Mocap to Robot Software Package User Manual

Klevis Aliaj

1 Introduction

This user manual aims to provide a brief overview of the "Mocap to Robot" software package. This package provides a set of tasks that aid in transforming motion capture data to a motion program for an industrial robot. It also provides tasks for verifying the resulting robot motion against the original motion capture trajectories. The task of optimally mapping a motion capture trajectory to a robot joint space trajectory is performed within a separate C++ software package. The "Mocap to Robot" package relies on the "Mocap to Robot" software library, which implements the underlying logic that supports the tasks contained in this package. From this perspective, this software package can be seen as a thin wrapper over the "Mocap to Robot" software library that reads parameters from configuration files and makes the appropriate calls to the underlying library functions.

2 Installation

The "Mocap to Robot" software package has been tested using Matlab R2018b and Python 3.6.8. Python was installed as part of the Anaconda 4.7.11 installation which provides the necessary scientific computing packages. The following additional Python packages are necessary:

- Plotly 4.1.0
- colorlover 0.3.0

This package relies on three configuration files (sample configuration files are supplied as part of each release) in order to function properly. Upon cloning or downloading the "Mocap to Robot" package a "parameters" directory must be created in the root directory of the package and the aforementioned configuration files placed in the "parameters" directory as follows:

- `Docs`
- `initClasses`
- `parameters`
  - `parametersM20.xml`
  - `robotM20.xml`
  - `tpProgram_humerus.xml`
- ...

As mentioned previously, the "Mocap to Robot" software package relies on the "Mocap to Robot" software library, which needs to be downloaded or cloned. It is recommended that the same release version is utilized for both the package and the library. The **mocapToRobotLibPath** parameter of the `parametersM20.xml` configuration file must be updated to point to the root directory of the "Mocap to Robot" software library (which should contain `initMocapToRobotLib.m`).

3 Task Listing

- Motion Capture to Robot Motion Program
  - Augment and Plot Trajectory Single
  - Augment and Plot Trajectory All
  - Augment and Export Trajectory Single
  - Augment and Export Trajectory All
  - Smooth and Plot By File
  - Smooth and Plot All
  - Smooth and Export By File
  - Smooth and Export All
  - Plot Optimized Trajectory By File
  - Plot All Optimized Trajectories
  - Process All Optimized Trajectories
  - Subsample by File
  - Subsample Plot by File
  - Subsample All
  - Subsample All Uniformly
  - Compute Subsampling Errors
  - Process Subsampling Errors
  - Write Teach Pendant Program
  - Write Linear Teach Pendant Program
- Robot Trajectory Verification
  - Establish Humerus Coordinate System
  - NDI to Robot Transform
  - Graph Paths
  - Simple Joint Plot
  - Process NDI Capture
  - Check File
  - Check Multiple Folders
  - Check Joints
  - Post Analyze File
  - Post Analyze Single Folder
  - Post Analyze Multiple Folders
  - Batch Process All Data

4 Data Organization

Although the intention of this document is to explain the "Mocap to Robot" software package, this section provides a brief overview of the data that this software package analyzes. This is done to provide background for the tasks contained in this package. The motion capture data and the robot trajectory verification data are housed in separate folders.

The motion capture data is organized first by activity, then by subject. Each subject will have up to 3 trials per activity. The motion capture data for each trial is exported from Visual3D and is contained in a `.c3d.txt` file. All the other files associated with a trial are obtained by one of the tasks above.

The robot trajectory verification data is organized first by activity, then by date upon which the experimental session took place. Each date will contain files related to robot frame identification procedure and one or more trial verifications. The verification data was collected using a custom hemisphere rigid body comprised of 16 light emitting diodes and the Northern Digital Inc. (NDI) Optotrak Certus. The NDI software outputs both the 3D positions of each of the markers and the 6D pose (position and orientation) of the rigid body. The files related to the robot frame identification procedure are:

- `Joint13D.csv` - marker trajectories while robot rotates its 1^{st} joint
- `Joint16D.csv` - hemisphere trajectory while robot rotates its 1^{st} joint
- `Joint23D.csv` - marker trajectories while robot rotates its 2^{nd} joint
- `Joint26D.csv` - hemisphere trajectory while robot rotates its 2^{nd} joint
- `Joint43D.csv` - marker trajectories while robot rotates its 4^{th} joint
- `Joint46D.csv` - hemisphere trajectory while robot rotates its 4^{th} joint
- `Position3D0.csv` - marker trajectories while robot rotates (without translating) about its end-effector joint
- `Position6D0.csv` - hemisphere trajectory while robot rotates (without translating) about its end-effector

The files related to the trial verification are:

- `[Activity]_[SubjectId]_[TrialId]_[SpeedMultiplier]_[(Non)UniformSubampling][DeprecatedIdentifiers]_3D.csv` - marker trajectories while robot performs the activity for the subject and trial indicated.
- `[Activity]_[SubjectId]_[TrialId]_[SpeedMultiplier]_[(Non)UniformSubampling][DeprecatedIdentifiers]_6D.csv` - hemisphere trajectory while robot performs the activity for the subject and trial indicated.
- `[Activity]_[SubjectId]_[TrialId]_[SpeedMultiplier]_[(Non)UniformSubampling][DeprecatedIdentifiers]_jointPos.csv` - positions of the robot's joints as captured from the M20ia robot controller.
- `[Activity]_[SubjectId]_[TrialId].toolframe.xml` - if the main humerus tool frame was not utilized to recreate this trajectory (activity/subject/trial), then this file contains the tool frame that was utilized.

5 Configuration Files

5.1 robotM20.xml

The `robotM20.xml` configuration file contains information regarding the robot utilized (M20ia) as well as the tools attached to the robot. All tasks under `tasks/opticalTrackingAnalysis` utilize this configuration file, but only a subset of the tasks under `tasks/trajectoryGeneration` utilize this configuration file. This configuration file contains the following parameters:

- **endEffector** - The name of the link within the URDF file for the robot representing the end-effector of the robot (e.g. tool0).
- **base** - The name of the link within the URDF file for the robot representing the base of the robot (e.g. base).

- **HSToolFrame** - The hemisphere tool frame with respect to the end-effector of the robot. Position is specified in mm. Orientation is specified via Euler angles in degrees. FANUC uses the XYZ extrinsic (or equivalently the zy'x" intrinsic) convention and that is utilized here.
- **HumerusToolFrame** - The humerus tool frame with respect to the end-effector of the robot. Position is specified in mm. Orientation is specified via Euler angles in degrees. FANUC uses the XYZ extrinsic (or equivalently the zy'x" intrinsic) convention and that is utilized here.
- **urdf** - File path to the Unified Robotics Description Format (URDF) file for the robot. (e.g. `D:\Optimization_Algorithm_Data\m20iag.urdf`).
- **velLimits** - A 6 component vector specifiying the velocity limits of each of the six joints of the robot - specified in rad/second.

5.2 tpProgram_humerus.xml

The `tpProgram_humerus.xml` file contains information regarding the teach pendant (motion) program for the robot utilized (M20ia). Only the **Write Teach Pendant Program** and **Write Teach Pendant Program Linear** tasks utilize this configuration file. This configuration file contains the following parameters:

- **tpProgramCntTag** - The CNT tag to utilize in the FANUC motion program.
- **tpProgramTimer** - The ID of the timer (specified by a positive integer) utilized by FANUC for timing the motion program.
- **tpProgramUFrame** - User frame ID (specified by a positive integer) utilized in the FANUC motion program to identify the reference frame for programmed positions.
- **tpProgramUTool** - Tool frame ID (specified by a positive integer) utilized in the FANUC motion program to identify the humerus tool frame.

5.3 parametersM20.xml

The `parameterM20.xml` file contains parameters for all of the tasks detailed in this manual. This configuration file contains the following parameters:

- **augmentNumPoints** - A 6-component vector specifying the number of frames to utilize for augmenting the trajectory at the beginning and the end. The first 3 components specify the number of frames for each of the 3 dimensions of position in the motion capture reference frame. The last 3 components specify the number of frames for each of the 3 dimensions of orientation (rotation vector) in the motion capture reference frame. This should be larger than the **interpNumPoints** parameter.
- **axisCalibrationFolder** - Path to the folder containing the hemisphere trajectory during the robot reference frame identification procedure. The files produced as part of this procedure are described in Data Organization (e.g. `D:\Motion_Verification_Data\JJ\05162018`).
- **axisCalibrationJ1Offset** - This parameter specifies the offset from zero of the angle of the first robot joint. This is not to be confused with the **tpProgramJ1** configuration parameter - the two are not related and are independent of each other.
- **axisCalibrationResults** - This parameter specifies the path to a file where the results of processing the dataset of the robot reference frame identification procedure should be stored.
- **axisCalibrationTcp** - This parameter specifies whether to determine the tool center point of the hemisphere from the dataset generated by the robot reference frame identification procedure.
- **batchProcessIR** - This parameter determines whether internal rotation trials will be batch processed. An integer value of 1 indicates that internal rotation trials will be batch processed, while an integer value of 0 indicates that internal rotation trials will NOT be batch processed.

