Yokogawa MEG Reader Toolbox for MATLAB

Revision 1.4

Specifications

2011.05.06

Copyright (c) 2011 YOKOGAWA Electric Corporation. All rights reserved.

The Receiving Party shall strictly maintain the secrecy of any Confidential Information
and shall not disclose any and all Confidential Information of
YOKOGAWA Electric Corporation to any third parties.

The Receiving Party shall refrain from reverse engineering, decompiling or disassembling
in connection with the Confidential Information disclosed by YOKOGAWA Electric Corporation
to the Receiving Party unless expressly permitted by applicable law.

This toolbox is written for MATLAB version 7.5 (R2007b) onwards.

This toolbox must not be used for clinical applications.
If MEG data are processed by this toolbox, they should not be later employed
for clinical and/or diagnostic purposes.

This toolbox consists of following functions:

Category	Function name	Purpose
Read MEG data	getYkgwData	Get measurement data.
Read MEG header	getYkgwHdrSystem	Get information about system.
	getYkgwHdrChannel	Get information about channel.
	getYkgwHdrAcqCond	Get information about data acquisition condition.
	getYkgwHdrEvent	Get information about trigger event.
	getYkgwHdrCoregist	Get information about coregistration.
	getYkgwHdrDigitize	Get information about digitization.
	getYkgwHdrSubject	Get information about subject.
	getYkgwHdrBookmark	Get information about bookmark.
	getYkgwHdrSource	Get information about analyzed sources.
Read MRI	getYkgwMriHdr	Get information about header of MRI file (*.mri).
Others	getYkgwVersion	Get information about version of this toolbox.

getYkgwData

This function retrieves the measurement data of whole channels by the specified file path and sample range.

```
data = getYkgwData(
                filepath,
                start_sample,
                sample_length
       );
```

Arguments:

filepath	string	[in] File path
start_sample	double	[in] Start sample or trial(frame) number for retrieving data. The start number corresponding to each acquisition type is as follows : - Continuous Raw : Start sample number for retrieving data. (0 origin) - Evoked Average : Start sample number for retrieving data. (0 origin) - Evoked Raw : Start frame number for retrieving data. (1 origin) When both *start_sample* and *sample_length* are omitted, you can get data of whole samples.
sample_length	double	[in] Sample length for retrieving data. The number of samples or trials(frames) corresponding to each acquisition type is as follows : - Continuous Raw : Number of samples for retrieving data. - Evoked Average : Number of samples for retrieving data. - Evoked Raw : Number of trials(frames) for retrieving data. When this parameter is omitted or is specified as 'Inf', you can get data from start_sample to the end of sample(frame).

Return values:

data	matrix(double)	[out] double matrix of measurement data. Row : number of channels(whole channel), Column : number of samples Unit of the each channel depends on channel type as follows:

MagnetoMeter	[Tesla]
AxialGradioMeter	[Tesla]
PlanarGradioMeter	[Tesla]
ReferenceMagnetoMeter	[Tesla]
ReferenceAxialGradioMeter	[Tesla]
ReferencePlanarGradioMeter	[Tesla]
TriggerChannel	[Volt]
EegChannel	[Volt] *This has already been reflected EEG gain
EcgChannel	[Volt] *This has already been reflected ECG gain
EtcChannel	[Volt]
NullChannel	[Volt]

getYkgwHdrSystem

This function retrieves information of the system.

```
system_info = getYkgwHdrSystem(
                filepath
       );
```

Arguments:

filepath	string	[in] File path

Return values:

system_info	structure	[out] The structure of system information.
.version	double	Data version
.revision	double	Data revision
.system_id	double	System ID
.system_name	string	System name
.model_name	string	Model name

getYkgwHdrChannel

This function retrieves information about channel.

```
        channel_info        =        getYkgwHdrChannel(
                                                          filepath
                                     );
```

