "drift": null},
    "y": {"value":   0, "unit": "mm", "uncertainty": {"value": 0, "unit": "mm"}, "drift": null},
    "z": {"value":   0, "unit": "mm", "uncertainty": {"value": 0, "unit": "mm"}, "drift": null}
  },

  "vector_u": {
    "x": {"value":  0, "drift": null},
    "y": {"value": -1, "drift": null},
    "z": {"value":  0, "drift": null}
  },
  "vector_w": {
    "x": {"value":  1, "drift": null},
    "y": {"value":  0, "drift": null},
    "z": {"value":  0, "drift": null}
  },

  "rotation": {
    "u": {"value": 0,      "unit": "rad", "uncertainty": {"value": 0, "unit": "rad"}, "drift": null},
    "v": {"value": 0,      "unit": "rad", "uncertainty": {"value": 0, "unit": "rad"}, "drift": null},
    "w": {"value": 2.3e-2, "unit": "rad", "uncertainty": {"value": 0, "unit": "rad"}, "drift": null},
    "known_to_reconstruction": true
  }
}
```

5.3 Source

The source can be modelled either as a cone-beam geometry or a parallel beam geometry. This behaviour is set by the `"type":` property, which can be either `"cone"` or `"parallel"`.

```
"type": "cone" "parallel"
```

In the case of a parallel beam, its divergence can be specified along both planar axes of the source:

```
"beam_divergence": {
        "u": {"value": 0, "unit": "rad", "uncertainty": {"value": 0, "unit": "rad"}, "drift": null},
        "v": {"value": 0, "unit": "rad", "uncertainty": {"value": 0, "unit": "rad"}, "drift": null}
}
```

For a cone beam geometry, this property should be set to `null`:

```
"beam_divergence": null
```

Spatially extended source intensity profiles are modelled as a rectangle. For cone-beam geometries, this will be a very small rectangle on the size scale of the spot size.

For parallel beam geometries, the optical axis of the system is assumed to be the $\vec{w}_{\mathrm{F}}$ axis of the source, and all rays should be parallel to this axis (apart from beam divergence). This means that the source rectangle in the model should ideally be of a size such that the entire detector is covered by radiation.

If a spot intensity profile is given as an image (see *source* for details), $\vec{u}_{\mathrm{F}}$ is the image's row vector, pointing from left to right in this image, and $\vec{v}_{\mathrm{F}}$ is the image's column vector, pointing from top to bottom.

To allow the correct orientation of the intensity profile image, the vector $\vec{w}_{\mathrm{F}}$ does not necessarily point in the main direction of radiation, but could also point in the opposite direction (due to the restraint of a right-handed coordinate system).

Within the WIPANO CTSimU project, we agreed to the convention of placing the source at (0, 0, 0). It is not a requirement.

```
"source":
{
  "type": "cone",
  "beam_divergence": {
    "u": {"value": 0, "unit": "rad", "uncertainty": {"value": 0, "unit": "rad"}, "drift": null},
    "v": {"value": 0, "unit": "rad", "uncertainty": {"value": 0, "unit": "rad"}, "drift": null}
  },

  "centre": {
    "x": {"value": 0, "unit": "mm", "uncertainty": {"value": 0, "unit": "mm"}, "drift": null},
    "y": {"value": 0, "unit": "mm", "uncertainty": {"value": 0, "unit": "mm"}, "drift": null},
    "z": {"value": 0, "unit": "mm", "uncertainty": {"value": 0, "unit": "mm"}, "drift": null}
```

(continues on next page)

(continued from previous page)

```
  },

  "vector_u": {
    "x": {"value":  0, "drift": null},
    "y": {"value": -1, "drift": null},
    "z": {"value":  0, "drift": null}
  },
  "vector_w": {
    "x": {"value":  1, "drift": null},
    "y": {"value":  0, "drift": null},
    "z": {"value":  0, "drift": null}
  },

  "rotation": {
    "u": {"value": 0, "unit": "rad", "uncertainty": {"value": 0, "unit": "rad"}, "drift": null},
    "v": {"value": 0, "unit": "rad", "uncertainty": {"value": 0, "unit": "rad"}, "drift": null},
    "w": {"value": 0, "unit": "rad", "uncertainty": {"value": 0, "unit": "rad"}, "drift": null},
    "known_to_reconstruction": true
  }
}
```

5.4 Stage

The normal vector $\vec{w}_O$ of the sample stage specifies the axis of rotation for the CT scan. By default, the sample stage coordinate system and all samples attached to it are meant to rotate around this axis, whereas the source and detector stay still (as for typical industrial CT scanners).

Samples that are placed in the stage coordinate system all take part in the rotation of the sample stage. Samples that are placed in the world coordinate system are fixed during the CT scan (i.e. fixed relative to source and detector, see *samples* for details).

```
"stage":
{
  "centre": {
    "x": {"value": 275, "unit": "mm", "uncertainty": {"value": 0, "unit": "mm"}, "drift": null},
    "y": {"value":   0, "unit": "mm", "uncertainty": {"value": 0, "unit": "mm"}, "drift": null},
    "z": {"value":   0, "unit": "mm", "uncertainty": {"value": 0, "unit": "mm"}, "drift": null}
  },

  "vector_u": {
    "x": {"value":  1, "drift": null},
    "y": {"value":  0, "drift": null},
    "z": {"value":  0, "drift": null}
```

(continues on next page)

(continued from previous page)

```
  },
  "vector_w": {
    "x": {"value":  0, "drift": null},
    "y": {"value":  0, "drift": null},
    "z": {"value":  1, "drift": null}
  },

  "rotation": {
    "u": {"value":  0, "unit": "rad", "uncertainty": {"value": 0, "unit": "rad"}, "drift": null},
    "v": {"value":  0, "unit": "rad", "uncertainty": {"value": 0, "unit": "rad"}, "drift": null},
    "w": {"value":  0, "unit": "rad", "uncertainty": {"value": 0, "unit": "rad"}, "drift": null},
    "known_to_reconstruction": true
  }
}
```

6 Detector

6.1 General properties

The following properties can be specified in the detector section of the scenario file.

The model name and the manufacturer, if an existing detector is modelled:

```
"model":        "",
"manufacturer": ""
```

Using the `"type":` keyword, it can be defined whether an **ideal** or a **real** detector is described or shall be simulated. An **ideal** detector is assumed to convert all incident radiation energy linearly into grey values. For an **ideal** detector, no scintillator should be defined in the scenario file. A **real** detector is assumed to take the absorption characteristics and quantum yield of the scintillator material into account.

```
"type": "real" "ideal"
```

The number of detector pixels in directions $\vec{u}_D$ and $\vec{v}_D$:

```
"columns": {"value": 2000, "unit": "px"},
"rows":    {"value": 1500, "unit": "px"}
```

The pixel pitch (distance between neighbouring pixels) in directions $\vec{u}_D$ and $\vec{v}_D$. The physical dimensions of the active detector area are calculated from the pixel pitch and the number of pixels per row and column.