- **batchProcessIRMocapData** - Path to the directory for the internal rotation motion capture data for all subjects (e.g. `D:\Motion_Capture_Data\IR`).
- **batchProcessIRVerificationData** - Path to the directory for the internal rotation robot motion verification data for all subjects (e.g. `D:\Motion_Verification_Data\IR`).
- **batchProcessJJ** - This parameter determines whether jumping jacks will be batch processed. An integer value of 1 indicates that jumping jacks will be batch processed, while an integer value of 0 indicates that jumping jacks will NOT be batch processed.
- **batchProcessJJMocapData** - Path to the directory for the jumping jacks motion capture data for all subjects (e.g. `D:\Motion_Capture_Data\JJ`).
- **batchProcessJJVerificationData** - Path to the directory for the jumping jacks robot motion verification data for all subjects (e.g. `D:\Motion_Verification_Data\JJ`).
- **batchProcessJL** - This parameter determines whether jug lifts will be batch processed. An integer value of 1 indicates that jug lifts will be batch processed, while an integer value of 0 indicates that jug lifts will NOT be batch processed.
- **batchProcessJLMocapData** - Path to the directory for the jug lifts motion capture data for all subjects (e.g. `D:\Motion_Capture_Data\JL`).
- **batchProcessJLVerificationData** - Path to the directory for the jug lifts robot motion verification data for all subjects (e.g. `D:\Motion_Verification_Data\JL`).
- **batchProcessJO** - This parameter determines whether jogging trials will be batch processed. An integer value of 1 indicates that jogging trials will be batch processed, while an integer value of 0 indicates that jogging trials will NOT be batch processed.
- **batchProcessJOMocapData** - Path to the directory for the jogging motion capture data for all subjects (e.g. `D:\Motion_Capture_Data\JO`).
- **batchProcessJOVerificationData** - Path to the directory for the jogging robot motion verification data for all subjects (e.g. `D:\Motion_Verification_Data\JO`).
- **batchProcessPrintDir** - Directory where graphs for batch processing are created.
- **butterworthCutoff** - Butterworth filter cutoff frequency (in Hz) to utilize for smoothing.
- **butterworthOrder** - Butterworth filter order to utilize for smoothing.
- **computeSubsamplingErrorsDir** - Directory path for a single motion capture activity containing multiple subjects (e.g. `D:\Motion_Capture_Data\JJ`).
- **interpNumPoints** - A 6-component vector specifying the number of frames to utilize when interpolating the trajectory at the beginning and the end. The first 3 components specify the number of frames for each of the 3 dimensions of position in the motion capture reference frame. The last 3 components specify the number of frames for each of the 3 dimensions of orientation (rotation vector) in the motion capture reference frame. This should be smaller than the **augmentNumPoints** parameter.
- **mocapToRobotLibPath** - Root directory of "Mocap to Robot" library, it should contain `initMocapToRobotLib.m`.
- **ndiSamplingPeriod** - This parameter specifies the sampling period (in seconds) of the optical tracking analysis hardware, in this case the Optotrak Certus.
- **period** - Motion capture sampling period in seconds.
- **plotAllOptDir** - Directory path for a single motion capture activity containing multiple subjects (e.g. `D:\Motion_Capture_Data\JJ`).

- **plotOptTrajFramesFile** - File path to the frames file that served as a desired operational space space trajectory for finding the optimized joint space trajectory (e.g. `D:\Motion_Capture_Data\JJ\UEK_015\UEK_015_JJ_free_01.smoothFrames.txt`.)
- **plotOptTrajJointsFile** - File path to an optimized joint space trajectory (e.g. `D:\Motion_Capture_Data\JJ\UEK_015\UEK_015_JJ_free_01.joints.txt`.)
- **plotOptTrajToolframe** - File path to the tool frame that was utilized in the optimization process (e.g. `D:\Optimization_Algorithm_Data\Toolframes\newToolframe.csv`.)
- **postAnalysisAccRbCs** - See **postAnalysisPoseRbCs**. In this case, the reference frames and rigid bodies are utilized for linear acceleration, angular acceleration, and Euler angle acceleration - in that order.
- **postAnalysisAngAccDtw** - The integer value of this parameter determines whether angular acceleration is included in the dynamic time-warping computation if the use of dynamic time-warping is specified in **postAnalysisPerformDtw**. Since angular acceleration is noisier that other kinematic variables it can skew the alignment procedure. An integer value of 1 indicates that angular acceleration should be utilized while dynamic time-warping, while an integer value of 0 indicates NOT to include angular acceleration.
- **postAnalysisComputeStats** - This parameter determines whether trajectory statistics will be computed while running a post-analysis task. An integer value of 1 indicates that trajectory statistics will be computed while an integer value of 0 indicates that trajectory statistics will NOT be computed.
- **postAnalysisCreateGraphs** - This parameter determines whether graphs will be displayed on screen during a post-analysis taks. An integer value of 1 indicates that graphs will be displayed on screen, while an integer value of 0 indicates that graphs will NOT be displayed on screen.
- **postAnalysisDataFolder** - Folder path to the motion capture data directory containing all subjects for the currently analyzed activity (e.g. `D:\Motion_Capture_Data\JJ`).
- **postAnalysisMainResultsFolder** - Folder path to directory containing multiple experimental sessions (e.g. `D:\Motion_Verification_Data\JJ`).
- **postAnalysisPerformDtw** - The integer value of this parameter determines whether dynamic time-warping is utilized to align the verification and motion capture trajectories in time. An integer value of 1 indicates the use dynamic-time warping, an integer value of 0 indicates the use of cross-correlation.

- **postAnalysisPlotDiff** - This parameter indicates whether trend plots graph the difference of the verification trajectory from the motion capture trajectory or the two trajectories are overlayed. An integer value of 1 indicates that the difference of the two trajectories is plotted against a zero baseline. An integer value of 0 indicates that the two trajectories are overlayed when plotted.
- **postAnalysisPlotPercentage** - When **postAnalysisPlotDiff** is set to display differences this parameter governs whether physical units or percentages are shown on the y-axes. An integer value of 1 indicates that percentages are shown, while an integer value of 0 indicates that physical units are shown. When **postAnalysisPlotDiff** is NOT set to display differences this parameter governs whether the statistics shown on the trend plots have physical units or percentages.
- **postAnalysisPoseRbCs** - An array of 6 strings indicating the reference frames and rigid bodies to utilize when analyzing the pose, velocity, and acceleration of the trajectory. The first and second component indicate the reference frame and rigid body for position, respectively. The third and fourth component indicate the reference frame and rigid body for the rotation vector, respectively. The fifth and sixth component indicate the reference frame and rigid body for Euler angles, respectively. The most useful reference frame and rigid body combinations are: (thorax,bone,thorax,bone,thorax,bone). The possible reference frames are: lab (motion capture), robot, and thorax. The possible rigid bodies are: hs (hemisphere) and bone (humerus).

- **postAnalysisPrintDir** - If **postAnalysisPrintGraphs** specifies that graphs should be written to disk, this parameter determines the directory where to write the resulting graphs.
- **postAnalysisPrintGraphs** - This parameter determines whether graphs will be written to disk during a post-analysis taks. An integer value of 1 indicates that graphs will be written to disk, while an integer value of 0 indicates that graphs will NOT be written to disk.
- **postAnalysisRepeatsAvail** - This parameter indicates how many repeated verification trials are available for a particular trajectory. These repeated verification trials are then time-aligned using cross-correlation and averaged in order to reduce noise in acceleration. Typically, this parameter should be 0 unless repeated verification trials were explicitly collected.
- **postAnalysisResultsFile** - File path to the summary output file of the **Process NDI Capture** task (e.g. `D:\Motion_Verification_Data\JJ\05162018\JJ_FREE_015_02_22_N11_sum.txt`).
- **postAnalysisResultsFolder** - Folder path to directory containing one experimental session (e.g. `D:\Motion_Verification_Data\JJ\05162018`).
- **postAnalysisVelRbCs** - See **postAnalysisPoseRbCs**. In this case, the reference frames and rigid bodies are utilized for linear velocity, angular velocity, and Euler angle rates - in that order.
- **processAllOptDir** - Directory path for a single motion capture activity containing multiple subjects (e.g. `D:\Motion_Capture_Data\JJ`).
- **processMocapC3DFile** - File path to the motion capture trial to utilize for comparison against the trial verification data (e.g. `D:\Motion_Capture_Data\JJ\UEK_015\UEK_015_JJ_free_01._V3D.c3d.txt`).
- **processMocapFolder** - Path to the folder containing the trial verification data as decribed in Data Organization (e.g. `D:\Motion_Verification_Data\JJ\05162018`).
- **processMocapJustGraph** - The integer value of this parameter determines whether the trial verification data is just graphed (1) or if it undergoes the full processing pipeline (0).
- **processMocapPrint** - The integer value of this parameter determines whether a PDF document containing a summary of the trial verification data processing should be created. A value of 1 indicates that the PDF document should be created, while a value of 0 indicates that the PDF document should NOT be created.
- **processMocapProgramName** - Name of the motion program in the trial verification data that is to be processed (e.g. `JJ_FREE_015_01_22_N11`).
- **processTrajOpt** - This parameter dictates how the file name of the optimized joint space trajectory is determined. Typically this should remain at **1** indicating that the file ends in `.joints.txt`. See code for other options.
- **smoothAndExportAllDir** - Directory path for a single motion capture activity containing multiple subjects (e.g. `D:\Motion_Capture_Data\JJ`).
- **smoothAndExportC3DFile** - File path to a single trial file (e.g. `D:\Motion_Capture_Data\JJ\UEK_015\UEK_015_JJ_free_01._V3D.c3d.txt`).
- **smoothAndExportFile** - File path where the smoothed trial should be output. (e.g. `D:\Motion_Capture_Data\JJ\UEK_015\UEK_015_JJ_free_01.smoothFrames.txt`).
- **smoothAndPlotAllDir** - Directory path for a single motion capture activity containing multiple subjects (e.g. `D:\Motion_Capture_Data\JJ`).
- **smoothAndPlotFile** - File path to a single trial file (e.g. `D:\Motion_Capture_Data\JJ\UEK_015\UEK_015_JJ_free_01._V3D.c3d.txt`).

- **smoothGaussInterval** - Gaussian interval (in terms of frames) to use for smoothing.
- **smoothPadLength** - Number of stationary frames to amend to beginning and end of trajectory before smoothing so as to avoid smoothing artefacts.
- **smoothPadRemoval** - Number of frames to remove once trajectory has been amended and smoothed.

