

MATLAB TOOLBOX FOR C3DSERVER – VERSION 2

[Matthew R. Walker](#), [Michael J. Rainbow](#)

Motion Analysis Laboratory
Shriners Hospitals for Children – Erie PA (USA)

Version 2: April 21, 2006

DESCRIPTION

The C3D directory contains MATLAB functions used to activate and use Motion Lab Systems (MLS) C3Dserver as a COM object in MATLAB. This allows access to all of the functions within the C3Dserver. Some example functions are provided.

Version 2 Additions:

- New functions available to provide examples of C3Dserver capabilities.
- Notes have been added to this documentation on PC requirements and a syntax change for use in MATLAB that was not mentioned in Version 1 of this toolbox.

DISCLAIMER OF WARRANTIES & INDEMNITY

These functions are offered to the biomechanics community free of charge and “as is” to increase the availability of a commonly used file format (*.c3d) in MATLAB. The use of these functions is at your own risk. Development of these functions was solely for the convenience of the authors, and was not in association with MLS in any way.

Shriners Hospitals for Children, the Motion Analysis Laboratory at Shriners Hospitals for Children - Erie PA (USA), and/or the authors make no warranty, express or implied, including, without limitation, any implied warranties of merchantability or of fitness for a particular purpose with respect to any research, information, data, patent, trademark, software, non-public or other information, or tangible research property which may be provided to User hereunder and hereby disclaims the same.

USAGE

Request from Authors:

Please leave the authors names intact on the files. The authors also request acknowledgement if the use of these functions was found helpful for publications. If you find errors, have questions/ comments, or would like to provide or request additions to these M-files, please contact either Matthew Walker (matthewwalker_1@hotmail.com) OR Michael Rainbow (Michael_Rainbow@brown.edu).

Requirements:

You must have both MATLAB 7.0 (or newer) and MLS C3Dserver (evaluation or registered copy) installed. These M-files were created and tested with MATLAB 7.0 (R14) and MLS C3Dserver (v.129.3), operating on Windows XP Pro.

Please Note:

- Routines have not been tested with other versions of MATLAB. Problems resulting from use of this toolbox with older MATLAB versions are not supported by the authors.

Instructions:

1. Download **C3D.zip**, and add the directory to your MATLAB path.
2. Choose a variable name (for example: 'test'), and type at the MATLAB prompt:
`>> test = c3dserver`
3. A COM object variable 'test' is now loaded. Once loaded, the C3Dserver functions are considered 'methods' of the class **COM.C3DServer_C3D**. The status and version of C3Dserver you are using will be displayed (access is limited in evaluation mode).
4. There are several MATLAB functions that will show you the 'methods' (ie. functions) available through C3DServer. For example, type any of the following:
`>> methods(test) ; invoke(test) ; methodview(test)`
5. The M-files can now be used as typical MATLAB functions to open, save, create, or close a *.c3d file. For help, type: **help 'function name'**. To open a file, type:
`>> openc3d(test)`
Also notice additional arguments for increased flexibility in some of the functions.
6. With access to all of the functions in C3Dserver, users can now write additional MATLAB functions (similar to those M-files in the toolbox) to perform operations on the data in C3D files.

Please Note:

- When using C3Dserver functions, please refer to the most recent C3Dserver manual.
- The BYTE *byScaled* arguments for functions in the C3Dserver manual must be input into MATLAB code as a string (ie. in single-quotations). As an example, the final argument in the following code needs to be in single quotation marks:
`>> xyzpos = test.GetPointData(0, 0, test.GetVideoFrame(0), '1');`
- No other syntax differences have been found.

ACKNOWLEDGEMENTS

The authors would like to thank Frank L. Buczek, Jr., Ph.D., Director of the Motion Analysis Laboratory at Shriners Hospitals for Children – Erie PA (USA), for his support and suggestions during this process.

