**Visual3D Guide to Automated Pipelines**

Syntax:

|  |  |
| --- | --- |
| ! | Commented out |
|  | Script Title |
| / | Command |
| ; | End script |
| :: | Pipeline Parameter (think of it as a placeholder variable where the value of the variable will be substituted) |
| & | Concatenation/Separation- Used to concatenate strings together and as a separator for the parser to parse pieces separately (ex. ::VISUAL3DFOLDER&Models) |
| \* | Multiply/All- Can be used to indicate all files \*.c3d or all run files \*run\*.c3d (ex. for files named S01\_run2.c3d) |

**Mathematical Operators**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **+** |  | add |
|  | **-** |  | subtract |
|  | **\*** |  | multiply |
|  | **/** |  | divide |
|  | **^** |  | power -> for example, x^2 = x to the power 2 |
|  | **|** |  | logical OR -> the adjective NOT is allowed |
|  | **&** |  | logical AND-> the adjective NOT is allowed |
|  | **<> or ><** |  | not equals |
|  | **==** |  | equals |
|  | **>= or =>** |  | greater than or equal to |

Tips & Tricks

You can open and edit Visual3D pipelines in notepad or notepad++

Expressions in V3D can refer to variables or scripts or commands

**Using Pipeline Parameters in an Expression**

The syntax for using a pipeline parameter as part of an expression is a bit unusual and takes an understanding of how Visual3D parses parameters and pre-processes commands.

The ampersand & is used in pipeline commands to concatenate strings together, and thus is a separator for the parser to find the pieces that need to be parsed separately.

For example, to use a pipeline parameter LP\_FREQ:

/EXPRESSION=2\*pi()\*&::LP\_FREQ

The ampersand tells the parser to take the 2\*pi()\* and the ::LP\_FREQ separately through the pre-parser. The first part is just taken as is since it doesn't have a prefix of :: while the value after the & does have a :: prefix, so it is substituted with the pipeline parameter.

If you have more complex expressions, you might need to surround each pipeline parameter with an ampersand

&::LP\_FREQ&\*&::MULTIPLIER&-&::CONSTANT which may evaluate to something like: 60\*1.4-90.0 once all the pipeline parameters are substituted.

The general rule is to surround the pipeline parameter with ampersands.

Pipeline Summary:

New Workspace

Set working folder

Open model file

Open motion files

Assign tags to files

Assign model to motion files

Assign subject height (=1 meter)

Calculate subject weight from force plate during static trial

Build Model

Interpolate and lowpass filter marker (target) and force (analog) data

Calculate gait events (HS, TO) using force plates

Compute variables

Save Workspace

Example Pipeline

!================================================================================

! Institution Name and pipeline title

! Created by: \_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_

! Description of what this pipeline does

!================================================================================

! Open new blank workspace

File\_New

;

! Use the processed (filtered) targets (marker trajectories) for pipeline calculations

Set\_Use\_Processed\_Targets

/USE\_PROCESSED=TRUE

;

! Used processed (filtered) force data for calculations

Set\_Use\_Processed\_Analog

/USE\_PROCESSED=TRUE

;

! Prompt the user for the working data folder, this will generate a pop-up for the user to select the

! folder where the C3D motion files are stored

Set\_Pipeline\_Parameter\_To\_Folder\_Path

/PARAMETER\_NAME=FOLDER

/PARAMETER\_VALUE=

! /PARAMETER\_VALUE\_SEARCH\_FOR=

! /PARAMETER\_VALUE\_REPLACE\_WITH=

! /PARAMETER\_VALUE\_APPEND=

/SET\_PROMPT=Browse to the subject directory

;

! Ask user for number of conditions

Prompt\_For\_Pipeline\_Parameter\_Value

/PIPELINE\_PARAMETER\_NAME=NumConditions

/PROMPT=How many conditions

/DATA\_TYPE=STRING

/DEFAULT\_VALUE=1

/USE\_POSSIBLE\_VALUES=TRUE

/POSSIBLE\_VALUES=1+2+3+4+5+6+7+8

! /USE\_UNITS=FALSE

! /DEFAULT\_UNITS=

! /USE\_POSSIBLE\_UNITS=FALSE

! /POSSIBLE\_UNITS=

! /CONVERSIONS=

;

!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

! Bring in static file and apply model template - using a parameter from for loop