```
"pixel_pitch": {
  "u": {"value": 0.127, "unit": "mm", "uncertainty": {"value": 0, "unit": "mm"}, "drift": null},
```

(continues on next page)

(continued from previous page)

```
  "v": {"value": 0.127, "unit": "mm", "uncertainty": {"value": 0, "unit": "mm"}, "drift": null}
}
```

The bit depth of the detector:

```
"bit_depth": {"value": 16}
```

The integration time is the exposure time to take one frame. The dead time is the time between exposure times, which includes the readout time of the detector.

```
"integration_time": {"value": 0.5, "unit": "s", "uncertainty": {"value": 0, "unit": "s"}, "drift": null},
"dead_time":        {"value": 50, "unit": "ms", "uncertainty": {"value": 0, "unit": "s"}, "drift": null}
```

The `"image_lag":` between subsequent frames can be specified, following loosely the definition of *GlobalLag1f* in ASTM E2597. [1]

$$\text{image lag} = \frac{\text{Grey value at first frame radiation fully off}}{\text{Grey value at radiation fully on}} \tag{6.1}$$

```
"image_lag": {"value": 0.05, "unit": null, "uncertainty": {"value": 0, "unit": null}, "drift": null}
```

As opposed to *GlobalLag1f*, `"image_lag":` refers to the scenario described in this file, particularly the specified radiation intensity and integration time. Here, it is therefore not treated as a parameter intrinsic to the detector, but describes its lag characteristics only for the specified scenario.

6.2 Grey value characteristics

There are three ways provided to model the grey value characteristics of the detector.

```
"grey_value":
{
  "imax":   {"value": 45000, "drift": null},
  "imin":   {"value":  1000, "drift": null},

  "factor": {"value": 3e15, "unit": "1/J", "uncertainty": {"value": 0, "unit": "1/J"}, "drift": null},
  "offset": {"value": 1800, "unit": null,  "uncertainty": {"value": 0, "unit": null}, "drift": null},

  "intensity_characteristics_file": {"value": "detector_intensity.csv", "drift": null},

  "efficiency":   {"value": 0.7, "unit": null, "uncertainty": {"value": 0, "unit": null}, "drift": null},
  "efficiency_characteristics_file": {"value": "detector_efficiency.csv", "drift": null}
}
```

Min/Max method

The first approach is to specify the average grey value at the maximum intensity of the free beam in the parameter `"imax":`, and the average grey value at no incident radiation in the parameter `"imin":`. Between these two values, a linear interpolation based on the incident intensity should take place to calculate the grey value of a pixel. In this case, the grey values are invariant under changes in the integration time or pixel size, and will be rescaled along with changes in the minimum or maximum intensity. It is similar to the re-normalization of the projection image to a specific grey value at maximum free-beam intensity.

Linear response function

The second method allows to specify a linear response function. This function assigns a grey value to the collected energy E (in J) for each pixel.

$$\text{Grey Value} = (\text{factor} \cdot E) + \text{offset} \tag{6.2}$$

The file format allows to provide a `"factor":` and an `"offset":` for this linear function. This method allows the grey values to change with a change in pixel size, tube power, integration time or a change in intensity due to a different focus-detector distance. It has precedence over the first method if `"factor":` and `"offset":` are set and not `"null":`. From eq. (6.2) it becomes clear that the `"factor":` should have the inverse unit of the deposited energy: 1/J.

External characteristics file

For a more general approach, it is also possible to provide an arbitrary grey value characteristics from a CSV file specified by the parameter `"intensity_characteristics_file":`. The file should contain the columns listed in Tab. 6.1, separated by commas or white space.

The interpolation method between these discrete values shall not be specified here and is left to the simulation software. However, all values are assumed to refer to the centre of their bin.

If a valid intensity characteristics CSV file is specified and the parameter is not set to `"null":`, this method has precedence over the first two methods.

Tab. 6.1: CSV for detector's intensity characteristics

Col.	Property
1	Energy E in J, collected by a pixel
2	Grey value
3	Grey value uncertainty *(optional)*

Quantum efficiency

The photon **conversion efficiency** (quantum efficiency) of the detector can also either be provided as a constant value for all photon energies – using the parameter `"efficiency":` – or as another characteristics curve through a CSV file, using the parameter `"efficiency_characteristics_file":`. In the latter case, this file should contain the columns listed in Tab. 6.2, separated by commas or white space. If a valid efficiency characteristics file is provided, it has precedence over the global `"efficiency":` property.

Tab. 6.2: CSV for detector's efficiency characteristics

Col.	Property
1	Photon energy in keV
2	Quantum efficiency, as the ratio of incident photons to absorbed photons
3	Quantum efficiency uncertainty *(optional)*

6.3 Image Quality

Noise

The noise in the projection image can be described by specifying the signal-to-noise ratio (SNR) at the maximum intensity (free beam) for one frame, with no frame averaging applied.

```
"noise":
{
  "snr_at_imax": {"value": 205, "unit": null, "uncertainty": {"value": 0, "unit": null}, "drift": null},
  "noise_characteristics_file": {"value": "noise.csv", "drift": null}
}
```

Note that the SNR's underlying grey value distribution must refer to an area of pixels that ideally has the same mean intensity, and is not subject to systematic grey value variations from other physical effects such as the $1/r^2$ law or a change in the angle of radiation incidence on the detector.

It is also possible to provide an intensity-dependent noise characteristics using a CSV file. A valid characteristics file has precedence over the SNR at maximum intensity. The characteristics file should contain the columns listed in Tab. 6.3, separated by commas or white space.

Tab. 6.3: CSV for noise characteristics

Col.	Property
1	Mean pixel grey value (intensity)
2	Signal-to-noise ratio (SNR)
3	SNR uncertainty *(optional)*

Gain

A gain factor can be specified using the `"gain":` parameter:

```
"gain":
{
  "value": 3,
  "drift": null,
  "scale_signal_and_noise": true
}
```

If the parameter `"scale_signal_and_noise":` is set to `true`, a simulation software shall use this parameter to scale the grey values (usually linearly), and noise (usually quadratically) with the gain value. This assumes that the grey value characteristics and noise characteristics that have been provided before refer to a gain factor of 1. If the grey value and noise characteristics are already provided for the given gain factor, the parameter must be set to `false`.

Unsharpness

There are three ways provided to specify the detector unsharpness.