- **subsampleAllDir** - Directory path for a single motion capture activity containing multiple subjects (e.g. `D:\Motion_Capture_Data\JJ`).
- **subsampleC3DFile** - File path to a single trial file (e.g. `D:\Motion_Capture_Data\JJ\UEK_015\UEK_015_JJ_free_01._V3D.c3d.txt`).
- **subsampleErrorsDir** - Directory path for a single motion capture activity containing multiple subjects (e.g. `D:\Motion_Capture_Data\JJ`).
- **subsampleEveryOther** - This parameter dictates the uniform subsampling rate. One frame out of the integer value frames dictated by this parameter are retained from the original trajectory.
- **subsamplePerAll** - This parameter dictates the non-uniform subsampling rate (as a fraction of the original trajectory).
- **subsamplePerBF** - This parameter dictates the non-uniform subsampling rate (as a fraction of the original trajectory).
- **subsamplePlotC3DFile** - File path to a single trial file (e.g. `D:\Motion_Capture_Data\JJ\UEK_015\UEK_015_JJ_free_01._V3D.c3d.txt`).
- **subsamplingMethodErrors** - The integer value of this parameter dictates whether to use uniform (0) or non-uniform (1) subsampling when computing and processing subsampling errors.
- **tickDefinition** - This parameter specifies the sampling period (in seconds) of the captured robot joint angles from the robot controller.
- **tpProgramIndicesFile** - Path to a subsampling file indicating which indices of a trajectory to utilize for subsampling. (e.g. `D:\Motion_Capture_Data\JJ\UEK_015\UEK_015_JJ_free_01.indices.txt`.)

- **tpProgramJ1** - This parameter's interpretation depends on the **tpProgramJ1Offset** parameter. This parameter can either specify to offset the angle of the first robot joint for every timepoint in the trajectory by a real number (in degrees). Or, it can specify the angle of the first robot joint for the first timepoint in the trajectory (in degrees) - the angle of the first robot joint for the remaining timepoints is offset relative to the first one.
- **tpProgramJ1Offset** - This parameter specifies whether **tpProgramJOne** should be treated as an offset angle for the first robot joint for the entire trajectory, or as a specification of the angle of the first robot joint for the first timepoint in the trajectory.
- **tpProgramJointsFile** - File path to an optimized joint space trajectory (e.g. `D:\Motion_Capture_Data\JJ\UEK_015\UEK_015_JJ_free_01.joints.txt`.)
- **tpProgramMultiplier** - A positive real number (usually greater than 1) indicating a multiplier for the average speed between two timepoints in a trajectory.

6 Task Descriptions

A task can utilize any of the three configuration files. Each task will indicate which configuration file it utilizes. The parameters listed for each task only pertain to the `parametersM20.xml` configuration file. If a task utilizes the `robotM20.xml` or the `tpProgram_humerus.xml` configuration file then it is assumed that it utilizes all of the parameters within that file.

Augment and Plot Trajectory Single

Description: This task augments the trajectory specified by **smoothAndPlotFile** with an artificial speed-up and slow-down section by utilizing a 4^{th} degree constant-jerk polynomial for both position and orientation. It then plots the amended trajectory for visual inspection.
Path: `tasks/trajectoryGeneration/augmentAndPlotTrajectorySingle.m`
Configuration Files: `parametersM20.xml`
Parameters:

- **smoothAndPlotFile** - File path to a single trial file (e.g. `D:\Motion_Capture_Data\JJ\UEK_015\UEK_015_JJ_free_01._V3D.c3d.txt`).
- **period** - Motion capture sampling period in seconds.
- **augmentNumPoints** - A 6-component vector specifying the number of frames to utilize for augmenting the trajectory at the beginning and the end. The first 3 components specify the number of frames for each of the 3 dimensions of position in the motion capture reference frame. The last 3 components specify the number of frames for each of the 3 dimensions of orientation (rotation vector) in the motion capture reference frame. This should be larger than the **interpNumPoints** parameter.
- **interpNumPoints** - A 6-component vector specifying the number of frames to utilize when interpolating the trajectory at the beginning and the end. The first 3 components specify the number of frames for each of the 3 dimensions of position in the motion capture reference frame. The last 3 components specify the number of frames for each of the 3 dimensions of orientation (rotation vector) in the motion capture reference frame. This should be smaller than the **augmentNumPoints** parameter.
- **smoothGaussInterval** - Gaussian interval (in terms of frames) to use for smoothing.

Augment and Plot Trajectory All

Description: This task augments all trajectories for the subjects contained in **smoothAndPlotAllDir** with an artificial speed-up and slow-down section by utilizing a 4^{th} degree constant-jerk polynomial for both position and orientation. It then plots the amended trajectories for visual inspection.
Path: `tasks/trajectoryGeneration/augmentAndPlotTrajectoryAll.m`
Configuration Files: `parametersM20.xml`
Parameters:

- **smoothAndPlotAllDir** - Directory path for a single motion capture activity containing multiple subjects (e.g. `D:\Motion_Capture_Data\JJ`).
- **period** - Motion capture sampling period in seconds.
- **augmentNumPoints** - A 6-component vector specifying the number of frames to utilize for augmenting the trajectory at the beginning and the end. The first 3 components specify the number of frames for each of the 3 dimensions of position in the motion capture reference frame. The last 3 components specify the number of frames for each of the 3 dimensions of orientation (rotation vector) in the motion capture reference frame. This should be larger than the **interpNumPoints** parameter.
- **interpNumPoints** - A 6-component vector specifying the number of frames to utilize when interpolating the trajectory at the beginning and the end. The first 3 components specify the number of frames for each of the 3 dimensions of position in the motion capture reference frame. The last 3 components specify the number of frames for each of the 3 dimensions of orientation (rotation vector) in the motion capture reference frame. This should be smaller than the **augmentNumPoints** parameter.
- **smoothGaussInterval** - Gaussian interval (in terms of frames) to use for smoothing.

Augment and Export Trajectory Single

Description: This task augments the trajectory specified by **smoothAndExportC3DFile** with an artificial speed-up and slow-down section by utilizing a 4^{th} degree constant-jerk polynomial for both position and orientation. It then exports the trajectory to the file specified by **smoothAndExportFile**.
Path: `tasks/trajectoryGeneration/augmentAndExportTrajectorySingle.m`
Configuration Files: `parametersM20.xml`
Parameters:

- **smoothAndExportC3DFile** - File path to a single trial file (e.g. `D:\Motion_Capture_Data\JJ\UEK_015\UEK_015_JJ_free_01._V3D.c3d.txt`).
- **smoothAndExportFile** - File path where the smoothed trial should be output. (e.g. `D:\Motion_Capture_Data\JJ\UEK_015\UEK_015_JJ_free_01.smoothFrames.txt`).
- **period** - Motion capture sampling period in seconds.
- **augmentNumPoints** - A 6-component vector specifying the number of frames to utilize for augmenting the trajectory at the beginning and the end. The first 3 components specify the number of frames for each of the 3 dimensions of position in the motion capture reference frame. The last 3 components specify the number of frames for each of the 3 dimensions of orientation (rotation vector) in the motion capture reference frame. This should be larger than the **interpNumPoints** parameter.
- **interpNumPoints** - A 6-component vector specifying the number of frames to utilize when interpolating the trajectory at the beginning and the end. The first 3 components specify the number of frames for each of the 3 dimensions of position in the motion capture reference frame. The last 3 components specify the number of frames for each of the 3 dimensions of orientation (rotation vector) in the motion capture reference frame. This should be smaller than the **augmentNumPoints** parameter.
- **smoothGaussInterval** - Gaussian interval (in terms of frames) to use for smoothing.

Augment and Export Trajectory All

Description: This task augments all trajectories for the subjects contained in **smoothAndExportAllDir** with an artificial speed-up and slow-down section by utilizing a 4^{th} degree constant-jerk polynomial for both position and orientation. It then exports each trajectory individually to a file ending in `smoothFrames.txt`.
Path: `tasks/trajectoryGeneration/augmentAndExportTrajectoryAll.m`
Configuration Files: `parametersM20.xml`
Parameters:

- **smoothAndExportAllDir** - Directory path for a single motion capture activity containing multiple subjects (e.g. `D:\Motion_Capture_Data\JJ`).
- **period** - Motion capture sampling period in seconds.
- **augmentNumPoints** - A 6-component vector specifying the number of frames to utilize for augmenting the trajectory at the beginning and the end. The first 3 components specify the number of frames for each of the 3 dimensions of position in the motion capture reference frame. The last 3 components specify the number of frames for each of the 3 dimensions of orientation (rotation vector) in the motion capture reference frame. This should be larger than the **interpNumPoints** parameter.
- **interpNumPoints** - A 6-component vector specifying the number of frames to utilize when interpolating the trajectory at the beginning and the end. The first 3 components specify the number of frames for each of the 3 dimensions of position in the motion capture reference frame. The last 3 components specify the number of frames for each of the 3 dimensions of orientation (rotation vector) in the motion capture reference frame. This should be smaller than the **augmentNumPoints** parameter.
- **smoothGaussInterval** - Gaussian interval (in terms of frames) to use for smoothing.

Smooth and Plot By File

Description: This task smooths the trajectory specified by **smoothAndPlotFile** using a Gaussian filter. It then plots the smoothed trajectory for visual inspection.
Path: `tasks/trajectoryGeneration/smoothAndPlotByFile.m`
Configuration Files: `parametersM20.xml`
Parameters:

- **smoothAndPlotFile** - File path to a single trial file (e.g. `D:\Motion_Capture_Data\JJ\UEK_015\UEK_015_JJ_free_01._V3D.c3d.txt`).
- **period** - Motion capture sampling period in seconds.
- **smoothGaussInterval** - Gaussian interval (in terms of frames) to use for smoothing.
- **smoothPadLength** - Number of stationary frames to amend to beginning and end of trajectory before smoothing so as to avoid smoothing artefacts.
- **smoothPadRemoval** - Number of frames to remove once trajectory has been amended and smoothed.