!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

! Prompt user to enter number of conditions to iterate thru

Set\_Pipeline\_Parameter\_From\_For\_Loop

/PARAMETER\_NAME=Conditions

/PARAMETER\_INDEX\_START=1

/PARAMETER\_INDEX\_END=::NumConditions

/PARAMETER\_INDEX\_STEP=1

! /PARAMETER\_INDEX\_TYPE=INTEGER

;

! Start a for loop for each condition and give the for loop a parameter name (ex. INDEX)

For\_Each

/ITERATION\_PARAMETER\_NAME=INDEX

! /ITERATION\_PARAMETER\_COUNT\_NAME=

/ITEMS=::Conditions

;

! Define model parameter for static trial

Set\_Pipeline\_Parameter

/PARAMETER\_NAME=MODELSTD

/PARAMETER\_VALUE=::FOLDER&\*B&::INDEX&\_std1.c3d

! /PARAMETER\_VALUE\_SEARCH\_FOR=

! /PARAMETER\_VALUE\_REPLACE\_WITH=

! /PARAMETER\_VALUE\_PREFIX=

! /PARAMETER\_VALUE\_APPEND=

! /MULTI\_PASS=FALSE

;

! Create model from static trial

Create\_Hybrid\_Model

/CALIBRATION\_FILE=::MODELSTD

!/SUFFIX=

! /RANGE=ALL\_FRAMES

;

!Apply model to standing \_Calibration file----

Apply\_Model\_Template

/CALIBRATION\_FILE=::MODELSTD

/MODEL\_TEMPLATE=::FOLDER&\*.mdh

!/SET\_PROMPT=Open model file

/VIEW\_BUILDMODEL\_RESULTS=

;

! End for loop for each condition

End\_For\_Each

/ITERATION\_PARAMETER\_NAME=INDEX

;

!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

! Open motion files - C3D files - and assign tag and model

!++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

! Open all run .c3d files from the working folder

File\_Open

/FILE\_NAME=::FOLDER&\*.c3d

/SUFFIX=\_Calibration

! /SET\_PROMPT=File\_Open

! /FILTER=

;

! Assign tags to static file

Assign\_Tags\_To\_Files

/MOTION\_FILE\_NAMES=\*std\*

/TAGS=STAND

;

! Assign "Run" tag to all files but not static

Assign\_Tags\_To\_Files

/MOTION\_FILE\_NAMES=\*run\*.c3d

/QUERY=NOT(STAND)

/TAGS=run

;

! Loop to assign tags to conditions and assign model to condition tags

For\_Each

/ITERATION\_PARAMETER\_NAME=INDEX

! /ITERATION\_PARAMETER\_COUNT\_NAME=

/ITEMS=::Conditions

;

Assign\_Tags\_To\_Files

/MOTION\_FILE\_NAMES=\*B&::INDEX&\*.c3d

! /QUERY=

/TAGS=B&::INDEX

;

Combine\_Tags

/ORIGINAL\_TAGS=B&::INDEX+run

/LOGICAL\_OPERATORS=AND

/NEW\_TAG=B&::INDEX&\_Run

;

Combine\_Tags

/ORIGINAL\_TAGS=B&::INDEX+STAND

/LOGICAL\_OPERATORS=AND

/NEW\_TAG=B&::INDEX&\_STAND

;

!--------------------------------------------------------------------

!Assign the motion files to the corresponding standing model file----

!--------------------------------------------------------------------

!Assign the motion files to the corresponding standing model file----

Assign\_Model\_File

/CALIBRATION\_FILE=::FOLDER&\*B&::INDEX&\_std1.c3d

/MOTION\_FILE\_NAMES=B&::INDEX

! /REMOVE\_EXISTING\_ASSIGNMENTS=FALSE

;

! Set height to 1 to eliminate error or warning

Set\_Subject\_Height

/CALIBRATION\_FILE=::FOLDER&\*B&::INDEX&\_std1.c3d

/Height=1

;

End\_For\_Each

/ITERATION\_PARAMETER\_NAME=INDEX

;

Build\_Model

! /CALIBRATION\_FILE=

/REBUILD\_ALL\_MODELS=TRUE

! /DISPLAY\_RESULTS=TRUE

;

!+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

! This section automatically computes standing body weight from force plate

!+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

For\_Each

/ITERATION\_PARAMETER\_NAME=INDEX

! /ITERATION\_PARAMETER\_COUNT\_NAME=

/ITEMS=::Conditions

;