```
"unsharpness":
{
  "basic_spatial_resolution": {"value": 0.1, "unit": "mm", "uncertainty": {"value": 0, "unit":
↪"mm"},         "drift": null},
  "mtf10_frequency": {"value": 10, "unit": "lp/mm", "uncertainty": {"value": 1, "unit": "lp/mm
↪"},             "drift": null},
  "mtf": {"value": "detector_mtf.csv", "drift": null}
}
```

1. The basic spatial resolution, as defined in ASTM E2597 [1], provided using the parameter `"basic_spatial_resolution":`.
2. The MTF10 frequency of the system's modulation transfer function (MTF). This value states the frequency at which the contrast in the detector response drops to 10% [2]. It can be provided using the parameter `"mtf10_frequency":`. If specified and not `null`, this parameter has precedence over the basic spatial resolution.
3. The complete modulation transfer function (MTF, [2]), provided through a CSV file. Its name is given using the parameter `"mtf":`, and it should contain the columns listed in Tab. 6.4, separated by commas or white space. If a valid MTF is provided, this has precedence over the first two parameters mentioned here.

Tab. 6.4: CSV for noise characteristics

Col.	Property
1	Frequency in lp/mm
2	Modulation contrast, from the interval [0, 1]
3	Modulation contrast uncertainty *(optional)*

Bad pixel map

The bad pixel map provided here should be a 2D grey-scale image file with a number of columns and rows that matches the detector. A pixel value of `0` means that the pixel is working properly. A pixel value other than `0` means that the pixel does not function correctly. If a TIFF file is provided, the `"type":` property is obsolete and can be omitted.

```
"bad_pixel_map":
{
  "value": "badpixels.raw",
  "type": "uint8",
  "drift": null
}
```

6.4 Scintillator & Filters

In this section, the scintillator and filter materials for the detector can be defined. The filter materials are split into `"front":` filters and `"rear":` filters (i.e. a back panel, mostly to consider backscattering). Any number of filters can be defined. For the materials of scintillator and filters, only a material ID is given here. The actual material definition and declaration of its chemical composition is found in the *materials section* of the JSON file.

Note that front and rear of the detector are not explicitly identified by the detector's normal vector or any other property, but only given implicitly by the side of the detector facing the source, and the side facing away from the source.

```
"scintillator":
{
  "material_id": "CsI",
  "thickness": {"value": 0.15, "unit": "mm", "uncertainty": {"value": 0, "unit": "mm"}, "drift": null}
},
"filters":
{
  "front":
  [
    {
      "material_id": "Al",
      "thickness": {"value": 0.2, "unit": "mm", "uncertainty": {"value": 0, "unit": "mm"}, "drift": null}
    },
    {
      "material_id": "Kapton",
      "thickness": {"value":0.13, "unit": "mm", "uncertainty": {"value": 0, "unit": "mm"}, "drift": null}
    }
  ],
  "rear":
  [
    {
```

(continues on next page)

(continued from previous page)

```
      "material_id": "Al",
      "thickness": {"value": 2.0, "unit": "mm", "uncertainty": {"value": 0, "unit": "mm"}, "drift": null}
    }
  ]
}
```

7 Source

7.1 General properties

The following properties can be specified in the source section of the scenario file.

The model name and the manufacturer, if an existing X-ray source is modelled:

```
"model":        "",
"manufacturer": ""
```

The tube acceleration voltage and the target current:

```
"voltage": {"value": 130, "unit": "kV", "uncertainty": {"value": 0, "unit": "kV"}, "drift": null},
"current": {"value": 100, "unit": "uA", "uncertainty": {"value": 0, "unit": "uA"}, "drift": null}
```

7.2 Target

The target material is defined by providing a `"material_id":` that refers to a material declaration in the *materials section* of the file.

```
"material_id": "W"
```

The `"type":` can be either a transmission target or a reflection target:

```
"type": "transmission" "reflection"
```

For a transmission target, its thickness can be specified:

```
"thickness": {"value": 3.0, "unit": "um", "uncertainty": {"value": 0, "unit": "um"}, "drift": null}
```

For a reflection target, this parameter should be set to `null`:

```
"thickness": null
```

The angles of electron incidence and main X-ray emission can be defined and refer to the angles between the target surface and electron beam or main photon emission direction, respectively.

```
"angle":
{
  "incidence": {"value": 45, "unit": "deg", "uncertainty": {"value": 0, "unit": "deg"}, "drift": null},
  "emission":  {"value": 45, "unit": "deg", "uncertainty": {"value": 0, "unit": "deg"}, "drift": null}
}
```

7.3 Spot intensity profile

The spot intensity profile is specified in the source's `"spot":` section. At first, the size of the virtual spot rectangle or volume is defined along the three axes of the source coordinate system. If only a two-dimensional spot profile is modelled, the size along the source's normal axis should be set to `0`.

```
"size": {
  "u": {"value": 100.0, "unit": "um", "uncertainty": {"value": 0, "unit": "um"}, "drift": null},
  "v": {"value": 100.0, "unit": "um", "uncertainty": {"value": 0, "unit": "um"}, "drift": null},
  "w": {"value":   0.0, "unit": "um", "uncertainty": {"value": 0, "unit": "um"}, "drift": null}
}
```

If the spot size is set to `null`, the simulation software is free to choose a size that matches the required (Gaussian) shape.

```
"size": null
```

The shape of the spot can be defined in the following three ways.

Simple Gaussian profiles

A simple Gaussian profile can be modelled $I(\vec{r}) = I_0 \cdot \exp(-|\vec{r}|^2/2\sigma^2)$, by specifying the spatial sigmas σ for each dimension:

```
"sigma": {
  "u": {"value":  50.0, "unit": "um", "uncertainty": {"value": 0, "unit": "um"}, "drift": null},
  "v": {"value":  50.0, "unit": "um", "uncertainty": {"value": 0, "unit": "um"}, "drift": null},
  "w": {"value":   0.0, "unit": "um", "uncertainty": {"value": 0, "unit": "um"}, "drift": null}
}
```

2D images

For a more detailed approach, the intensity profile can also be provided from an external image file. In this case, the $\vec{u}$ vector of the source coordinate system points from left to right in the image, and the $\vec{v}$ vector points from top to bottom, in analogy to the *detector geometry*. The image shall be resized to match the `"size":` of the spot rectangle, without necessarily retaining the original aspect ratio of the image. The picture is recommended to be a 32 bit float grey-scale image, and the pixel values in the interval [0, 1], with 0 meaning no intensity, and 1 meaning full intensity. However, a simulation software must also support grey-scale integer file formats, and be able to re-normalize the provided grey values accordingly. If a valid spot intensity image is provided, this method takes precedence over the previously described specification of Gaussian sigmas.

If a RAW file is provided, the number of pixels in directions x (i.e. $\vec{u}$ in the source coordinate system) and y (i.e. $\vec{v}$) must be specified, as well as its type (see *formats of referred data files* for details). For TIFF images, the parameters `"type":`, `"dim_x":` and `"dim_y":` are obsolete and can be omitted.

```
"intensity_map": {
  "file": "spot_intensities.raw",
  "type": "float",
  "dim_x": 100,
  "dim_y": 100,
  "drift": null
}
```

3D volumes

To describe a three-dimensional spot profile, a RAW file can be provided. It is recommended to be a 32 bit float volume with values between `0` and `1`, and should otherwise be re-normalized by the simulation software. The lowest value (0) means no intensity, the highest value (1) means maximum intensity. If specified, this method takes precedence over the first two described methods.

The x direction of the given volume points along the $\vec{u}$ vector of the source coordinate system, y points in direction $\vec{v}$, and z in direction $\vec{w}$. The data type and volume dimensions need to be specified as well (see section *formats of referred data files* for details). The volume shall be resized to match the `"size":` of the spot volume, without necessarily retaining the original aspect ratio of the volume file.