Smooth and Plot All

Description: This task smooths all trajectories for the subjects contained in **smoothAndPlotAllDir** using a Gaussian filter. It then plots the smoothed trajectories for visual inspection.
Path: `tasks/trajectoryGeneration/smoothAndPlotAll.m`
Configuration Files: `parametersM20.xml`
Parameters:

- **smoothAndPlotAllDir** - Directory path for a single motion capture activity containing multiple subjects (e.g. `D:\Motion_Capture_Data\JJ`).
- **period** - Motion capture sampling period in seconds.
- **smoothGaussInterval** - Gaussian interval (in terms of frames) to use for smoothing.
- **smoothPadLength** - Number of stationary frames to amend to beginning and end of trajectory before smoothing so as to avoid smoothing artefacts.
- **smoothPadRemoval** - Number of frames to remove once trajectory has been amended and smoothed.

Smooth and Export by File

Description: This task smooths the trajectory specified by **smoothAndExportC3DFile** using a Gaussian filter and exports it to the file specified by **smoothAndExportFile**.
Path: `tasks/trajectoryGeneration/smoothAndExportByFile.m`
Configuration Files: `parametersM20.xml`
Parameters:

- **smoothAndExportC3DFile** - File path to a single trial file (e.g. `D:\Motion_Capture_Data\JJ\UEK_015\UEK_015_JJ_free_01._V3D.c3d.txt`).
- **smoothAndExportFile** - File path where the smoothed trial should be output. (e.g. `D:\Motion_Capture_Data\JJ\UEK_015\UEK_015_JJ_free_01.smoothFrames.txt`).
- **period** - Motion capture sampling period in seconds.
- **smoothGaussInterval** - Gaussian interval (in terms of frames) to use for smoothing.

- **smoothPadLength** - Number of stationary frames to amend to beginning and end of trajectory before smoothing so as to avoid smoothing artefacts.
- **smoothPadRemoval** - Number of frames to remove once trajectory has been amended and smoothed.

Smooth and Export All

Description: This task smooths all trajectories for the subjects contained in **smoothAndPlotAllDir** using a Gaussian filter. It then exports each trajectory individually to a file ending in `smoothFrames.txt`.
Path: `tasks/trajectoryGeneration/smoothAndExportAll.m`
Configuration Files: `parametersM20.xml`
Parameters:

- **smoothAndExportAllDir** - Directory path for a single motion capture activity containing multiple subjects (e.g. `D:\Motion_Capture_Data\JJ`).
- **period** - Motion capture sampling period in seconds.
- **smoothGaussInterval** - Gaussian interval (in terms of frames) to use for smoothing.
- **smoothPadLength** - Number of stationary frames to amend to beginning and end of trajectory before smoothing so as to avoid smoothing artefacts.
- **smoothPadRemoval** - Number of frames to remove once trajectory has been amended and smoothed.

Plot Optimized Trajectory By File

Description: This task plots an optimized joint space trajectory specified by **plotOptTrajJointsFile** and verifies that it matches the desired operational space trajectory specified in **plotOptTrajFramesFile**. If the **plotOptTrajToolframe** parameter is blank then the humerus tool frame specified in `robotM20.xml` will be utilized.
Path: `tasks/trajectoryGeneration/plotOptimizedTrajByFile.m`
Configuration Files: `parametersM20.xml`, `robotM20.xml`
Parameters:

- **plotOptTrajJointsFile** - File path to an optimized joint space trajectory (e.g. `D:\Motion_Capture_Data\JJ\UEK_015\UEK_015_JJ_free_01.joints.txt`.)
- **plotOptTrajFramesFile** - File path to the frames file that served as a desired operational space space trajectory for finding the optimized joint space trajectory (e.g. `D:\Motion_Capture_Data\JJ\UEK_015\UEK_015_JJ_free_01.smoothFrames.txt`.)
- **plotOptTrajToolframe** - File path to the tool frame that was utilized in the optimization process (e.g. `D:\Optimization_Algorithm_Data\Toolframes\newToolframe.csv`.)
- **period** - Motion capture sampling period in seconds.

Plot All Optimized Trajectories

Description: This task plots all optimized joint space trajectories for the subjects specified by **plotAllOptDir** and verifies that they match the desired operational space trajectories. The name of the optimized joint space trajectory file is determined by **processTrajOpt**, although typically this will simply be a file ending in `.joints.txt`. The corresponding desired operational space trajectory is assumed to be a file ending in `.smoothFrames.txt`. If a corresponding `.toolframe.xml` file is found, the tool frame contained in this file is assumed to have been utilized for the optimization process. Otherwise, the toolframe specified in `robotM20.xml` is assumed to have been utilized for the optimization process.

Path: `tasks/trajectoryGeneration/plotAllOptimizedTraj.m`
Configuration Files: `parametersM20.xml`, `robotM20.xml`
Parameters:

- **plotAllOptDir** - Directory path for a single motion capture activity containing multiple subjects (e.g. `D:\Motion_Capture_Data\JJ`).
- **processTrajOpt** - This parameter dictates how the file name of the optimized joint space trajectory is determined. Typically this should remain at **1** indicating that the file ends in `.joints.txt`. See code for other options.
- **period** - Motion capture sampling period in seconds.

Process All Optimized Trajectories

Description: This task examines all optimized joint space trajectories for the subjects specified by **plotAllOptDir** to verify that they match the desired operational space trajectories and that joint velocity limits have been met. The name of the optimized joint space trajectory file is determined by **processTrajOpt**, although typically this will simply be a file ending in `.joints.txt`. The corresponding desired operational space trajectory is assumed to be a file ending in `.smoothFrames.txt`. If a corresponding `.toolframe.xml` file is found, the tool frame contained in this file is assumed to have been utilized for the optimization process. Otherwise, the toolframe specified in `robotM20.xml` is assumed to have been utilized for the optimization process.
Path: `tasks/trajectoryGeneration/processAllOptimizedTraj.m`
Configuration Files: `parametersM20.xml`, `robotM20.xml`
Parameters:

- **processAllOptDir** - Directory path for a single motion capture activity containing multiple subjects (e.g. `D:\Motion_Capture_Data\JJ`).
- **processTrajOpt** - This parameter dictates how the file name of the optimized joint space trajectory is determined. Typically this should remain at **1** indicating that the file ends in `.joints.txt`. See code for other options.
- **period** - Motion capture sampling period in seconds.

Subsample by File

Description: This task non-uniformly subsamples a trajectory according to the subsampling rate specified in **subsamplePerBF**. It creates a file ending in `indices.txt` in the same directory as **subsampleC3DFile**.
Path: `tasks/trajectoryGeneration/subsampleByFile.m`
Configuration Files: `parametersM20.xml`
Parameters:

- **subsampleC3DFile** - File path to a single trial file (e.g. `D:\Motion_Capture_Data\JJ\UEK_015\UEK_015_JJ_free_01._V3D.c3d.txt`).
- **subsamplePerBF** - This parameter dictates the non-uniform subsampling rate (as a fraction of the original trajectory).
- **period** - Motion capture sampling period in seconds.

Subsample Plot by File

Description: This task generates plots that aids in the visualization of a subsampled trajectory by plotting the original and subsampled trajectory simultaneously. Unlike most tasks in this software package, this task utilizes Jupyter notebook - specifically the Plotly package.
Path: `tasks/trajectoryGeneration/SubSamplePlotByFile.ipynb`

Configuration Files: `parametersM20.xml`
Parameters:

- **subsamplePlotC3DFile** - File path to a single trial file (e.g. `D:\Motion_Capture_Data\JJ\UEK_015\UEK_015_JJ_free_01._V3D.c3d.txt`).

Subsample All

Description: This task non-uniformly subsamples all trajectories for the subjects in **subsampleAllDir** according to the subsampling rate specified in **subsamplePerAll**. It creates a file ending in `indices.txt` for each of the trial files for each subject in **subsampleAllDir**.
Path: `tasks/trajectoryGeneration/subsampleAll.m`
Configuration Files: `parametersM20.xml`
Parameters:

- **subsampleAllDir** - Directory path for a single motion capture activity containing multiple subjects (e.g. `D:\Motion_Capture_Data\JJ`).
- **subsamplePerAll** - This parameter dictates the non-uniform subsampling rate (as a fraction of the original trajectory).
- **period** - Motion capture sampling period in seconds.

Subsample All Uniformly

Description: This task uniformly subsamples all trajectories for the subjects in **subsampleAllDir** according to the subsampling rate specified in **subsampleEveryOther**. It creates a file ending in `indices.txt` for each of the trial files for each subject in **subsampleAllDir**.
Path: `tasks/trajectoryGeneration/subsampleAllUniform.m`
Configuration Files: `parametersM20.xml`
Parameters:

- **subsampleAllDir** - Directory path for a single motion capture activity containing multiple subjects (e.g. `D:\Motion_Capture_Data\JJ`).
- **subsampleEveryOther** - This parameter dictates the uniform subsampling rate. One frame out of the integer value frames dictated by this parameter are retained from the original trajectory.
- **period** - Motion capture sampling period in seconds.