! Select active file - tagged STAND for static trial

Select\_Active\_File

/FILE\_NAME=S&::INDEX&\_STAND

! /QUERY=

;

! Filter Analog @ 20 hz rather than Force, COP, Freemoment if we are going to use force structure

! Force structure adds all force plates together to treat as 1 big force plate

Lowpass\_Filter

/SIGNAL\_TYPES=ANALOG

! /SIGNAL\_FOLDER=ORIGINAL

! /SIGNAL\_NAMES=

! /RESULT\_FOLDER=PROCESSED

! /RESULT\_SUFFIX=

! /FILTER\_CLASS=BUTTERWORTH

/FREQUENCY\_CUTOFF=50

! /NUM\_REFLECTED=6

! /NUM\_EXTRAPOLATED=0

! /TOTAL\_BUFFER\_SIZE=6

! /NUM\_BIDIRECTIONAL\_PASSES=1

;

! Adds force plates 2, 3, 4, 5, and 6 together to treat as 1 big force plate (force plate 1 is in a different

! part of the lab)

Modify\_Force\_Structure\_Parameters

/USED=1

/TYPE=0

/NUM\_FP\_IN=5

/FP\_INDEX=2+3+4+5+6

/NUM\_SURFACES\_OUT=1

! /SPEED\_VALUES=

! /SPEED\_CHANNELS=

! /SPEED\_SCALES=

/CORNER1=1200.110078+1208.050132+0.000000

/CORNER2=-604.800105+1208.050132+0.000000

/CORNER3=-604.800105+-600.040019+0.000000

/CORNER4=1200.110078+-600.040019+0.000000

! /USE\_FORCES\_FOR\_KINETICS=TRUE

/COMBINE\_INPUT\_FORCES=TRUE

! /UPDATE\_C3D\_FILE=FALSE

;

! Set a START and END Frame for the calculations - needs to be a timeframe where there is little

! movement and subject's weight is on the plate - the static calibration file is the best trial to use for this

! Set a Start Event at frame 1

Explicit

/EVENT\_NAME=Start

/FRAME=1

!/TIME=

;

! Set an End Event at frame EOF

Event\_Explicit

/EVENT\_NAME=End

/FRAME=EOF

!/TIME=

;

! Calculate mean weight by calculating the mean vertical force between START and END events

Metric\_Mean

/RESULT\_METRIC\_NAME=Body\_Weight

! /APPLY\_AS\_SUFFIX\_TO\_SIGNAL\_NAME=FALSE

! /RESULT\_METRIC\_FOLDER=PROCESSED

/SIGNAL\_TYPES=FORCE

/SIGNAL\_NAMES=FS1\_1

/SIGNAL\_FOLDER=ORIGINAL

/SIGNAL\_COMPONENTS=Z

/EVENT\_SEQUENCE=Start+End

/EXCLUDE\_EVENTS=

/GENERATE\_MEAN\_AND\_STDDEV=FALSE

! /APPEND\_TO\_EXISTING\_VALUES=FALSE

;

! Calculate Mass from force by dividing by gravity and saving the result in the metric folder

Divide\_Signal\_By\_Constant

/SIGNAL\_TYPES=METRIC

/SIGNAL\_NAMES=Body\_Weight

/SIGNAL\_FOLDER=PROCESSED

/RESULT\_NAMES=Body\_Mass

/RESULT\_TYPES=METRIC

/RESULT\_FOLDER=PROCESSED

! /RESULT\_SUFFIX=

/SIGNAL\_COMPONENTS=0

/CONSTANT=9.81

;

! Set a parameter value for body mass, this saves the body mass as a variable in the pipeline

Set\_Pipeline\_Parameter\_To\_Data\_Value

/PARAMETER\_NAME=Body\_Mass

/SIGNAL\_TYPES=METRIC

/SIGNAL\_NAMES=Body\_Mass

/SIGNAL\_FOLDER=PROCESSED

/SIGNAL\_COMPONENTS=0

;

! Set Subject weight

Set\_Subject\_Weight

/CALIBRATION\_FILE=::FOLDER&\*S&::INDEX&\_static\*.c3d

/WEIGHT= ::Body\_Mass

;

End\_For\_Each

/ITERATION\_PARAMETER\_NAME=INDEX

;

Build\_Model

! /CALIBRATION\_FILE=

/REBUILD\_ALL\_MODELS=TRUE

! /DISPLAY\_RESULTS=TRUE

;

Recalc

;

!+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

! Interpolate and lowpass filter targets and Analog (force)

