

simVITRO Musculoskeletal Simulator File Organization, Contents, and Conventions Specification

Rev 01

Table of Contents

1	Revision History	2
2	Purpose.....	2
3	Directory Definitions	3
3.1	Drive	3
3.2	Test Cell Directory.....	4
3.3	Project Directory	5
3.4	Specimen Directory	6
3.5	Experiment Run Directory	7
3.6	Subcategory Directory	8
4	File Types.....	9
4.1	Project Configuration Files	9
4.2	Specimen Configuration Files.....	11
4.3	Experiment Run Configuration Files	12
4.4	Trajectory Files	13
4.5	Summary Files	16
4.6	Log Files	17
4.7	Experiment Run Files	18
4.8	Optimized Files	20
4.9	Zip Files	21
5	File Details	22
5.1	Tendon Actuator Configuration	22
5.2	Force Plate Configuration	23
5.3	Bone Strain Configuration	24
5.4	Experiment Setup.....	25
5.5	Trajectory Set.....	26
5.6	Ground/Tibia Motion Trajectory	27
5.7	Ground Reaction Forces Trajectory.....	29
5.8	Tendon Actuator Trajectory.....	31
5.9	Summary File.....	32
5.10	Main Data File	33
5.11	Strain Data File	38
5.12	Processed Main Data File	40
5.13	Optimized Ground/Tibia Motion Trajectory File	41
5.14	Optimized Ground Reaction Forces Trajectory File.....	42
5.15	Optimized Tendon Actuator Trajectory File.....	43
5.16	Experiment Zip File	44
5.17	Message Log File.....	45
5.18	Processed Strain Data File	46
5.19	Control Trajectory.....	47
	Appendix I. REFERENCES	48

1 Revision History

Revision	Date	Name	Comment
01	11-25-13	R. Colbrunn	Created from Rev 07 UMS document. Incomplete.

2 Purpose

This document outlines the file naming and storage conventions for the simVITRO Musculoskeletal Simulator system. It also includes all the details about what data is stored in what file, how files are formatted, what portions of files are considered header, etc. Design intent is noted as appropriate.

The simVITRO software, as far as the fundamental functionality and user interface is concerned, is designed for musculoskeletal simulation of human and animal joints for biomechanical research. However, any controlled mechatronic system can typically be broken up into several fundamental building blocks: Sensor, States, Controllers, and Actuators. Sensor data can be used to compute states. State data can be used as an input to a controller. The controller output is sent to an actuator whose influence on the system is likely to change sensor output, closing the loop. These classes, along with a data class covering the collection of data from these sources, are combined into a stand alone software package called the Sensor-State-Controller-Actuator-Data (SSCAD) toolkit. Configuration files for each of these elements are needed and the contents are generally outlined herein.

Other simVITRO files include trajectory, summary, parameter, setup, log, and data files.

All open-ended issues are colored in red.

3 Directory Definitions

3.1 Drive

Path = *Drive*

This is the drive letter to which the information is to be saved or loaded from. Typically this would be C but should be able to be configured.

3.2 Test Cell Directory

Path = *Drive\Test Cell*

It is the root folder in which all experiment data will reside.

Test Cell is unique to the machine or test cell in which research being performed at any given time. In the case of the simVITRO product line this is the default:

- simVITRO

3.3 Project Directory

Path = *Drive\Test Cell\Project Identifier*

It is the folder in which all experiment data for a given project done in that particular test cell will reside.

Project Identifier is unique to the research being performed at any given time. It is generally advisable to use a project unique identifier rather than a testing date or general anatomy. If your general lab practices don't involve the use of project numbers, consider adding this functionality as it will greatly help in organizing your data.

Examples of project identifiers are:

- RB-0021-01 (*Lab Project Number*)
- Knee Replacement Development
- Shoulder – Hill Sachs Defect

3.4 Specimen Directory

Path = Drive\Test Cell\Project Identifier\Specimen Number

It is the folder in which all experiment data for a given project specimen in that particular test cell will reside.

Specimen Number is unique to the specimen under test at any given time. A system should be developed to keep track of specimens to ensure all numbers are unique and that all specimens are marked.

3.5 Experiment Run Directory

Path = *Drive\Test Cell\Project Identifier\Specimen Number\Experiment Run Number_ Experiment Run Name*

It is the folder in which all experiment data for a given run of a given project specimen in that particular test cell will reside.