Compute Subsampling Errors

Description: This task computes the subsampling error for all the trajectories of the subjects in **computeSubsamplingErrorsDir**. The **subsamplingMethodErrors** parameter dictates whether the results of uniform or non-uniform subsampling are utilized. For each trajectory, this task outputs a file ending in `ssErrors.txt` or `ssErrorsUniform.txt` based on whether uniform or non-uniform subsampling was utilized, respectively, containing the subsampling error associated with every timepoint in the trajectory.
Path: `tasks/trajectoryGeneration/computeSubsamplingErrors.m`
Configuration Files: `parametersM20.xml`
Parameters:

- **computeSubsamplingErrorsDir** - Directory path for a single motion capture activity containing multiple subjects (e.g. `D:\Motion_Capture_Data\JJ`).
- **subsamplingMethodErrors** - The integer value of this parameter dictates whether to use uniform (0) or non-uniform (1) subsampling when computing and processing subsampling errors.
- **period** - Motion capture sampling period in seconds.

Process Subsampling Errors

Description: This is a convenience task that creates summary statistics upon previously computed subsampling errors for all the trajectories of the subjects in **subsampleErrorsDir**. The **subsamplingMethodErrors** parameter dictates whether the results of uniform or non-uniform subsampling are utilized.
Path: `tasks/trajectoryGeneration/processAllSubsamplingErrors.m`
Configuration Files: `parametersM20.xml`
Parameters:

- **subsampleErrorsDir** - Directory path for a single motion capture activity containing multiple subjects (e.g. `D:\Motion_Capture_Data\JJ`).
- **subsamplingMethodErrors** - The integer value of this parameter dictates whether to use uniform (0) or non-uniform (1) subsampling when computing and processing subsampling errors.

Write Teach Pendant Program

Description: This task write a FANUC motion program with speed specified in deg/sec based on the trajectory specified in **tpProgramJointsFile**. Since replicating a trajectory allows a degree of freedom about the gravitational axis, specification of the angle of the first robot joint (whose axis of rotation is coincident with the gravitational axis) has been made variable for convenience. Likewise, a parameter for multiplying the computed average speed between two timepoints is supplied since the motion program expects the maximum, not average, speed between two timepoints. This parameter must be determined empirically and varies by trajectory.
Path: `tasks/trajectoryGeneration/writeTPProgram.m`
Configuration Files: `parametersM20.xml`, `robotM20.xml`, `tpProgram_humerus.xml`
Parameters:

- **tpProgramJointsFile** - File path to an optimized joint space trajectory (e.g. `D:\Motion_Capture_Data\JJ\UEK_015\UEK_015_JJ_free_01.joints.txt`.)
- **tpProgramIndicesFile** - Path to a subsampling file indicating which indices of a trajectory to utilize for subsampling. (e.g. `D:\Motion_Capture_Data\JJ\UEK_015\UEK_015_JJ_free_01.indices.txt`.)
- **tpProgramJ1** - This parameter's interpretation depends on the **tpProgramJ1Offset** parameter. This parameter can either specify to offset the angle of the first robot joint for every timepoint in the trajectory by a real number (in degrees). Or, it can specify the angle of the first robot joint for the first timepoint in the trajectory (in degrees) - the angle of the first robot joint for the remaining timepoints is offset relative to the first one.
- **tpProgramJ1Offset** - This parameter specifies whether **tpProgramJOne** should be treated as an offset angle for the first robot joint for the entire trajectory, or as a specification of the angle of the first robot joint for the first timepoint in the trajectory.
- **tpProgramMultiplier** - A positive real number (usually greater than 1) indicating a multiplier for the average speed between two timepoints in a trajectory.

Write Linear Teach Pendant Program

Description: This task write a FANUC motion program with speed specified in mm/sec based on the trajectory specified in **tpProgramJointsFile**. Since replicating a trajectory allows a degree of freedom about the gravitational axis, specification of the angle of the first robot joint (whose axis of rotation is coincident with the gravitational axis) has been made variable for convenience. Likewise, a parameter for multiplying the computed average speed between two timepoints is supplied since the motion program expects the maximum, not average, speed between two timepoints. This parameter must be determined empirically and varies by trajectory.
Path: `tasks/trajectoryGeneration/writeTPProgramLinear.m`

Configuration Files: `parametersM20.xml`, `robotM20.xml`, `tpProgram_humerus.xml`
Parameters:

- **tpProgramJointsFile** - File path to an optimized joint space trajectory (e.g. `D:\Motion_Capture_Data\JJ\UEK_015\UEK_015_JJ_free_01.joints.txt`.)
- **tpProgramIndicesFile** - Path to a subsampling file indicating which indices of a trajectory to utilize for subsampling. (e.g. `D:\Motion_Capture_Data\JJ\UEK_015\UEK_015_JJ_free_01.indices.txt`.)
- **tpProgramJ1** - This parameter's interpretation depends on the **tpProgramJ1Offset** parameter. This parameter can either specify to offset the angle of the first robot joint for every timepoint in the trajectory by a real number (in degrees). Or, it can specify the angle of the first robot joint for the first timepoint in the trajectory (in degrees) - the angle of the first robot joint for the remaining timepoints is offset relative to the first one.
- **tpProgramJ1Offset** - This parameter specifies whether **tpProgramJOne** should be treated as an offset angle for the first robot joint for the entire trajectory, or as a specification of the angle of the first robot joint for the first timepoint in the trajectory.
- **tpProgramMultiplier** - A positive real number (usually greater than 1) indicating a multiplier for the average speed between two timepoints in a trajectory.

Establish Humerus Coordinate System

Description: This task establishes the rigid body relationship between the end-effector and the humerus. The input dataset for this task is in the format of a saved Matlab workspace and the required variables are described in the header of the task. The only reason why `parameterM20.xml` is needed is to specify the **mocapToRobotLibPath**.
Path: `tasks/opticalTrackingAnalysis/establishHumerusCS.m`
Configuration Files: `parametersM20.xml`
Parameters:

- None

NDI to Robot Transform

Description: This task processes the dataset created by the robot reference frame identification procedure and establishes the robot frame within the optical tracking frame as well as the rigid body relationship between the hemisphere and the robot end-effector. The robot reference frame identification procedure can be offset (via the teach pendant) for the first robot joint depending on the position of the optical tracking system during the experimental session. The **axisCalibrationJ1Offset** parameter specifies this offset so it can be properly accounted for during computations.
Path: `tasks/opticalTrackingAnalysis/ndiToRobotTransform.m`
Configuration Files: `parametersM20.xml`
Parameters:

- **axisCalibrationFolder** - Path to the folder containing the hemisphere trajectory during the robot reference frame identification procedure. The files produced as part of this procedure are described in Data Organization (e.g. `D:\Motion_Verification_Data\JJ\05162018`).
- **axisCalibrationJ1Offset** - This parameter specifies the offset from zero of the angle of the first robot joint. This is not to be confused with the **tpProgramJ1** configuration parameter - the two are not related and are independent of each other.
- **axisCalibrationTcp** - This parameter specifies whether to determine the tool center point of the hemisphere from the dataset generated by the robot reference frame identification procedure.
- **axisCalibrationResults** - This parameter specifies the path to a file where the results of processing the dataset of the robot reference frame identification procedure should be stored.

Graph Paths

Description: This task graphs the position of the hemisphere as the robot rotates about its end-effector without translating during the robot reference frame identification procedure. This task is implemented as a Jupyter notebook.
Path: `tasks/opticalTrackingAnalysis/GraphPaths.ipynb`
Configuration Files: `parametersM20.xml`
Parameters:

- **axisCalibrationFolder** - Path to the folder containing the hemisphere trajectory during the robot reference frame identification procedure. The files produced as part of this procedure are described in Data Organization (e.g. `D:\Motion_Verification_Data\JJ\05162018`).

Simple Joint Plot

Description: This task graphs the position of the hemisphere as the robot rotates its 1^{st}, 2^{nd}, and 4^{th} joints during the robot reference frame identification procedure. This task is implemented as a Jupyter notebook.
Path: `tasks/opticalTrackingAnalysis/SimpleJointPlot.ipynb`
Configuration Files: `parametersM20.xml`
Parameters:

- **axisCalibrationFolder** - Path to the folder containing the hemisphere trajectory during the robot reference frame identification procedure. The files produced as part of this procedure are described in Data Organization (e.g. `D:\Motion_Verification_Data\JJ\05162018`).

Process NDI Capture

Description: This task processes the trial verification data files (see Data Organization) against the motion capture trial specified by **processMocapC3DFile**. This task aligns the trajectory to be verified (achieved) against the motion capture trajectory (desired) either automatically (jumping jacks) or semi-automatically (all other activities). Based on the alignment process, it computes a velocity multiplier by which the currently programmed velocities should be multiplied. Multiple iterations of motion program generation, verification, and velocity multiplication should stabilize the velocity multiplier to approximately unity. This task outputs a file ending in `_sum.txt`, a necessary component of the post-processing pipeline, summarizing the processing results.
Path: `tasks/opticalTrackingAnalysis/processNdiCapture.m`
Configuration Files: `parametersM20.xml`, `robotM20.xml`
Parameters:

- **processMocapFolder** - Path to the folder containing the trial verification data as decribed in Data Organization (e.g. `D:\Motion_Verification_Data\JJ\05162018`).
- **processMocapProgramName** - Name of the motion program in the trial verification data that is to be processed (e.g. `JJ_FREE_015_01_22_N11`).
- **processMocapC3DFile** - File path to the motion capture trial to utilize for comparison against the trial verification data (e.g. `D:\Motion_Capture_Data\JJ\UEK_015\UEK_015_JJ_free_01._V3D.c3d.txt`).
- **butterworthCutoff** - Butterworth filter cutoff frequency (in Hz) to utilize for smoothing.
- **butterworthOrder** - Butterworth filter order to utilize for smoothing.
- **period** - Motion capture sampling period in seconds.
- **ndiSamplingPeriod** - This parameter specifies the sampling period (in seconds) of the optical tracking analysis hardware, in this case the Optotrak Certus.