!+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

! Select ALL\_FILES as the active files

Select\_Active\_File

/FILE\_NAME=ALL\_FILES

;

! Interpolate gap based on 10% of sample rate (200 Hz is 20)

Interpolate

/SIGNAL\_TYPES=TARGET

! /SIGNAL\_NAMES=

/SIGNAL\_FOLDER=ORIGINAL

! /RESULT\_SUFFIX=

/RESULT\_FOLDER=PROCESSED

/MAXIMUM\_GAP=20

! /NUM\_FIT=3

! /POLYNOMIAL\_ORDER=3

;

! 13 Hz butterworth filter

Lowpass\_Filter

/SIGNAL\_TYPES=TARGET

! /SIGNAL\_NAMES=

/SIGNAL\_FOLDER=PROCESSED

! /RESULT\_SUFFIX=

/RESULT\_FOLDER=PROCESSED

! /FILTER\_CLASS=BUTTERWORTH

/FREQUENCY\_CUTOFF=13

! /NUM\_REFLECTED=6

! /TOTAL\_BUFFER\_SIZE=6

! /NUM\_BIDIRECTIONAL\_PASSES=1

;

Recalc

;

!==================================================

! Calculate Gait Events - HS, TO - using force plates

!==================================================

! Select motion files as the active files

! This will select files with the “run” tag

Select\_Active\_File

/FILE\_NAME=run

;

! Automatically computes timing of gait events (heel strike (HS) + toe-off (TO)) based on force plate

! contacts. USE\_TPR=TRUE estimates kinematic (non-force plate) gait events using algorithm based on

! force plate gait events

Automatic\_Gait\_Events

! /FRAME\_WINDOW=8

! /USE\_TPR=TRUE

;

!==================================================

! Calculate Variables

!==================================================

! Calculates right knee angle (angle between right shank (RSK) and right thigh (RTH) segments) using

! cardan sequence

Compute\_Model\_Based\_Data

/RESULT\_NAME=Right\_Knee\_Angle

/FUNCTION=JOINT\_ANGLE

/SEGMENT=RSK

/REFERENCE\_SEGMENT=RTH

/RESOLUTION\_COORDINATE\_SYSTEM=

/USE\_CARDAN\_SEQUENCE=TRUE

! /NORMALIZATION=FALSE

! /NORMALIZATION\_METHOD=

! /NORMALIZATION\_METRIC=

! /NEGATEX=FALSE

! /NEGATEY=FALSE

! /NEGATEZ=FALSE

! /AXIS1=X

! /AXIS2=Y

! /AXIS3=Z

! /TREADMILL\_DATA=FALSE

! /TREADMILL\_DIRECTION=UNIT\_VECTOR(0,1,0)

! /TREADMILL\_SPEED=0.0

;

! Calculates right ankle moment (joint moment between right shank (RSK) and right foot (RFT)); can

! normalize joint moment to body weight if desired

Compute\_Model\_Based\_Data

/RESULT\_NAME=Right\_Ankle\_Moment

/FUNCTION=JOINT\_MOMENT

/SEGMENT=RSK

/REFERENCE\_SEGMENT=RFT

/RESOLUTION\_COORDINATE\_SYSTEM=

/USE\_CARDAN\_SEQUENCE=TRUE

! /NORMALIZATION=FALSE

! /NORMALIZATION\_METHOD=

! /NORMALIZATION\_METRIC=

! /NEGATEX=FALSE

! /NEGATEY=FALSE

! /NEGATEZ=FALSE

! /AXIS1=X

! /AXIS2=Y

! /AXIS3=Z

! /TREADMILL\_DATA=FALSE

! /TREADMILL\_DIRECTION=UNIT\_VECTOR(0,1,0)

! /TREADMILL\_SPEED=0.0

;

Recalc

;

!====================================================================================

! Save Workspace As

!====================================================================================

!----Prompt the user to enter the subject number----

Prompt\_For\_Pipeline\_Parameter\_Value

/PIPELINE\_PARAMETER\_NAME=Subject\_number

/PROMPT=Enter the subject code (ex. A01):

/DATA\_TYPE=CHAR

/DEFAULT\_VALUE=

;

File\_Save\_As

/FILE\_NAME=::FOLDER&::Subject\_number&\_Results.cmz

! /SET\_PROMPT=Save CMO file as

! /SAVE\_EMBEDDED\_GRAPHICS=FALSE

;