Arguments:

filepath	string	[in] File path

Return values:

channel_info	structure	[out] The structure of channel information.
.channel_count	double	The number of whole channels.
.channel	structure array	The detail information of channels. ('index 1' corresponds to 'channel 0')
.type	double	Channel type as follow table:

NullChannel	= 0;
MagnetoMeter	= 1;
AxialGradioMeter	= 2;
PlanarGradioMeter	= 3;
ReferenceMagnetoMeter	= 257;
ReferenceAxialGradioMeter	= 258;
ReferencePlanarGradioMeter	= 259;
TriggerChannel	= -1;
EegChannel	= -2;
EcgChannel	= -3;
EtcChannel	= -4;

.data	structure	The geometrical information of a channel. These fields is based on MEG device coordinate system. These fields of each channel type are as follows: See Figure.1 and Figure.2.

If channel type is AxialGradioMeter or ReferenceAxialGradioMeter (see Figure.3),

.x	double	x coordinate of inner coil position [meter]
.y	double	y coordinate of inner coil position [meter]
.z	double	z coordinate of inner coil position [meter]
.zdir	double	Sensor orientation from z-axis [degree]
.xdir	double	Sensor orientation from x-axis [degree]
.baseline	double	Baseline length [meter]
.size	double	Inner coil size [meter]
.name	string	Abbreviation name

If channel type is PlanarGradioMeter or ReferencePlanarGradioMeter (see Figure 4),

.x	double	x coordinate of inner coil position [meter]
.y	double	y coordinate of inner coil position [meter]
.z	double	z coordinate of inner coil position [meter]
.zdir1	double	Sensor orientation from z-axis [degree]
.xdir1	double	Sensor orientation from x-axis [degree]
.zdir2	double	Baseline orientation from z-axis [degree]
.xdir2	double	Baseline orientation from x-axis [degree]
.baseline	double	Baseline length [meter]
.size	double	Inner coil size [meter]

If channel type is MagnetoMeter or ReferenceMagnetoMeter,

.x	double	x coordinate of coil position [meter]
.y	double	y coordinate of coil position [meter]
.z	double	z coordinate of coil position [meter]
.zdir	double	Sensor orientation from z-axis [degree]
.xdir	double	Sensor orientation from x-axis [degree]
.size	double	Inner coil size [meter]
.name	string	Abbreviation name

If channel type is EegChannel or EcgChannel,

.type	double	Type
.id	double	ID
.name	string	Abbreviation name
.gain	double	Gain

If channel type is TriggerChannel or EtcChannel,

.type	double	Type
.id	double	ID
.name	string	Abbreviation name

If channel type is NullChannel, there is no field.

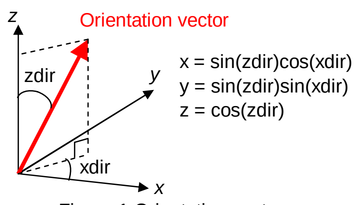

Figure.1 Orientation vector

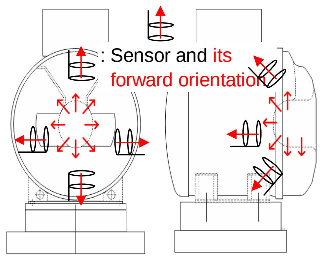

Figure.2 Sensor orientation in the dewar

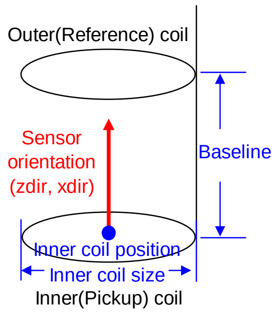

Figure.3 AxialGradioMeter parameter

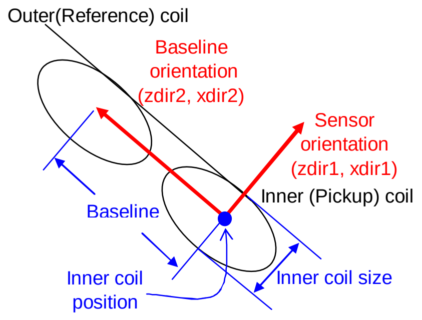

Figure.4 PlanarGradioMeter parameter

getYkgwHdrAcqCond
This function retrieves information about data acquisition condition.

```
acq_cond        =       getYkgwHdrAcqCond(
                                         filepath
                        );
```