- **tickDefinition** - This parameter specifies the sampling period (in seconds) of the captured robot joint angles from the robot controller.
- **processMocapPrint** - The integer value of this parameter determines whether a PDF document containing a summary of the trial verification data processing should be created. A value of 1 indicates that the PDF document should be created, while a value of 0 indicates that the PDF document should NOT be created.
- **processMocapJustGraph** - The integer value of this parameter determines whether the trial verification data is just graphed (1) or if it undergoes the full processing pipeline (0).

Check File

Description: This task provides a visual comparison of the trial verification data against the motion capture trajectory taking into account position, velocity, and acceleration.
Path: `tasks/opticalTrackingAnalysis/checkFile.m`
Configuration Files: `parametersM20.xml`, `robotM20.xml`
Parameters:

- **postAnalysisResultsFile** - File path to the summary output file of the **Process NDI Capture** task (e.g. `D:\Motion_Verification_Data\JJ\05162018\JJ_FREE_015_02_22_N11_sum.txt`).
- **postAnalysisDataFolder** - Folder path to the motion capture data directory containing all subjects for the currently analyzed activity (e.g. `D:\Motion_Capture_Data\JJ`).
- **period** - Motion capture sampling period in seconds.
- **ndiSamplingPeriod** - This parameter specifies the sampling period (in seconds) of the optical tracking analysis hardware, in this case the Optotrak Certus.
- **tickDefinition** - This parameter specifies the sampling period (in seconds) of the captured robot joint angles from the robot controller.
- **butterworthCutoff** - Butterworth filter cutoff frequency (in Hz) to utilize for smoothing.
- **butterworthOrder** - Butterworth filter order to utilize for smoothing.
- **postAnalysisPoseRbCs** - An array of 6 strings indicating the reference frames and rigid bodies to utilize when analyzing the pose, velocity, and acceleration of the trajectory. The first and second component indicate the reference frame and rigid body for position, respectively. The third and fourth component indicate the reference frame and rigid body for the rotation vector, respectively. The fifth and sixth component indicate the reference frame and rigid body for Euler angles, respectively. The most useful reference frame and rigid body combinations are: (thorax,bone,thorax,bone,thorax,bone). The possible reference frames are: lab (motion capture), robot, and thorax. The possible rigid bodies are: hs (hemisphere) and bone (humerus).
- **postAnalysisVelRbCs** - See **postAnalysisPoseRbCs**. In this case, the reference frames and rigid bodies are utilized for linear velocity, angular velocity, and Euler angle rates - in that order.
- **postAnalysisAccRbCs** - See **postAnalysisPoseRbCs**. In this case, the reference frames and rigid bodies are utilized for linear acceleration, angular acceleration, and Euler angle acceleration - in that order.
- **postAnalysisPerformDtw** - The integer value of this parameter determines whether dynamic time-warping is utilized to align the verification and motion capture trajectories in time. An integer value of 1 indicates the use dynamic-time warping, an integer value of 0 indicates the use of cross-correlation.

- **postAnalysisAngAccDtw** - The integer value of this parameter determines whether angular acceleration is included in the dynamic time-warping computation if the use of dynamic time-warping is specified in **postAnalysisPerformDtw**. Since angular acceleration is noisier that other kinematic variables it can skew the alignment procedure. An integer value of 1 indicates that angular acceleration should be utilized while dynamic time-warping, while an integer value of 0 indicates NOT to include angular acceleration.
- **postAnalysisOffsetEndpts** - This parameter indicates whether the endpoints of the trajectory should be offset according to the augmentation that was performed when calculating trajectory statistics. An integer value of 1 indicates that the endpoints should be offset, while an integer value of 0 indicates that they should not be offset. Typically this parameter should be set to 1.
- **postAnalysisRepeatsAvail** - This parameter indicates how many repeated verification trials are available for a particular trajectory. These repeated verification trials are then time-aligned using cross-correlation and averaged in order to reduce noise in acceleration. Typically, this parameter should be 0 unless repeated verification trials were explicitly collected.
- **postAnalysisPlotDiff** - This parameter indicates whether trend plots graph the difference of the verification trajectory from the motion capture trajectory or the two trajectories are overlayed. An integer value of 1 indicates that the difference of the two trajectories is plotted against a zero baseline. An integer value of 0 indicates that the two trajectories are overlayed when plotted.
- **postAnalysisPlotPercentage** - When **postAnalysisPlotDiff** is set to display differences this parameter governs whether physical units or percentages are shown on the y-axes. An integer value of 1 indicates that percentages are shown, while an integer value of 0 indicates that physical units are shown. When **postAnalysisPlotDiff** is NOT set to display differences this parameter governs whether the statistics shown on the trend plots have physical units or percentages.
- **postAnalysisMonitorNum** - This parameter indicates the monitor number where the graphs should be generated.

Check Multiple Folders

Description: This task is analogous to the **Check File** task but runs over multiple experimental sessions.
Path: `tasks/opticalTrackingAnalysis/checkMultipleFolders.m`
Configuration Files: `parametersM20.xml`, `robotM20.xml`
Parameters:

- **postAnalysisMainResultsFolder** - Folder path to directory containing multiple experimental sessions (e.g. `D:\Motion_Verification_Data\JJ`).
- **postAnalysisDataFolder** - Folder path to the motion capture data directory containing all subjects for the currently analyzed activity (e.g. `D:\Motion_Capture_Data\JJ`).
- **period** - Motion capture sampling period in seconds.
- **ndiSamplingPeriod** - This parameter specifies the sampling period (in seconds) of the optical tracking analysis hardware, in this case the Optotrak Certus.
- **tickDefinition** - This parameter specifies the sampling period (in seconds) of the captured robot joint angles from the robot controller.
- **butterworthCutoff** - Butterworth filter cutoff frequency (in Hz) to utilize for smoothing.
- **butterworthOrder** - Butterworth filter order to utilize for smoothing.

- **postAnalysisPoseRbCs** - An array of 6 strings indicating the reference frames and rigid bodies to utilize when analyzing the pose, velocity, and acceleration of the trajectory. The first and second component indicate the reference frame and rigid body for position, respectively. The third and fourth component indicate the reference frame and rigid body for the rotation vector, respectively. The fifth and sixth component indicate the reference frame and rigid body for Euler angles, respectively. The most useful reference frame and rigid body combinations are: (thorax,bone,thorax,bone,thorax,bone). The possible reference frames are: lab (motion capture), robot, and thorax. The possible rigid bodies are: hs (hemisphere) and bone (humerus).
- **postAnalysisVelRbCs** - See **postAnalysisPoseRbCs**. In this case, the reference frames and rigid bodies are utilized for linear velocity, angular velocity, and Euler angle rates - in that order.
- **postAnalysisAccRbCs** - See **postAnalysisPoseRbCs**. In this case, the reference frames and rigid bodies are utilized for linear acceleration, angular acceleration, and Euler angle acceleration - in that order.
- **postAnalysisPerformDtw** - The integer value of this parameter determines whether dynamic time-warping is utilized to align the verification and motion capture trajectories in time. An integer value of 1 indicates the use dynamic-time warping, an integer value of 0 indicates the use of cross-correlation.
- **postAnalysisAngAccDtw** - The integer value of this parameter determines whether angular acceleration is included in the dynamic time-warping computation if the use of dynamic time-warping is specified in **postAnalysisPerformDtw**. Since angular acceleration is noisier that other kinematic variables it can skew the alignment procedure. An integer value of 1 indicates that angular acceleration should be utilized while dynamic time-warping, while an integer value of 0 indicates NOT to include angular acceleration.
- **postAnalysisOffsetEndpts** - This parameter indicates whether the endpoints of the trajectory should be offset according to the augmentation that was performed when calculating trajectory statistics. An integer value of 1 indicates that the endpoints should be offset, while an integer value of 0 indicates that they should not be offset. Typically this parameter should be set to 1.
- **postAnalysisRepeatsAvail** - This parameter indicates how many repeated verification trials are available for a particular trajectory. These repeated verification trials are then time-aligned using cross-correlation and averaged in order to reduce noise in acceleration. Typically, this parameter should be 0 unless repeated verification trials were explicitly collected.
- **postAnalysisPlotDiff** - This parameter indicates whether trend plots graph the difference of the verification trajectory from the motion capture trajectory or the two trajectories are overlayed. An integer value of 1 indicates that the difference of the two trajectories is plotted against a zero baseline. An integer value of 0 indicates that the two trajectories are overlayed when plotted.
- **postAnalysisPlotPercentage** - When **postAnalysisPlotDiff** is set to display differences this parameter governs whether physical units or percentages are shown on the y-axes. An integer value of 1 indicates that percentages are shown, while an integer value of 0 indicates that physical units are shown. When **postAnalysisPlotDiff** is NOT set to display differences this parameter governs whether the statistics shown on the trend plots have physical units or percentages.
- **postAnalysisMonitorNum** - This parameter indicates the monitor number where the graphs should be generated.

Check Joints

Description: This task provides a visual comparison of the joint angles, velocities, and acceleration as captured from the robot controller (actual) vs the optimized joint space trajectory (desired).
Path: `tasks/opticalTrackingAnalysis/checkJoints.m`
Configuration Files: `parametersM20.xml`, `robotM20.xml`
Parameters:

- **postAnalysisResultsFile** - File path to the summary output file of the **Process NDI Capture** task (e.g. `D:\Motion_Verification_Data\JJ\05162018\JJ_FREE_015_02_22_N11_sum.txt`).
- **postAnalysisDataFolder** - Folder path to the motion capture data directory containing all subjects for the currently analyzed activity (e.g. `D:\Motion_Capture_Data\JJ`).
- **period** - Motion capture sampling period in seconds.
- **tickDefinition** - This parameter specifies the sampling period (in seconds) of the captured robot joint angles from the robot controller.
- **postAnalysisMonitorNum** - This parameter indicates the monitor number where the graphs should be generated.