```
"intensity_map": {
  "file": "spot_intensities.raw",
  "type": "float",
  "dim_x": 100,
  "dim_y": 100,
  "dim_z": 100,
  "drift": null
}
```

7.4 Spectrum

If the spectrum is to be calculated by the simulation software, the following three parameters decide whether only a monochromatic energy scenario is described, or a complete spectrum, with the ability to artificially turn on and off the components from Bremsstrahlung and characteristic emission from the target material. If `"bremsstrahlung":` or `"characteristic":` (or both) is set to `true`, these take precedence over the monochromatic case.

```
"spectrum":
{
  "monochromatic": false,
  "bremsstrahlung": true,
  "characteristic": true,
  "file": {"value": "tube_spectrum.csv", "drift": null}
}
```

A spectrum can also be provided through a CSV file. This spectrum is assumed to be already filtered by the tube's window material, but not yet by any of the filters in front of the tube (as specified under *Tube window and filters*). The CSV file should contain the columns listed in Tab. 7.1, separated by commas or white-space.

Tab. 7.1: CSV format for an X-ray spectrum

Col.	Property
1	Photon energy in keV
2	Number of photons in 1 / (s · sr · mA)
3	Uncertainty in the number of photons *(optional)*

The energy values correspond to the centre of the histogram bins. The interpolation between the bin values shall not be specified here and is left to the simulation software. If a valid spectrum file is specified and the `"file":` parameter is not set to `null`, this has precedence over the spectrum calculated by the simulation software.

7.5 Tube window and filters

The `"window":` material(s) and the additional `"filters":` in front of the tube are specified in two separate JSON arrays. For both, an arbitrary number of materials and thicknesses can be specified. If a *source spectrum* is provided in a file, the spectrum is assumed to be already filtered by all `"window":` materials, but not yet by any `"filters":`.

The `"material_id":` refers to the *material definition* in the `"materials":` section of the file. Also, a window and filter `"thickness":` must be provided.

```
"window":
[
  {
    "material_id": "Al",
    "thickness": {"value": 4.0, "unit": "mm", "uncertainty": {"value": 0, "unit": "mm"}, "drift": null}
  }
],
```

(continues on next page)

(continued from previous page)

```
"filters":
[
  {
    "material_id": "Brass",
    "thickness": {"value": 0.5, "unit": "mm", "uncertainty": {"value": 0, "unit": "mm"}, "drift": null}
  },
  {
    "material_id": "Cu",
    "thickness": {"value": 0.25, "unit": "mm", "uncertainty": {"value": 0, "unit": "mm"}, "drift": null}
  }
]
```

8 Samples

8.1 General properties

Any number of samples can be defined in the `"samples":` array and placed in the scene, either *attached to the rotating sample stage*, or *fixed to the world coordinate system*. A sample object has the following properties.

The sample name:

```
"name": "Step Cylinder"
```

A reference to the model file (e.g. STL or CAD file) that describes the sample geometry:

```
"file": {"value": "stepCyl.stl", "drift": null}
```

The unit of length that is used in the model file:

```
"unit": "mm"
```

A scaling factor for each axis of the sample coordinate system, if the model should be resized by a constant factor:

```
"scaling_factor": {
  "r": {"value": 1.0, "drift": null},
  "s": {"value": 1.0, "drift": null},
  "t": {"value": 1.0, "drift": null}
}
```

The material of the sample, given by a `"material_id":` that references a material definition in the material section of the file:

```
"material_id": "Glass Ceramic"
```

8.2 Sample positioning

Sample attached to the stage coordinate system

The sample coordinate system {r, s, t} is equivalent to the surface model's proper {x, y, z} coordinate system (except for the location of the origin). It is specific to each sample and can be placed in the stage coordinate system by providing its centre coordinates in terms of $\{u, v, w\}_O$ (see Fig. 8.1 for an illustration). If done so, the sample is attached to the stage and will follow any rotations and translations performed by the stage during the CT scan.

The description follows the convention that has been established for the *placement of objects in the world coordinate system*, with the following sub-elements of the sample's `"position":` property.

```
"centre": {
  "u": {"value":  0, "unit": "mm", "uncertainty": {"value": 0, "unit": "mm"}, "drift": null},
  "v": {"value": 20, "unit": "mm", "uncertainty": {"value": 0, "unit": "mm"}, "drift": null},
  "w": {"value":  0, "unit": "mm", "uncertainty": {"value": 0, "unit": "mm"}, "drift": null}
}
```

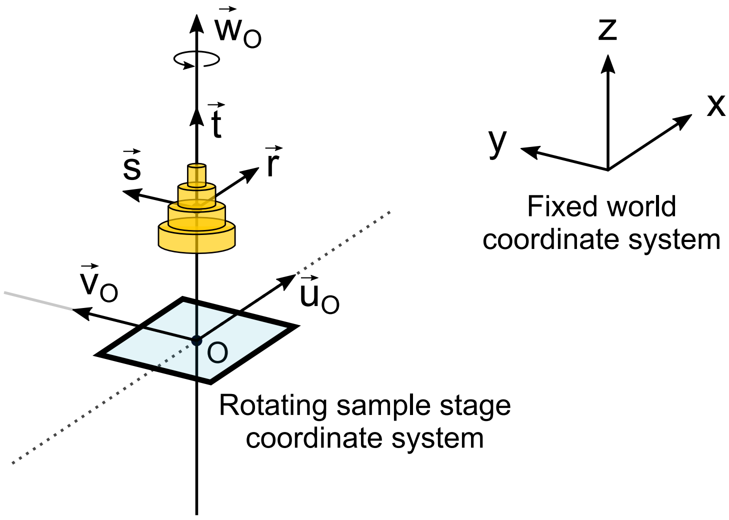


Fig. 8.1: World coordinate system {x, y, z}, stage coordinate system $\{u, v, w\}_O$ and sample coordinate system {r, s, t}.

To define the sample's orientation, its $\vec{r}$ and $\vec{t}$ vector must also be expressed in terms of the stage coordinate system $\{u, v, w\}_O$:

```
"vector_r": {
  "u": {"value":  1,   "drift": null},
  "v": {"value":  0,   "drift": null},
  "w": {"value":  0,   "drift": null}
},
"vector_t": {
  "u": {"value":  0,   "drift": null},
  "v": {"value": -0.2, "drift": null},
  "w": {"value":  1,   "drift": null}
}
```

In analogy to the rotational deviations of the source, stage and detector (see *geometry*), the sample's rotational deviations should also be expressed in terms of the **sample coordinate system.** Rotations are meant to be applied in the following manner: $ts'r''$ which means that the rotation around the sample's $\vec{t}$ axis is performed first, followed by the rotation around the new $\vec{s}'$ axis, and finally the rotation around the resulting $\vec{r}''$ axis.