Arguments:

filepath	string	[in] File path

Return values:

acq_cond	structure	[out] The structure of information about data acquisition condition.

.acq_type	double	Acquisition type
		AcqTypeContinuousRaw = 1;
		AcqTypeEvokedAve = 2;
		AcqTypeEvokedRaw = 3;

If acquisition type is AcqTypeContinuousRaw,		
.sample_rate	double	Sampling rate [Hz]
.sample_count	double	The number of samples which were actually acquired [sample]
.specified_sample_count	double	The number of samples which were specified before starting acquisition [sample]
If acquisition type is AcqTypeEvokedAve or AcqTypeEvokedRaw,		
.sample_rate	double	Sampling rate [Hz]
.frame_length	double	Frame length (The number of samples per one trial) [sample]
.pretrigger_length	double	Pretrigger length (The number of samples before trigger per one trial) [sample]
.average_count	double	The number of trials(frames) which were actually acquired [trial]
.specified_average_count	double	The number of trials(frames) which were specified before starting acquisition [trials]
.multi_trigger	structure	The structure of multi trigger information.
.enable	boolean	Is multi trigger mode ? (true : multi trigger mode)
.count	double	Number of multi triggers
.list	structure array	List of multi triggers (If not multi trigger mode, this structure array is set to empty.)
.enable	boolean	Is current multi trigger set to enable ? (true : enable)
.code	double	Event code (1 origin)
.name	string	Event name
.average_count	double	The number of trials(frames) which were actually acquired [trial]
.specified_average_count	double	The number of trials(frames) which were specified before starting acquisition [trials]

getYkgwHdrEvent
This function retrieves information about trigger event.

```
event           =       getYkgwHdrEvent(
                                         filepath
                        );
```

Arguments:

filepath	string	[in] File path

Return values:

event	structure array	[out] The structure array of trigger event corresponding to each trial.
.sample_no	double	Sample number of current event (0 origin)
.code	double	Event code (1 origin)
.name	string	Event name

getYkgwHdrCoregist

This function retrieves information about coregistration.

```
        coregist              =        getYkgwHdrCoregist(
                                                         filepath
                                       );
```

Arguments:

filepath	string	[in] File path

Return values:

coregist	structure	[out] The structure of information about coregistration.
.done	boolean	Is coregistration done ? (true : done)

.mri_type	double	MRI type
		NoMriFile = 0;
		NormalMriFile = 1;
		VirtualMriFile = 2;

.mri_file	string	File path of MRI file (*.mri)
.hpi_file	string	File path of HPI(Head Position Indicator) file (*.mrk)
.meg2mri	matrix(double)	4 x 4 matrix which transforms MEG device coordinate to MRI coordinate [meter] usage: [*xmri, ymri, zmri, 1*]' = *coregist.meg2mri* * [*xmeg, ymeg, zmeg, 1*]'
.mri2meg	matrix(double)	4 x 4 matrix which transforms MRI coordinate to MEG device coordinate [meter] usage: [*xmeg, ymeg, zmeg, 1*]' = *coregist.meg2mri* * [*xmri, ymri, zmri, 1*]'
.hpi	structure array	The structure array of HPI(Head Position Indicator)
.meg_pos	matrix(double)	HPI position [*x, y, z*] on MEG device coordinate [meter]
.mri_pos	matrix(double)	HPI position [*x, y, z*] on MRI coordinate [meter] (Before coregistration, this field is set to [0,0,0])
.label	string	HPI label as follows: 'LPA' : Left PreAuricular 'RPA' : Right PreAuricular 'CPF' : Center PreFrontal 'LPF' : Left PreFrontal 'RPF' : Right PreFrontal
.model	structure	The structure of conductor model.