Experiment Run Number is unique to the specimen under test at any given time. These numbers will be automatically generated such that there will be no duplicates for a given specimen.

The *Experiment Run Name* does not need to be unique and should be a few word summary of the given run. This name will allow for easy human sorting of data files as needed. For example, the run name could be Run, Jump, Normal Gait, Pathological Gait, etc.

3.6 Subcategory Directory

Path = *Directory/SubCategory*

A subcategory is a subdirectory of any of the Test Cell, Project, Specimen, or Experiment Run directories. The following are the names of the Subcategory directories.

Configuration – Configuration data includes what ever is needed to define the configuration of the particular category.

Data – Data includes any information that is related to sensor data. This is typically used for an actual experiment run. A rule of thumb is that anything coming in the DAQ board should be stored here.

For example:

The data files for given run will be stored under the following path.

Path = *Drive\Test Cell\Project Identifier\Specimen Number\Experiment Run Number_ Experiment Run Name\Data*

The configuration files for a given specimen will be stored under the following path.

Path = *Drive\Test Cell\Project Identifier\Specimen Number\Configuration*

A subcategory directory is not required for the storage of all data. In these cases, the files could be located at the root. Path = *Drive\Test Cell\Project Identifier\Specimen Number\Experiment Run Number_ Experiment Run Name*

4 File Types

4.1 Project Configuration Files

The simVITRO software, as far as the fundamental functionality and user interface is concerned, is designed for musculoskeletal simulation of human and animal joints for biomechanical research. However, any controlled mechatronic system can typically be broken up into several fundamental building blocks: Sensor, States, Controllers, and Actuators. Sensor data can be used to compute states. State data can be used as an input to a controller. The controller output is sent to an actuator whose influence on the system is likely to change sensor output, closing the loop. These classes, along with a data class covering the collection of data from these sources, are combined into a stand alone software package called the Sensor-State-Controller-Actuator-Data (SSCAD) toolkit.

There are several types of configuration files. These files are essentially a reusable parameter set that can be saved and loaded to allow for quick and repeatable setup of the experiment.

Some configuration files will exist for the duration of the project execution and others will vary based on the specimen, or experiment run.

Files that would typically remain constant for the duration of the project execution are:

- Tendon Actuator Configuration
- Force Plate Configuration
- Bone Strain Gauge Configuration

The default versions of these files will reside at Path = *Drive\Test Cell\Configuration*.

The project specific versions of these files will reside at Path = *Drive\Test Cell\Project Identifier\Configuration*.

Over time, other custom configurations may be developed and saved in these directories. The thing to remember is that the files defined in this document are the “working” files and will be programmatically overwritten during the experiment run. Unique configurations should be stored under unique names.

Loading:

The project specific versions of the file will load when the project identifier is entered. When the program is first started or if there is not a project specific one available, the default files will be loaded.

Saving:

By default, when the save functions are called the files will be overwritten in this configuration directory. If the Project Identifier field is blank, it will overwrite the non-project specific ‘default’ files.

4.2 Sensor Configuration File

Part of the Sensor-State-Controller-Actuator (SSCAD) toolkit, the sensor class is a hardware abstraction layer that has a sensor manager to allow the user to configure a number of possible sensors. For example, the user can choose from analog sensors (single channel, 3 channel, 6 channel, or 2D grid) based configurations. They can also choose to acquire data from digitized sensors. Digitized sensors are defined as sensors whose data does not come in from the Analog to Digital (A/D) board, but whose values are streamed across a network or from other connected peripheral hardware. Digitized sensors require drivers, but those drivers can be wrapped in the hardware abstraction layer. The drivers can also be made to interface with the simVITRO or any other high level software using the SSCAD toolkit. The current version of the toolkit allows for the configuration and acquisition of single and multi-channel force, position, and pressure sensors. All the data for a sensor configuration file is stored in a human readable ‘.ini’ type format.

4.3 State Configuration File

A state is a system variable used to quantify a unique measurement of any particular system. The state class is part of the SSCAD toolkit and builds system states based on the output of sensors or other states. These states can be the loads transformed to any joint coordinate system (JCS), or they can be relative spatial relationships. The states can also use input from multiple sensors for redundancy and error checking, if desired. The simVITRO software requires two states to operate, the JCS kinematics based on the robot position and the JCS loads based on the primary 6-channel load cell. Other than that, the user can implement states from the pre-built templates, or they can code their own states. The system software is able to compute and record the state data for any additionally configured states. An example of a state template would be the 6-DOF kinematic relationship for each FSU for a spine test.