```
"rotation": {
  "r": {"value": 0, "unit": "rad", "uncertainty": {"value": 0, "unit": "rad"}, "drift": null},
  "s": {"value": 0, "unit": "rad", "uncertainty": {"value": 0, "unit": "rad"}, "drift": null},
  "t": {"value": 0, "unit": "rad", "uncertainty": {"value": 0, "unit": "rad"}, "drift": null}
}
```

Fixed sample position in the world coordinate system

If the sample is placed in the fixed world coordinate system, it will not follow any motions performed by the sample stage, but it will stay fixed relative to source and detector.

The description is very similar to placing a sample in the stage coordinate system, as described in the *previous section*. The only difference is that the object's centre and basis vectors $\vec{r}$ and $\vec{t}$ are now expressed in terms of the world coordinate system {x, y, z}, just like it is done for the source and detector. The rotational deviations are still described in the sample coordinate system {r, s, t}. The following listing gives an example of an aluminium frame around the sample stage that is fixed to the world coordinate system.

```
"name": "Attachment Frame",
"file": {"value": "frame.stl", "drift": null},
"unit": "mm",
"scaling_factor": {
  "r": {"value": 1.0, "drift": null},
  "s": {"value": 1.0, "drift": null},
  "t": {"value": 1.0, "drift": null}
},
"material_id": "Al",
"position":
{
  "centre": {
    "x": {"value": 275, "unit": "mm", "uncertainty": {"value": 0, "unit": "mm"}, "drift": null}
↪,
    "y": {"value":   0, "unit": "mm", "uncertainty": {"value": 0, "unit": "mm"}, "drift": null}
↪,
    "z": {"value":   0, "unit": "mm", "uncertainty": {"value": 0, "unit": "mm"}, "drift": null}
  },

  "vector_r": {
    "x": {"value":  1, "drift": null},
    "y": {"value":  0, "drift": null},
    "z": {"value":  0, "drift": null}
  },
  "vector_t": {
    "x": {"value":  0, "drift": null},
```

(continues on next page)

(continued from previous page)

```
    "y": {"value":  1, "drift": null},
    "z": {"value":  0, "drift": null}
  },

  "rotation": {
    "r": {"value": 0, "unit": "rad", "uncertainty": {"value": 0, "unit": "rad"}, "drift": null},
    "s": {"value": 0, "unit": "rad", "uncertainty": {"value": 0, "unit": "rad"}, "drift": null},
    "t": {"value": 0, "unit": "rad", "uncertainty": {"value": 0, "unit": "rad"}, "drift": null}
  }
}
```

9 Acquisition

9.1 Sample stage rotation

The stage is assumed to perform a rotation around its $\vec{w}_O$ axis during the scan, resulting in a circular sample trajectory. Other trajectories can be modelled using *drifts*, especially drifts of the stage geometry parameters. For example, a helix scan can be modelled by starting with a standard circular trajectory of several rotations (as described in the following), and an additional drift of the stage centre's z position. If the scan trajectory is completely described by drifts, the stage rotation described here should be deactivated by setting `"start_angle":` and `"stop_angle":` to `0`.

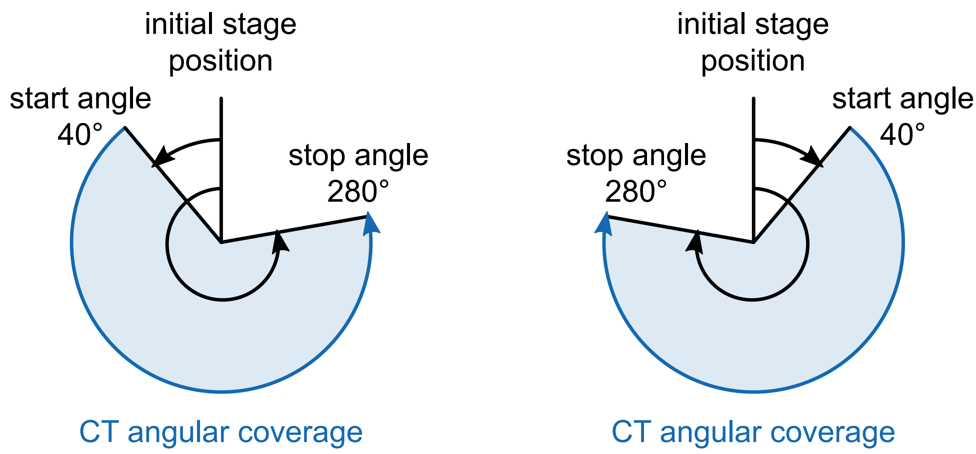
For a circular CT scan, the start and stop angle of the sample stage are defined in this section. An angle of `0` refers to the orientation of the stage as defined in the *geometry section* (for frame 0). All other angles express a rotation around the $\vec{w}_O$ axis of the stage. The direction of rotation must be mathematically positive in the case of counter-clockwise acquisition direction (`"CCW"`), and mathematically negative in case of clockwise acquisition direction (`"CW"`). The stage reaches the start and stop angle by rotating in the given `"direction":` around its normal axis (Fig. 9.1). This means that the `"direction":` parameter affects both the positions of start and stop angle, as well as the direction of rotation in which the CT scan is performed. It also means that **the start angle must always be less than (or equal to) the stop angle.** However, negative angular positions and positions greater than 360° are allowed, as well as an angular coverage of more than a full circle, e.g. to perform multiple rotations during one scan.

```
"start_angle": {"value":  40, "unit": "deg", "uncertainty": {"value": 0, "unit": "deg"}},
"stop_angle":  {"value": 280, "unit": "deg", "uncertainty": {"value": 0, "unit": "deg"}}
```

The direction of the sample stage rotation can be counter-clockwise (`"CCW"`, mathematically positive) or clockwise (`"CW"`, mathematically negative) around the $\vec{w}_O$ axis:

```
"direction": "CW"  "CCW"
```

The parameter `"scan_mode":` defines if the rotation stops while a projection is taken, or if it runs continuously.



(a) CCW
mathematically positive

(b) CW
mathematically negative

Fig. 9.1: The given start angle and stop angle refer to opposite angular positions in **(a)** counter-clockwise or **(b)** clockwise direction (as seen from "above" the stage). They describe the angular range covered by the CT scan.

```
337 "scan_mode": "stop+go"  "continuous"
```

The property for scan speed should only be used for continuous-motion scans. If undefined, it may be calculated from the detector's integration and dead time.

```
338 "scan_speed": {"value": 360, "unit": "deg/h", "uncertainty": 0, "uncertainty_unit": "deg/h"}
```

For stop&go scans, it should be set to `null`:

```
338 "scan_speed": null
```

9.2 Frames and projections

Number of projections

The `"number_of_projections":` is also given in the acquisition section.

```
339 "number_of_projections": 2001
```

Beginning from the start angle, the necessary number of angular steps is performed (in the case of a stop&go scan). It is assumed that a frame is taken before each step (starting with the first frame at the start angle). The parameter `"include_final_angle":` can be set to `true` if the last projection should be taken after the stop angle has been reached.