.type	double	Conductor model type
		UNKNOWN_MODEL = -1;
		NO_MODEL = 0;
		SPHERICAL_MODEL = 1;
		LAYERED_MODEL = 2;

If Conductor model type is SPHERICAL_MODEL,

.cx	double	x coordinate of spherical center position on MRI coordinate [meter]
.cy	double	y coordinate of spherical center position on MRI coordinate [meter]
.cz	double	z coordinate of spherical center position on MRI coordinate [meter]
.radius	double	radius of spherical conductor on MRI coordinate [meter]

If Conductor model type is LAYERED_MODEL,

.ax	double	Coefficient 'ax' of planar equation 'ax * x + ay * y + az * z = c'
.ay	double	Coefficient 'ay' of planar equation 'ax * x + ay * y + az * z = c'
.az	double	Coefficient 'az' of planar equation 'ax * x + ay * y + az * z = c'
.c	double	Coefficient 'c' of planar equation 'ax * x + ay * y + az * z = c'

getYkgwHdrDigitize

This function retrieves information of the digitization.

```
        digitize              =        getYkgwHdrDigitize(
                                                         filepath
                                       );
```

Arguments:

filepath	string	[in] File path

Return values:

digitize	structure	[out] The structure of information and points about digitization.
.info	structure	The structure of information about digitization.
.digitizer_file	string	File path of digitizer file
.done	boolean	Is matching done? (true : done)
.meg2digitizer	matrix(double)	4 x 4 matrix which transforms MEG coordinate to Digitizer coordinate.
.digitizer2meg	matrix(double)	4 x 4 matrix which transforms Digitizer coordinate to MEG coordinate.
.point	structure array	The structure of point data about digitization.
.name	string	Point name
.x	double	x-coordinate on digitizer coordinate [meter]
.y	double	y-coordinate on digitizer coordinate [meter]
.z	double	z-coordinate on digitizer coordinate [meter]

getYkgwHdrSubject

This function retrieves information of the subject.

```
        subject         =       getYkgwHdrSubject(
                                                    filepath
                                );
```

Arguments:

filepath	string	[in] File path

Return values:

subject		[out] The structure of subject information.
.id	string	ID
.name	string	Name
.birthday	string	Birthday
.sex	string	Sex
.handed	string	Handed

getYkgwHdrBookmark

This function retrieves information about bookmark.

```
        bookmark        =       getYkgwHdrBookmark(
                                                    filepath
                                );
```

Arguments:

filepath	string	[in] File path

Return values:

bookmark	structure array	[out] The structure array of bookmark information.
.sample_no	double	Sample number of bookmark
.label	double	Label of bookmark
.comment	string	Comment of bookmark

getYkgwHdrSource

This function retrieves information of the sources.

```
        source          =       getYkgwHdrSource(
                                                    filepath
                                );
```

Arguments:

filepath	string	[in] File path

Return values:

source	structure array	[out] The structure array of analyzed source information. Note : Sources are arranged in order of estimated time.

.type	double	Type of source
		DipoleModel = 1;
		DistributedSourceModel = 2;

.time	double	Analyzed Time [second] from 1970.1.1
.sample_no	double	Time sample index of source
.channel_list	row vector(double)	Channel number (0 origin) list which used to estimate
.model	structure	The structure of conductor model.