BRIEF DESCRIPTION OF M-FILES (for more info see m-file code)

- | | |
|--------------------|--|
| c3dserver.m | itf = c3dserver()
Activates C3DServer as a COM (COM.c3dserver_c3d), and returns 'itf' as a COM object for the server. |
| openc3d.m | openc3d(itf, getoption, fname)
Loads a C3D file into the C3DServer domain.

itf = variable name used for COM object
getoption = 1 for Windows dialog, other than 1 to enter filename
fname = filename, with or without path and c3d extension
Note: openc3d will handle 'fname' with or without the '.c3d' extension.
'fname' must be entered with directory path unless current MATLAB path is where the file is located. |

closec3d.m	closec3d(itf) Removes C3D file from the C3DServer domain. itf = variable name used for COM object
savec3d.m	savec3d(itf, newfname, filetype) Saves C3D file. itf = variable name used for COM object newfname = new filename to save, 'a' to save to open file filetype = Intel(1), DEC(2), SGI(3), existing type (-1) Note: ' newfname ' and ' filetype ' default to the name of the currently open file and the existing filetype if only ' itf ' is entered as input. savec3d will handle ' newfname ' with or without the '.c3d' extension. ' newfname ' can be entered with directory path.
createc3d.m	createc3d(itf,name,vfrt,nfram,nmkr,avr,achn,ftype,dtype,pscal) Creates C3D file with the input parameters. itf = variable name used for COM object name = name given to C3D file vfrt = video frame rate, must be > 0, default 120 nfram = number of frames, default 500 nmkr = number of markers, not < 0, default 20 avr = analog to video ratio, default 13 achn = number of analog channels, not < 0, default 28 ftype = Intel(1), DEC(2), SGI(3), default 1 dtype = Integer(1), Floating Point(2), default 2 pscal = point data scaling factor, must be > 0, default 0.1 Note: Only requires 2 inputs (' itf ' and ' name ') if user accepts defaults for other parameters. createc3d will handle ' name ' with or without the '.c3d' extension. ' name ' can be entered with directory path.
nframes.m	frames = nframes(itf) Returns integer value of the number of video frames contained in an open C3D file. itf = variable name used for COM object frames = integer value of number of video frames in c3d file
analog2 ... videoframes.m	truncated = analog2videoframes(itf, signals, index1, index2) Truncates input signals to video frame rate. itf = variable name used for COM object signals = array to be truncated index1 = video start frame index, all frames if not used as an argument index2 = video end frame index, all frames if not used as an argument truncated = truncated signals array

get3dtarget.m **XYZPOS = get3dtarget(itf, signalname, residual, index1, index2)**
Returns nx4 array containing X,Y,Z trajectory data and residual.

itf = variable name used for COM object
signalname = string name of desired marker
residual = Return matrix with point residual in column 4. 0 or no 3rd argument = false (returns nx3 with XYZ data only). 1 = true (returns nx4 with XYZ and residuals)
index1 = start frame index, all frames if not used as an argument
index2 = end frame index, all frames if not used as an argument
XYZPOS = nx3/4 matrix with n frames and X, Y, Z, and/or residual as columns

get3dtargets.m **XYZPOS = get3dtargets(itf, residual, index1, index2)**
Returns structure containing all X,Y,Z trajectory data and residuals if chosen.

itf = variable name used for COM object
residual = Return matrix with point residual in column 4. 0 or no 3rd argument = false (returns nx3 with XYZ data only). 1 = true (returns nx4 with XYZ and residuals)
index1 = start frame index, all frames if not used as an argument
index2 = end frame index, all frames if not used as an argument
XYZPOS = structure with target fields, x,y,z and/or residual columns

getanalog ... channel.m **ACHANNEL = getanalogchannel(itf, signalname, index1, index2)**
Returns nx1 array containing analog data. The returned analog data is scaled with offsets removed, but is in the force plate coordinate system.

itf = variable name used for COM object
signalname = string name of desired signal
index1 = start frame index, all frames if not used as an argument
index2 = end frame index, all frames if not used as an argument
ACHANNEL = nx1 matrix of analog data

getanalog ... channels.m **ACHANNEL = getanalogchannels(itf, index1, index2)**
Returns structure with nx1 analog data fields. The returned analog data is scaled with offsets removed, but is in the force plate coordinate system.

itf = variable name used for COM object
index1 = start frame index, all frames if not used as an argument
index2 = end frame index, all frames if not used as an argument
ACHANNEL = structure with nx1 matrix of analog data fields (+units)

**getemg ...
channels.m**

EMG = getemgchannels(itf,chanstart,chanend,type,index1,index2)
Returns emg data in raw or processed forms.

itf = variable name used for COM object

chanstart = first emg channel

chanend = last emg channel

type = 0 or no input for raw, 1 for FWR, 2 for LE (6Hz dual-pass)

index1 = start frame index, all frames if not used as an argument

index2 = end frame index, all frames if not used as an argument

EMG = stucture with channel fields