```
340 "include_final_angle": true  false
```

Frame averaging

The number of frames to be averaged for one projection image can be specified:

```
"frame_average": 3
```

Dark field and flat field acquisition and correction

If dark field and flat field images are acquired along with the projections, their numbers and frame averages can be specified. An `"ideal":` image means that the simulation of noise is to be omitted by the simulation software, possibly in contrast to the *noise* specification in the detector section. The parameter `"correction":` tells whether the projection images already come in a corrected form as a result of the scan (`true`) or if they are taken as uncorrected files (`false`).

```
"dark_field":
{
  "number": 1,
  "frame_average": 1,
  "ideal": true,
  "correction": true
},
"flat_field":
{
  "number": 3,
  "frame_average": 20,
  "ideal": false,
  "correction": true
}
```

Pixel binning

The number of pixels to bin in directions $vec{u}$ and $vec{v}$ of the detector:

```
"pixel_binning": {"u": 1, "v": 1}
```

The binning operation is not described here and left to the software.

9.3 Scattering

This parameter specifies if X-ray scattering should be simulated or not.

```
"scattering": false  true
```

10 Materials

In all previous declarations, materials are referred by their `"material_id":`. The last section of the JSON file, the materials array, finally contains the specifications for all materials used throughout the scenario description. Each element of the materials array must have the following properties:

`"id":` names the material ID that is used to refer to the material from the other sections of the file.

`"name":` is a trivial name that can be given to the material for better identification.

`"density":` provides the mass density of the material.

`"composition":` gives the chemical composition: a string of chemical symbols, each followed by their corresponding **number fraction** (integers or floating-point numbers) within the material. White space between symbols and numbers is allowed. If the number fraction of an element is 1, the number can be omitted and the next chemical symbol can follow right away. All chemical symbols start with a capital letter, potentially followed lowercase letters.

Example for a multi-component material:

```
{
  "id":   "Brass",
  "name": "Brass",
  "density": {"value": 8860, "unit": "kg/m^3", "uncertainty": null, "drift": null},
  "composition": {"value": "CuZn5", "drift": null}
 }
```

Example for a material with unusual number fractions, derived from a mass ratio of 40% Al_2O_3 and 60% SiO_2:

```
{
  "id":   "Glass Ceramic",
  "name": "Glass Ceramic",
  "density": {"value": 2.53, "unit": "g/cm^3", "uncertainty": null, "drift": null},
  "composition": {"value": "Al 0.21172 Si 0.28044 O 0.50784", "drift": null}
}
```

11 Software-specific properties

If a simulation software requires additional information or simulation parameters, those software-specific properties can be provided in the section called `"simulation":`. For each software, its own sub-section should be created.

```
"simulation":
{
  "aRTist":
  {
    "multisampling_detector": {"value": null, "drift": null},
    "multisampling_spot": {"value": "30", "drift": null},
```

(continues on next page)

(continued from previous page)

```
    "scattering_mcray_photons": {"value": "5e+007", "drift": null},
    "long_range_unsharpness":
    {
      "extension": {"value": 2, "unit":  "mm", "drift": null},
      "ratio":     {"value": 10, "unit": "%", "drift": null}
    }
  }
}
```

12 Full Example

```
{
  "file":
  {
    "name": "Example Scenario",
    "description": "Step cylinder in a rigid frame.",

    "contact": "David Plotzki",
    "date_created": "2020-04-23",
    "date_changed": "2021-04-18",
    "version": {"major": 1, "minor": 4},

    "file_type": "CTSimU Scenario",
    "file_format_version": {"major": 0, "minor": 9}
  },

  "environment":
  {
    "material_id": "Air",
    "temperature": {"value": 20, "unit": "C", "uncertainty": {"value": 0.5, "unit": "C"}, "drift": [{"file": "temperate_drift.csv", "unit": "C"}]}
  },

  "geometry":
  {
    "detector":
    {
      "centre": {
        "x": {"value": 400, "unit": "mm", "uncertainty": {"value": 0, "unit": "mm"}, "drift": null},
        "y": {"value":   0, "unit": "mm", "uncertainty": {"value": 0, "unit": "mm"}, "drift": null},
        "z": {"value":   0, "unit": "mm", "uncertainty": {"value": 0, "unit": "mm"}, "drift": null}
      },

      "vector_u": {
```

(continues on next page)

(continued from previous page)

```
        "x": {"value":  0, "drift": null},
        "y": {"value": -1, "drift": null},
        "z": {"value":  0, "drift": null}
      },
      "vector_w": {
        "x": {"value":  1, "drift": null},
        "y": {"value":  0, "drift": null},
        "z": {"value":  0, "drift": null}
      },

      "rotation": {
        "u": {"value": 0,      "unit": "rad", "uncertainty": {"value": 0, "unit": "rad"}, "drift": null},
        "v": {"value": 0,      "unit": "rad", "uncertainty": {"value": 0, "unit": "rad"}, "drift": null},
        "w": {"value": 2.3e-2, "unit": "rad", "uncertainty": {"value": 0, "unit": "rad"}, "drift": null},
        "known_to_reconstruction": true
      }
    },

    "source":
    {
      "type": "cone",
      "beam_divergence": {
        "u": {"value": 0, "unit": "rad", "uncertainty": {"value": 0, "unit": "rad"}, "drift": null},
        "v": {"value": 0, "unit": "rad", "uncertainty": {"value": 0, "unit": "rad"}, "drift": null}
      },

      "centre": {
        "x": {"value": 0, "unit": "mm", "uncertainty": {"value": 0, "unit": "mm"}, "drift": null},
        "y": {"value": 0, "unit": "mm", "uncertainty": {"value": 0, "unit": "mm"}, "drift": null},
        "z": {"value": 0, "unit": "mm", "uncertainty": {"value": 0, "unit": "mm"}, "drift": null}
      },

      "vector_u": {
        "x": {"value":  0, "drift": null},
        "y": {"value": -1, "drift": null},
        "z": {"value":  0, "drift": null}
      },
      "vector_w": {
        "x": {"value":  1, "drift": null},
        "y": {"value":  0, "drift": null},
        "z": {"value":  0, "drift": null}
```

(continues on next page)

(continued from previous page)

```
      },

      "rotation": {
        "u": {"value": 0, "unit": "rad", "uncertainty": {"value": 0, "unit": "rad"}, "drift": null},
        "v": {"value": 0, "unit": "rad", "uncertainty": {"value": 0, "unit": "rad"}, "drift": null},
        "w": {"value": 0, "unit": "rad", "uncertainty": {"value": 0, "unit": "rad"}, "drift": null},
        "known_to_reconstruction": true
      }
    },

    "stage":
    {
      "centre": {
        "x": {"value": 275, "unit": "mm", "uncertainty": {"value": 0, "unit": "mm"}, "drift": null},
        "y": {"value":   0, "unit": "mm", "uncertainty": {"value": 0, "unit": "mm"}, "drift": null},
        "z": {"value":   0, "unit": "mm", "uncertainty": {"value": 0, "unit": "mm"}, "drift": null}
      },

      "vector_u": {
        "x": {"value":  1, "drift": null},
        "y": {"value":  0, "drift": null},
        "z": {"value":  0, "drift": null}
      },
      "vector_w": {
        "x": {"value":  0, "drift": null},
        "y": {"value":  0, "drift": null},
        "z": {"value":  1, "drift": null}
      },