Post Analyze File

Description: This task provides similar functionality to the **Check File** task. The major difference is that the resulting graphs can also be written to disk for later analysis.
Path: `tasks/opticalTrackingAnalysis/postAnalysisFile.m`
Configuration Files: `parametersM20.xml`, `robotM20.xml`
Parameters:

- **postAnalysisResultsFile** - File path to the summary output file of the **Process NDI Capture** task (e.g. `D:\Motion_Verification_Data\JJ\05162018\JJ_FREE_015_02_22_N11_sum.txt`).
- **postAnalysisDataFolder** - Folder path to the motion capture data directory containing all subjects for the currently analyzed activity (e.g. `D:\Motion_Capture_Data\JJ`).
- **period** - Motion capture sampling period in seconds.
- **ndiSamplingPeriod** - This parameter specifies the sampling period (in seconds) of the optical tracking analysis hardware, in this case the Optotrak Certus.
- **tickDefinition** - This parameter specifies the sampling period (in seconds) of the captured robot joint angles from the robot controller.
- **butterworthCutoff** - Butterworth filter cutoff frequency (in Hz) to utilize for smoothing.
- **butterworthOrder** - Butterworth filter order to utilize for smoothing.
- **postAnalysisPoseRbCs** - An array of 6 strings indicating the reference frames and rigid bodies to utilize when analyzing the pose, velocity, and acceleration of the trajectory. The first and second component indicate the reference frame and rigid body for position, respectively. The third and fourth component indicate the reference frame and rigid body for the rotation vector, respectively. The fifth and sixth component indicate the reference frame and rigid body for Euler angles, respectively. The most useful reference frame and rigid body combinations are: (thorax,bone,thorax,bone,thorax,bone). The possible reference frames are: lab (motion capture), robot, and thorax. The possible rigid bodies are: hs (hemisphere) and bone (humerus).
- **postAnalysisVelRbCs** - See **postAnalysisPoseRbCs**. In this case, the reference frames and rigid bodies are utilized for linear velocity, angular velocity, and Euler angle rates - in that order.
- **postAnalysisAccRbCs** - See **postAnalysisPoseRbCs**. In this case, the reference frames and rigid bodies are utilized for linear acceleration, angular acceleration, and Euler angle acceleration - in that order.
- **postAnalysisPerformDtw** - The integer value of this parameter determines whether dynamic time-warping is utilized to align the verification and motion capture trajectories in time. An integer value of 1 indicates the use dynamic-time warping, an integer value of 0 indicates the use of cross-correlation.

- **postAnalysisAngAccDtw** - The integer value of this parameter determines whether angular acceleration is included in the dynamic time-warping computation if the use of dynamic time-warping is specified in **postAnalysisPerformDtw**. Since angular acceleration is noisier that other kinematic variables it can skew the alignment procedure. An integer value of 1 indicates that angular acceleration should be utilized while dynamic time-warping, while an integer value of 0 indicates NOT to include angular acceleration.

- **postAnalysisOffsetEndpts** - This parameter indicates whether the endpoints of the trajectory should be offset according to the augmentation that was performed when calculating trajectory statistics. An integer value of 1 indicates that the endpoints should be offset, while an integer value of 0 indicates that they should not be offset. Typically this parameter should be set to 1.

- **postAnalysisRepeatsAvail** - This parameter indicates how many repeated verification trials are available for a particular trajectory. These repeated verification trials are then time-aligned using cross-correlation and averaged in order to reduce noise in acceleration. Typically, this parameter should be 0 unless repeated verification trials were explicitly collected.

- **postAnalysisCreateGraphs** - This parameter determines whether graphs will be displayed on screen during a post-analysis taks. An integer value of 1 indicates that graphs will be displayed on screen, while an integer value of 0 indicates that graphs will NOT be displayed on screen.

- **postAnalysisPrintGraphs** - This parameter determines whether graphs will be written to disk during a post-analysis taks. An integer value of 1 indicates that graphs will be written to disk, while an integer value of 0 indicates that graphs will NOT be written to disk.

- **postAnalysisComputeStats** - This parameter determines whether trajectory statistics will be computed while running a post-analysis task. An integer value of 1 indicates that trajectory statistics will be computed while an integer value of 0 indicates that trajectory statistics will NOT be computed.

- **postAnalysisPlotDiff** - This parameter indicates whether trend plots graph the difference of the verification trajectory from the motion capture trajectory or the two trajectories are overlayed. An integer value of 1 indicates that the difference of the two trajectories is plotted against a zero baseline. An integer value of 0 indicates that the two trajectories are overlayed when plotted.

- **postAnalysisPlotPercentage** - When **postAnalysisPlotDiff** is set to display differences this parameter governs whether physical units or percentages are shown on the y-axes. An integer value of 1 indicates that percentages are shown, while an integer value of 0 indicates that physical units are shown. When **postAnalysisPlotDiff** is NOT set to display differences this parameter governs whether the statistics shown on the trend plots have physical units or percentages.

- **postAnalysisPrintDir** - If **postAnalysisPrintGraphs** specifies that graphs should be written to disk, this parameter determines the directory where to write the resulting graphs.

Post Analyze Single Folder

Description: This task is similar to the **Post Analyze File** task but it analyzes all trial verification data in a single folder, i.e. an experimental session.
Path: `tasks/opticalTrackingAnalysis/postAnalysisSingleFolder.m`
Configuration Files: `parametersM20.xml`, `robotM20.xml`
Parameters:

- **postAnalysisResultsFolder** - Folder path to directory containing one experimental session (e.g. `D:\Motion_Verification_Data\JJ\05162018`).

- **postAnalysisDataFolder** - Folder path to the motion capture data directory containing all subjects for the currently analyzed activity (e.g. `D:\Motion_Capture_Data\JJ`).

- **period** - Motion capture sampling period in seconds.

- **ndiSamplingPeriod** - This parameter specifies the sampling period (in seconds) of the optical tracking analysis hardware, in this case the Optotrak Certus.
- **tickDefinition** - This parameter specifies the sampling period (in seconds) of the captured robot joint angles from the robot controller.
- **butterworthCutoff** - Butterworth filter cutoff frequency (in Hz) to utilize for smoothing.
- **butterworthOrder** - Butterworth filter order to utilize for smoothing.
- **postAnalysisPoseRbCs** - An array of 6 strings indicating the reference frames and rigid bodies to utilize when analyzing the pose, velocity, and acceleration of the trajectory. The first and second component indicate the reference frame and rigid body for position, respectively. The third and fourth component indicate the reference frame and rigid body for the rotation vector, respectively. The fifth and sixth component indicate the reference frame and rigid body for Euler angles, respectively. The most useful reference frame and rigid body combinations are: (thorax,bone,thorax,bone,thorax,bone). The possible reference frames are: lab (motion capture), robot, and thorax. The possible rigid bodies are: hs (hemisphere) and bone (humerus).
- **postAnalysisVelRbCs** - See **postAnalysisPoseRbCs**. In this case, the reference frames and rigid bodies are utilized for linear velocity, angular velocity, and Euler angle rates - in that order.
- **postAnalysisAccRbCs** - See **postAnalysisPoseRbCs**. In this case, the reference frames and rigid bodies are utilized for linear acceleration, angular acceleration, and Euler angle acceleration - in that order.
- **postAnalysisPerformDtw** - The integer value of this parameter determines whether dynamic time-warping is utilized to align the verification and motion capture trajectories in time. An integer value of 1 indicates the use dynamic-time warping, an integer value of 0 indicates the use of cross-correlation.
- **postAnalysisAngAccDtw** - The integer value of this parameter determines whether angular acceleration is included in the dynamic time-warping computation if the use of dynamic time-warping is specified in **postAnalysisPerformDtw**. Since angular acceleration is noisier that other kinematic variables it can skew the alignment procedure. An integer value of 1 indicates that angular acceleration should be utilized while dynamic time-warping, while an integer value of 0 indicates NOT to include angular acceleration.
- **postAnalysisOffsetEndpts** - This parameter indicates whether the endpoints of the trajectory should be offset according to the augmentation that was performed when calculating trajectory statistics. An integer value of 1 indicates that the endpoints should be offset, while an integer value of 0 indicates that they should not be offset. Typically this parameter should be set to 1.
- **postAnalysisRepeatsAvail** - This parameter indicates how many repeated verification trials are available for a particular trajectory. These repeated verification trials are then time-aligned using cross-correlation and averaged in order to reduce noise in acceleration. Typically, this parameter should be 0 unless repeated verification trials were explicitly collected.
- **postAnalysisCreateGraphs** - This parameter determines whether graphs will be displayed on screen during a post-analysis taks. An integer value of 1 indicates that graphs will be displayed on screen, while an integer value of 0 indicates that graphs will NOT be displayed on screen.
- **postAnalysisPrintGraphs** - This parameter determines whether graphs will be written to disk during a post-analysis taks. An integer value of 1 indicates that graphs will be written to disk, while an integer value of 0 indicates that graphs will NOT be written to disk.
- **postAnalysisComputeStats** - This parameter determines whether trajectory statistics will be computed while running a post-analysis task. An integer value of 1 indicates that trajectory statistics will be computed while an integer value of 0 indicates that trajectory statistics will NOT be computed.

- **postAnalysisPlotDiff** - This parameter indicates whether trend plots graph the difference of the verification trajectory from the motion capture trajectory or the two trajectories are overlayed. An integer value of 1 indicates that the difference of the two trajectories is plotted against a zero baseline. An integer value of 0 indicates that the two trajectories are overlayed when plotted.
- **postAnalysisPlotPercentage** - When **postAnalysisPlotDiff** is set to display differences this parameter governs whether physical units or percentages are shown on the y-axes. An integer value of 1 indicates that percentages are shown, while an integer value of 0 indicates that physical units are shown. When **postAnalysisPlotDiff** is NOT set to display differences this parameter governs whether the statistics shown on the trend plots have physical units or percentages.
- **postAnalysisPrintDir** - If **postAnalysisPrintGraphs** specifies that graphs should be written to disk, this parameter determines the directory where to write the resulting graphs.