.type	double	Conductor model type
		UNKNOWN_MODEL = -1;
		NO_MODEL = 0;
		SPHERICAL_MODEL = 1;
		LAYERED_MODEL = 2;

If Conductor model type is SPHERICAL_MODEL,		
.cx	double	x coordinate of spherical center position on MEG coordinate [meter]
.cy	double	y coordinate of spherical center position on MEG coordinate [meter]
.cz	double	z coordinate of spherical center position on MEG coordinate [meter]
.radius	double	radius of spherical conductor on MEG coordinate [meter]
If Conductor model type is LAYERED_MODEL,		
.ax	double	Coefficient 'ax' of planar equation 'ax * x + ay * y + az * z = c'
.ay	double	Coefficient 'ay' of planar equation 'ax * x + ay * y + az * z = c'
.az	double	Coefficient 'az' of planar equation 'ax * x + ay * y + az * z = c'
.c	double	Coefficient 'c' of planar equation 'ax * x + ay * y + az * z = c'

.algorithm	structure	The structure of conductor algorithm.

.magnetic_field_calc	double	Algorithm of magnetic field calculation
		BiotSavartLaw = 1;
		SarvasLaw = 2;
		MagneticDipoleLaw = 3;

.variable_restraint	double	Algorithm of variable restraint
		NoRestraint = 0;
		PositionRestraint = 1;
		DirectionRestraint = 2;
		IntensityRestraint = 3;

.optimization	double	Algorithm of optimization
		GradientAlgorithm = 1;
		LeadFieldReconstructionAlgorithm = 2;
		ManualSetAlgorithm = 3;
		UserAlgorithm = 4;

.filter	structure	The structure of spectral filter setting.
.hpf , .lpf	structure	The structure of high-pass / low-pass filter setting.
.enable	boolean	Does this filter enable?
.cutoff_frequency	double	Cutoff frequency [Hz]

.window_type	double	Window type
		NoWindow = 0;
		HanningWindow = 1;
		HammingWindow = 2;

.width	double	Filter width
.bpf, .bef	structure	The structure of band-pass / band-eliminate filter setting.
.enable	boolean	Does this filter enable?
.low_frequency	double	Low frequency [Hz]
.high_frequency	double	High frequency [Hz]
.window_type	double	Window type
.width	double	Filter width
.moveave	structure	The structure of moving average setting.
.enable	boolean	Does this filter enable?
.width	double	Filter width
.baseadj	structure	The structure of baseline adjustment setting.
.enable	boolean	Does this filter enable?

.type	double	Type of baseline adjustment
		PretriggerBaselineAdjust = 0;
		PosttriggerBaselineAdjust = 1;
		AllRangeBaselineAdjust = 2;
		ExplicitBaselineAdjust = 3;

.start_time	double	Start time [millisecond]
.end_time	double	End time [millisecond]
.gof	double	Goodness-of-fit (GOF)
.correlation	double	Corrlation Coefficiency
.label	double	Label
.comment	string	Comment
.total_intensity	double	Total intensity of sources
.dipole_count	double	Number of dipole sources
.dipole_list	structure array	The structure array of dipole sources
.x	double	x coordinate of dipole position on MEG coordinate [meter]
.y	double	y coordinate of dipole position on MEG coordinate [meter]
.z	double	z coordinate of dipole position on MEG coordinate [meter]
.zdir	double	Dipole orientation from z-axis [degree]
.xdir	double	Dipole orientation from z-axis [degree]
.intensity	double	Dipole intensity (moment) [Ampere Meter]

getYkgwMriHdr

This function retrieves header information of specified mri file (*.mri).

```
        mri_header              =       getYkgwMriHdr(
                                                              filepath
                                        );
```