      "rotation": {
        "u": {"value":  0, "unit": "rad", "uncertainty": {"value": 0, "unit": "rad"}, "drift": null},
        "v": {"value":  0, "unit": "rad", "uncertainty": {"value": 0, "unit": "rad"}, "drift": null},
        "w": {"value":  0, "unit": "rad", "uncertainty": {"value": 0, "unit": "rad"}, "drift": null},
        "known_to_reconstruction": true
      }
    }
  },

  "detector":
  {
```

(continues on next page)

(continued from previous page)


```
    "model":        "",
    "manufacturer": "",
    "type":         "real",
    "columns":      {"value": 2000, "unit": "px"},
    "rows":         {"value": 1500, "unit": "px"},
    "pixel_pitch": {
      "u": {"value": 0.127, "unit": "mm", "uncertainty": {"value": 0, "unit": "mm"}, "drift": null},
      "v": {"value": 0.127, "unit": "mm", "uncertainty": {"value": 0, "unit": "mm"}, "drift": null}
    },
    "bit_depth":  {"value": 16},
    "integration_time": {"value": 0.5, "unit": "s", "uncertainty": {"value": 0, "unit": "s"}, "drift": null},
    "dead_time":  {"value": 50,  "unit": "ms", "uncertainty": {"value": 0, "unit": "s"}, "drift": null},
    "image_lag":  {"value": 0.05, "unit": null, "uncertainty": {"value": 0, "unit": null}, "drift": null},
    "grey_value":
    {
      "imax":   {"value": 45000, "drift": null},
      "imin":   {"value":  1000, "drift": null},

      "factor": {"value": 3e15, "unit": "1/J", "uncertainty": {"value": 0, "unit": "1/J"}, "drift": null},
      "offset": {"value": 1800, "unit": null,  "uncertainty": {"value": 0, "unit": null}, "drift": null},

      "intensity_characteristics_file": {"value": "detector_intensity.csv", "drift": null},

      "efficiency":   {"value": 0.7, "unit": null, "uncertainty": {"value": 0, "unit": null}, "drift": null},
      "efficiency_characteristics_file": {"value": "detector_efficiency.csv", "drift": null}
    },
    "noise":
    {
      "snr_at_imax": {"value": 205.3, "unit": null, "uncertainty": {"value": 0, "unit": null}, "drift": null},
      "noise_characteristics_file": {"value": "noise.csv", "drift": null}
    },
    "gain":
    {
      "value": 3,
      "drift": null,
      "scale_signal_and_noise": true
    },
    "unsharpness":
    {
      "basic_spatial_resolution": {"value": 0.1, "unit": "mm", "uncertainty": {"value": 0, "unit": "mm"}, "drift": null},
```


(continues on next page)

(continued from previous page)


```
      "mtf10_frequency": {"value": 10, "unit": "lp/mm", "uncertainty": {"value": 1, "unit": "lp/mm"}, "drift": null},
      "mtf": {"value": "detector_mtf.csv", "drift": null}
    },
    "bad_pixel_map":
    {
      "value": "badpixels.raw",
      "type": "uint8",
      "drift": null
    },
    "scintillator":
    {
      "material_id": "CsI",
      "thickness": {"value": 0.15, "unit": "mm", "uncertainty": {"value": 0, "unit": "mm"}, "drift": null}
    },
    "filters":
    {
      "front":
      [
        {
          "material_id": "Al",
          "thickness": {"value": 0.2, "unit": "mm", "uncertainty": {"value": 0, "unit": "mm"}, "drift": null}
        },
        {
          "material_id": "Kapton",
          "thickness": {"value": 0.13, "unit": "mm", "uncertainty": {"value": 0, "unit": "mm"}, "drift": null}
        }
      ],
      "rear":
      [
        {
          "material_id": "Al",
          "thickness": {"value": 2.0, "unit": "mm", "uncertainty": {"value": 0, "unit": "mm"}, "drift": null}
        }
      ]
    }
  },

  "source":
  {
    "model":        "",
    "manufacturer": "",
    "voltage": {"value": 130, "unit": "kV", "uncertainty": {"value": 0, "unit": "kV"}, "drift": null},
    "current": {"value": 100, "unit": "uA", "uncertainty": {"value": 0, "unit": "uA"}, "drift": null},
```


(continues on next page)

(continued from previous page)

```
    "target":
    {
      "material_id": "W",
      "type": "reflection",
      "thickness": null,
      "angle":
      {
        "incidence": {"value": 45, "unit": "deg", "uncertainty": {"value": 0, "unit": "deg"}, "drift": null},
        "emission":  {"value": 45, "unit": "deg", "uncertainty": {"value": 0, "unit": "deg"}, "drift": null}
      }
    },
    "spot":
    {
      "size": {
        "u": {"value": 100.0, "unit": "um", "uncertainty": {"value": 0, "unit": "um"}, "drift": null},
        "v": {"value": 100.0, "unit": "um", "uncertainty": {"value": 0, "unit": "um"}, "drift": null},
        "w": {"value":   0.0, "unit": "um", "uncertainty": {"value": 0, "unit": "um"}, "drift": null}
      },
      "sigma": {
        "u": {"value":  50.0, "unit": "um", "uncertainty": {"value": 0, "unit": "um"}, "drift": null},
        "v": {"value":  50.0, "unit": "um", "uncertainty": {"value": 0, "unit": "um"}, "drift": null},
        "w": {"value":   0.0, "unit": "um", "uncertainty": {"value": 0, "unit": "um"}, "drift": null}
      },
      "intensity_map": {
        "value": "spot_intensities.raw",
        "type": "float",
        "dim_x": 100,
        "dim_y": 100,
        "dim_z": 100,
        "drift": null
      }
    },
    "spectrum":
    {
      "monochromatic": false,
      "bremsstrahlung": true,
      "characteristic": true,
      "file": {"value": "tube_spectrum.csv", "drift": null}
    },
    "window":
    [
```

(continues on next page)

(continued from previous page)

```
      {
        "material_id": "Al",
        "thickness": {"value": 4.0, "unit": "mm", "uncertainty": {"value": 0, "unit": "mm"}, "drift": null}
      }
    ],
    "filters":
    [
      {
        "material_id": "Brass",
        "thickness": {"value": 0.5, "unit": "mm", "uncertainty": {"value": 0, "unit": "mm"}, "drift": null}
      },
      {
        "material_id": "Cu",
        "thickness": {"value": 0.25, "unit": "mm", "uncertainty": {"value": 0, "unit": "mm"}, "drift": null}
      }
    ]
  },