Post Analyze Multiple Folders

Description: This task is similar to the **Post Analyze File** task but it analyzes all trial verification data for multiple folders, i.e. multiple experimental sessions.
Path: `tasks/opticalTrackingAnalysis/postAnalysisMultipleFolders.m`
Configuration Files: `parametersM20.xml`, `robotM20.xml`
Parameters:

- **postAnalysisMainResultsFolder** - Folder path to directory containing multiple experimental sessions (e.g. `D:\Motion_Verification_Data\JJ`).
- **postAnalysisDataFolder** - Folder path to the motion capture data directory containing all subjects for the currently analyzed activity (e.g. `D:\Motion_Capture_Data\JJ`).
- **period** - Motion capture sampling period in seconds.
- **ndiSamplingPeriod** - This parameter specifies the sampling period (in seconds) of the optical tracking analysis hardware, in this case the Optotrak Certus.
- **tickDefinition** - This parameter specifies the sampling period (in seconds) of the captured robot joint angles from the robot controller.
- **butterworthCutoff** - Butterworth filter cutoff frequency (in Hz) to utilize for smoothing.
- **butterworthOrder** - Butterworth filter order to utilize for smoothing.
- **postAnalysisPoseRbCs** - An array of 6 strings indicating the reference frames and rigid bodies to utilize when analyzing the pose, velocity, and acceleration of the trajectory. The first and second component indicate the reference frame and rigid body for position, respectively. The third and fourth component indicate the reference frame and rigid body for the rotation vector, respectively. The fifth and sixth component indicate the reference frame and rigid body for Euler angles, respectively. The most useful reference frame and rigid body combinations are: (thorax,bone,thorax,bone,thorax,bone). The possible reference frames are: lab (motion capture), robot, and thorax. The possible rigid bodies are: hs (hemisphere) and bone (humerus).
- **postAnalysisVelRbCs** - See **postAnalysisPoseRbCs**. In this case, the reference frames and rigid bodies are utilized for linear velocity, angular velocity, and Euler angle rates - in that order.
- **postAnalysisAccRbCs** - See **postAnalysisPoseRbCs**. In this case, the reference frames and rigid bodies are utilized for linear acceleration, angular acceleration, and Euler angle acceleration - in that order.
- **postAnalysisPerformDtw** - The integer value of this parameter determines whether dynamic time-warping is utilized to align the verification and motion capture trajectories in time. An integer value of 1 indicates the use dynamic-time warping, an integer value of 0 indicates the use of cross-correlation.

- **postAnalysisAngAccDtw** - The integer value of this parameter determines whether angular acceleration is included in the dynamic time-warping computation if the use of dynamic time-warping is specified in **postAnalysisPerformDtw**. Since angular acceleration is noisier that other kinematic variables it can skew the alignment procedure. An integer value of 1 indicates that angular acceleration should be utilized while dynamic time-warping, while an integer value of 0 indicates NOT to include angular acceleration.

- **postAnalysisOffsetEndpts** - This parameter indicates whether the endpoints of the trajectory should be offset according to the augmentation that was performed when calculating trajectory statistics. An integer value of 1 indicates that the endpoints should be offset, while an integer value of 0 indicates that they should not be offset. Typically this parameter should be set to 1.

- **postAnalysisRepeatsAvail** - This parameter indicates how many repeated verification trials are available for a particular trajectory. These repeated verification trials are then time-aligned using cross-correlation and averaged in order to reduce noise in acceleration. Typically, this parameter should be 0 unless repeated verification trials were explicitly collected.

- **postAnalysisCreateGraphs** - This parameter determines whether graphs will be displayed on screen during a post-analysis taks. An integer value of 1 indicates that graphs will be displayed on screen, while an integer value of 0 indicates that graphs will NOT be displayed on screen.

- **postAnalysisPrintGraphs** - This parameter determines whether graphs will be written to disk during a post-analysis taks. An integer value of 1 indicates that graphs will be written to disk, while an integer value of 0 indicates that graphs will NOT be written to disk.

- **postAnalysisComputeStats** - This parameter determines whether trajectory statistics will be computed while running a post-analysis task. An integer value of 1 indicates that trajectory statistics will be computed while an integer value of 0 indicates that trajectory statistics will NOT be computed.

- **postAnalysisPlotDiff** - This parameter indicates whether trend plots graph the difference of the verification trajectory from the motion capture trajectory or the two trajectories are overlayed. An integer value of 1 indicates that the difference of the two trajectories is plotted against a zero baseline. An integer value of 0 indicates that the two trajectories are overlayed when plotted.

- **postAnalysisPlotPercentage** - When **postAnalysisPlotDiff** is set to display differences this parameter governs whether physical units or percentages are shown on the y-axes. An integer value of 1 indicates that percentages are shown, while an integer value of 0 indicates that physical units are shown. When **postAnalysisPlotDiff** is NOT set to display differences this parameter governs whether the statistics shown on the trend plots have physical units or percentages.

- **postAnalysisPrintDir** - If **postAnalysisPrintGraphs** specifies that graphs should be written to disk, this parameter determines the directory where to write the resulting graphs.

Batch Process All Data

Description: This task batch processes all trial verification data creating individual and summary statistics, as well as trend plots for pose, velocity, and acceleration.
Path: `tasks/opticalTrackingAnalysis/batchProcessAllData.m`
Configuration Files: `parametersM20.xml`, `robotM20.xml`
Parameters:

- **batchProcessPrintDir** - Directory where graphs for batch processing are created.

- **batchProcessJJ** - This parameter determines whether jumping jacks will be batch processed. An integer value of 1 indicates that jumping jacks will be batch processed, while an integer value of 0 indicates that jumping jacks will NOT be batch processed.

- **batchProcessJL** - This parameter determines whether jug lifts will be batch processed. An integer value of 1 indicates that jug lifts will be batch processed, while an integer value of 0 indicates that jug lifts will NOT be batch processed.
- **batchProcessIR** - This parameter determines whether internal rotation trials will be batch processed. An integer value of 1 indicates that internal rotation trials will be batch processed, while an integer value of 0 indicates that internal rotation trials will NOT be batch processed.
- **batchProcessJO** - This parameter determines whether jogging trials will be batch processed. An integer value of 1 indicates that jogging trials will be batch processed, while an integer value of 0 indicates that jogging trials will NOT be batch processed.
- **batchProcessJJMocapData** - Path to the directory for the jumping jacks motion capture data for all subjects (e.g. `D:\Motion_Capture_Data\JJ`).
- **batchProcessJLMocapData** - Path to the directory for the jug lifts motion capture data for all subjects (e.g. `D:\Motion_Capture_Data\JL`).
- **batchProcessJOMocapData** - Path to the directory for the jogging motion capture data for all subjects (e.g. `D:\Motion_Capture_Data\JO`).
- **batchProcessIRMocapData** - Path to the directory for the internal rotation motion capture data for all subjects (e.g. `D:\Motion_Capture_Data\IR`).
- **batchProcessJJVerificationData** - Path to the directory for the jumping jacks robot motion verification data for all subjects (e.g. `D:\Motion_Verification_Data\JJ`).
- **batchProcessJLVerificationData** - Path to the directory for the jug lifts robot motion verification data for all subjects (e.g. `D:\Motion_Verification_Data\JL`).
- **batchProcessJOVerificationData** - Path to the directory for the jogging robot motion verification data for all subjects (e.g. `D:\Motion_Verification_Data\JO`).
- **batchProcessIRVerificationData** - Path to the directory for the internal rotation robot motion verification data for all subjects (e.g. `D:\Motion_Verification_Data\IR`).
- **period** - Motion capture sampling period in seconds.
- **ndiSamplingPeriod** - This parameter specifies the sampling period (in seconds) of the optical tracking analysis hardware, in this case the Optotrak Certus.
- **tickDefinition** - This parameter specifies the sampling period (in seconds) of the captured robot joint angles from the robot controller.
- **butterworthCutoff** - Butterworth filter cutoff frequency (in Hz) to utilize for smoothing.
- **butterworthOrder** - Butterworth filter order to utilize for smoothing.
- **postAnalysisPoseRbCs** - An array of 6 strings indicating the reference frames and rigid bodies to utilize when analyzing the pose, velocity, and acceleration of the trajectory. The first and second component indicate the reference frame and rigid body for position, respectively. The third and fourth component indicate the reference frame and rigid body for the rotation vector, respectively. The fifth and sixth component indicate the reference frame and rigid body for Euler angles, respectively. The most useful reference frame and rigid body combinations are: (thorax,bone,thorax,bone,thorax,bone). The possible reference frames are: lab (motion capture), robot, and thorax. The possible rigid bodies are: hs (hemisphere) and bone (humerus).
- **postAnalysisVelRbCs** - See **postAnalysisPoseRbCs**. In this case, the reference frames and rigid bodies are utilized for linear velocity, angular velocity, and Euler angle rates - in that order.

- **postAnalysisAccRbCs** - See **postAnalysisPoseRbCs**. In this case, the reference frames and rigid bodies are utilized for linear acceleration, angular acceleration, and Euler angle acceleration - in that order.
- **postAnalysisOffsetEndpts** - This parameter indicates whether the endpoints of the trajectory should be offset according to the augmentation that was performed when calculating trajectory statistics. An integer value of 1 indicates that the endpoints should be offset, while an integer value of 0 indicates that they should not be offset. Typically this parameter should be set to 1.