Arguments:

filepath	string	[in] File path

Return values:

mri_header	structure	[out] The structure of mri header information.
.data_style	double	Data style (0 : DICOM, others : Polhemus)
.model	structure	The structure of conductor model.
.done	boolean	Is conductor model defined ? (true : defined)
.type	double	Conductor model type
		UNKNOWN_MODEL = -1;
		NO_MODEL = 0;
		SPHERICAL_MODEL = 1;
		LAYERED_MODEL = 2;
If Conductor model type is SPHERICAL_MODEL,		
.cx	double	x coordinate of spherical center position on MRI coordinate [meter]
.cy	double	y coordinate of spherical center position on MRI coordinate [meter]
.cz	double	z coordinate of spherical center position on MRI coordinate [meter]
.radius	double	radius of spherical conductor on MRI coordinate [meter]
If Conductor model type is LAYERED_MODEL,		
.ax	double	Coefficient 'ax' of planar equation 'ax * x + ay * y + az * z = c'
.ay	double	Coefficient 'ay' of planar equation 'ax * x + ay * y + az * z = c'
.az	double	Coefficient 'az' of planar equation 'ax * x + ay * y + az * z = c'
.c	double	Coefficient 'c' of planar equation 'ax * x + ay * y + az * z = c'
.hpi	structure array	The structure of point data about picked HPI.
.done	boolean	Is pick-up of a HPI point done ? (true : done)
.mri_pos	double	HPI position [x, y, z] on MRI coordinate [meter]
.label	string	HPI label as follows:
		'LPA' : Left PreAuricular
		'RPA' : Right PreAuricular
		'CPF' : Center PreFrontal
		'LPF' : Left PreFrontal
		'RPF' : Right PreFrontal
.image_parameter	structure	The structure of image parameters.
.intensity	vector(double)	1 x 2 row vector, minimum and maximum of image values
.initial_color	vector(double)	1 x 2 row vector, minimum and maximum of initial brightness
.color	vector(double)	1 x 2 row vector, minimum and maximum of current brightness
.normalize	structure	The structure of normalized HEAD coordinate system (LPA(x-), RPA(x+), nasion(y+)). See Figure.5.
.done	boolean	Is HEAD coordinate system defined ? (true : defined)
.mri2normalize	matrix(double)	4 x 4 matrix which transforms MRI coordinate to HEAD coordinate [meter]
		usage: [*xhead, yhead, zhead* , 1]' = *mri_header.normalize.mri2normalize* * [*xmri, ymri, zmri* , 1]'
.point	structure array	The structure of point data about HEAD fiducial points.
.done	boolean	Is pick-up of a HEAD fiducial point done ? (true : done)
.name	string	Name of HEAD fiducial points.
.x	double	x coordinate of a HEAD fiducial point on MRI coordinate [meter]
.y	double	y coordinate of a HEAD fiducial point on MRI coordinate [meter]
.z	double	z coordinate of a HEAD fiducial point on MRI coordinate [meter]
.besa_fiducial	structure	The structure of BESA fiducial information.
.point	structure array	The structure of point data about BESA fiducial points.
.done	boolean	Is pick-up of a BESA fiducial point done ? (true : done)
.x	double	x coordinate of a BESA fiducial point on MRI coordinate [meter]
.y	double	y coordinate of a BESA fiducial point on MRI coordinate [meter]
.z	double	z coordinate of a BESA fiducial point on MRI coordinate [meter]

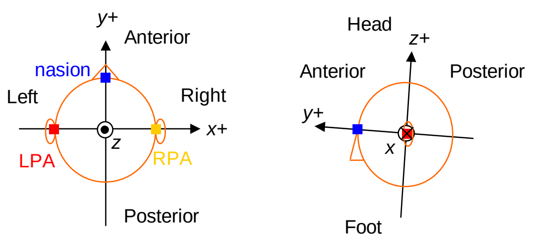

Figure.5 Normalized HEAD coordinate system

getYkgwVersion

This function retrieves version of this toolbox.

```
        ykgw_ver              =       getYkgwVersion;
```

Arguments: none

Return values:

ykgw_ver		[out] structure of toolbox version
.version	string	toolbox version major.minor[.build[.revision]]
.major	double	toolbox major version
.minor	double	toolbox minor version
.build	double	toolbox build version
.revision	double	toolbox revision version
.date	string	release date yyyy.mm.dd