  "samples":
  [
    {
      "name": "Step Cylinder",
      "file": {"value": "stepCyl.stl", "drift": null},
      "unit": "mm",
      "scaling_factor": {
        "r": {"value": 1.0, "drift": null},
        "s": {"value": 1.0, "drift": null},
        "t": {"value": 1.0, "drift": null}
      },
      "material_id": "Glass Ceramic",
      "position":
      {
        "centre": {
          "u": {"value":  0, "unit": "mm", "uncertainty": {"value": 0, "unit": "mm"}, "drift": null},
          "v": {"value": 20, "unit": "mm", "uncertainty": {"value": 0, "unit": "mm"}, "drift": null},
          "w": {"value":  0, "unit": "mm", "uncertainty": {"value": 0, "unit": "mm"}, "drift": null}
        },

        "vector_r": {
          "u": {"value":  1,   "drift": null},
          "v": {"value":  0,   "drift": null},
          "w": {"value":  0,   "drift": null}
        },
```

(continues on next page)

(continued from previous page)


```
        "vector_t": {
          "u": {"value":  0,   "drift": null},
          "v": {"value": -0.2, "drift": null},
          "w": {"value":  1,   "drift": null}
        },

        "rotation": {
          "r": {"value": 0, "unit": "rad", "uncertainty": {"value": 0, "unit": "rad"}, "drift": null},
          "s": {"value": 0, "unit": "rad", "uncertainty": {"value": 0, "unit": "rad"}, "drift": null},
          "t": {"value": 0, "unit": "rad", "uncertainty": {"value": 0, "unit": "rad"}, "drift": null}
        }
      }
    },
    {
      "name": "Attachment Frame",
      "file": {"value": "frame.stl", "drift": null},
      "unit": "mm",
      "scaling_factor": {
        "r": {"value": 1.0, "drift": null},
        "s": {"value": 1.0, "drift": null},
        "t": {"value": 1.0, "drift": null}
      },
      "material_id": "Al",
      "position":
      {
        "centre": {
          "x": {"value": 275, "unit": "mm", "uncertainty": {"value": 0, "unit": "mm"}, "drift": null},
          "y": {"value":   0, "unit": "mm", "uncertainty": {"value": 0, "unit": "mm"}, "drift": null},
          "z": {"value":   0, "unit": "mm", "uncertainty": {"value": 0, "unit": "mm"}, "drift": null}
        },

        "vector_r": {
          "x": {"value":  1, "drift": null},
          "y": {"value":  0, "drift": null},
          "z": {"value":  0, "drift": null}
        },
        "vector_t": {
          "x": {"value":  0, "drift": null},
          "y": {"value":  1, "drift": null},
          "z": {"value":  0, "drift": null}
        },

        "rotation": {
```


(continues on next page)

(continued from previous page)


```
          "r": {"value": 0, "unit": "rad", "uncertainty": {"value": 0, "unit": "rad"}, "drift": null},
          "s": {"value": 0, "unit": "rad", "uncertainty": {"value": 0, "unit": "rad"}, "drift": null},
          "t": {"value": 0, "unit": "rad", "uncertainty": {"value": 0, "unit": "rad"}, "drift": null}
        }
      }
    }
  ],

  "acquisition":
  {
    "start_angle": {"value":    0, "unit": "deg", "uncertainty": {"value": 0, "unit": "deg"}},
    "stop_angle":  {"value":  360, "unit": "deg", "uncertainty": {"value": 0, "unit": "deg"}},
    "direction": "CCW",
    "scan_mode": "stop+go",
    "scan_speed": null,
    "number_of_projections": 2001,
    "include_final_angle": true,
    "frame_average": 3,
    "dark_field":
    {
      "number": 1,
      "frame_average": 1,
      "ideal": true,
      "correction": true
    },
    "flat_field":
    {
      "number": 3,
      "frame_average": 20,
      "ideal": false,
      "correction": true
    },
    "pixel_binning": {"u": 1, "v": 1},
    "scattering": false
  },

  "materials":
  [
    {
      "id":   "Air",
      "name": "Air",
      "density": {"value": 1.293, "unit": "kg/m^3", "uncertainty": {"value": 0, "unit": "kg/m^3"}, "drift": null},
      "composition": {"value": "N 0.785 O 0.21 Ar 0.005", "drift": null}
    },
    {
```


(continues on next page)

(continued from previous page)


```
      "id":   "Al",
      "name": "Aluminium",
      "density": {"value": 2.6989, "unit": "g/cm^3", "uncertainty": {"value": 0, "unit": "g/cm^3"}, "drift": null},
      "composition": {"value": "Al", "drift": null}
    },
    {
      "id":   "W",
      "name": "Tungsten",
      "density": {"value": 19.25, "unit": "g/cm^3", "uncertainty": {"value": 0, "unit": "g/cm^3"}, "drift": null},
      "composition": {"value": "W", "drift": null}
    },
    {
      "id":   "CsI",
      "name": "Caesium Iodide",
      "density": {"value": 4.51, "unit": "g/cm^3", "uncertainty": {"value": 0, "unit": "g/cm^3"}, "drift": null},
      "composition": {"value": "CsI", "drift": null}
    },
    {
      "id":   "Brass",
      "name": "Brass",
      "density": {"value": 8860, "unit": "kg/m^3", "uncertainty": {"value": 0, "unit": "g/cm^3"}, "drift": null},
      "composition": {"value": "CuZn5", "drift": null}
    },
    {
      "id":   "Cu",
      "name": "Copper",
      "density": {"value": 8.92, "unit": "g/cm^3", "uncertainty": {"value": 0, "unit": "g/cm^3"}, "drift": null},
      "composition": {"value": "Cu", "drift": null}
    },
    {
      "id":   "Kapton",
      "name": "Kapton 50HN (Polyimide film)",
      "density": {"value": 1.42, "unit": "g/cm^3", "uncertainty": {"value": 0, "unit": "g/cm^3"}, "drift": null},
      "composition": {"value": "C16H14N4O4", "drift": null}
    },
    {
      "id":   "Glass Ceramic",
      "name": "Glass Ceramic",
      "density": {"value": 2.53, "unit": "g/cm^3", "uncertainty": {"value": 0, "unit": "g/cm^3"}, "drift": null},
      "composition": {"value": "Al 0.21172 Si 0.28044 O 0.50784", "drift": null}
    }
  ],
```


(continues on next page)

(continued from previous page)

```
  "simulation":
  {
    "aRTist":
    {
      "multisampling_detector": {"value": null, "drift": null},
      "multisampling_spot": {"value": "30", "drift": null},
      "scattering_mcray_photons": {"value": "5e+007", "drift": null},
      "long_range_unsharpness":
      {
        "extension": {"value": 2, "unit":  "mm", "drift": null},
        "ratio":     {"value": 10, "unit": "%", "drift": null}
      }
    }
  }
}
```

References

[1] ASTM Subcommittee E07.01. ASTM E2597 / E2597M-14, Standard Practice for Manufacturing Characterization of Digital Detector Arrays. *ASTM International, West Conshohocken, PA, 2014, www.astm.org*, 2014. doi:10.1520/E2597_E2597M-14.

[2] Kurt Rossmann. Point Spread-Function, Line Spread-Function, and Modulation Transfer Function: Tools for the Study of Imaging Systems. *Radiology*, 93(2):257–272, August 1969. doi:10.1148/93.2.257.