By placing motion tracking sensors in each vertebra and selecting the state template from the spine module that computes FSU kinematics, the system will direct the user to digitize the proper anatomical landmarks to compute the proper coordinate systems. Then, during the experiment, the motion tracking sensor data is used to compute the kinematic state of the FSU. The state can also be displayed on the screen and written to a data file. In a similar manner, multiple states can be created for measuring level-by-level spinal kinematics with each one directed at a different spinal level (C4-C5, C5-C6, C6-C7).

States can also be combined with sensors to calculate other states. If the load cell is moving relative to the gravity field, gravity compensation is required to eliminate gravity based sensor cross-talk. Another state template designed for this class is the 6-DOF load class. This class uses the robot position sensor, the load cell sensor, and the state that contains the relative spatial relationship between the load cell and the anatomical reference frame that the loads are controlled in. The information from the sensors and the position state, also built from a template, is combined to compensate for gravity and transform the loads to the anatomical reference frame. This demonstrates how states can be combined while increasing flexibility and minimizing the recalculation of variables.

The current version of the toolkit allows for the configuration of all licensed simVITRO anatomical modules. All the data for a state configuration file is stored in a human readable '.ini' type format. If a configuration file contains data for a module that is not installed, it will identify the error.

4.4 Control Configuration File

The controller class provides a means by which one or more states can be controlled. The licensed version of the simVITRO software typically does not provide user access to controller configurations due to liability concerns.

4.5 Actuator Configuration File

Similar to the Sensor class, the Actuator class is part of the SSCAD toolkit and serves as a hardware abstraction layer to permit a wide range of actuators to be utilized with little or no modification to higher level system software. The Actuator class configures 1 to 6-DOF actuators and allows them to be moved to accomplish a certain task. The distinction to consider with the actuator class is that it uses actuators to accomplish something. A single rotary axis can be used to rotate an item. A 6-DOF robot can be used to fully locate an item in space. In theory, a series of rotary actuators can be stacked to create a 6-DOF robot, though the Actuator class is not designed to create an actuator group that “stacks” single axis elements to build the 6-DOF system. The Actuator class is designed to quickly and efficiently provide access to the different actuator systems so that they can, in a coordinated fashion, accomplish something. In addition, nothing greater than a 6-DOF system is required since it fully defines a rigid body in space. The number of joints may exceed 6 for these systems, but the degrees of freedom are still 6. For example, a mobile robot with a 6-DOF robotic arm attached to it can place the object at the end effector at any position and orientation in space. Range of motion constraints of the serial arm may require the coordination of motion between the mobile robot and the arm. The optimization algorithms to provide this coordination and divide up the work will take place in a child class of the 6-DOF actuator class. Actuators do not need to be limited to motion creation devices. They could be digital or analog outputs of a board. They could be relays and switches. They could be heaters and light sources. It should also be noted that due to the nature of the hardware abstraction layer, it permits virtual actuators to be created and utilized in simulation based scenarios. In this way, system software can be developed, and significant debugging completed, without having any of the safety risks or capital costs associated with moving hardware. The simVITRO software is designed to require at least one 6-DOF robot.

Additional actuators can be added (e.g. linear muscle force actuators), though implementation and use is licensed due to liability concerns.

4.6 Specimen Configuration Files

Similar to the project configuration files, the specimen configuration files are a way to store parameters for a given specimen.

Files that would typically remain constant for a given mounting of a specimen are:

- Experiment Setup
- Trajectory Set

The default versions of these files will reside at Path = *Drive\Test Cell\Configuration*. The default files will be an empty experiment setup parameter set and an empty trajectory set.

The specimen specific versions of these files will reside at Path = *Drive\Test Cell\Project Identifier\Specimen Number\Configuration*.

Loading:

The specimen specific versions of the file will load when the Specimen Number is entered. When the program is first started or if there is not a specimen specific one available, the default files will be loaded.

Note: Some contents of these files (i.e. those due to microscribe points) may become obsolete if the specimen is removed and remounted. Care should be taken when loading this data to prevent inaccurate usage of the obsolete data.

Saving:

By default, when the save functions are called the files will be overwritten in this configuration directory. If the Specimen Number field is blank, it will prompt the user to provide a specimen number.

4.7 Experiment Run Configuration Files

It is possible for the user to modify Project and Specimen files throughout the course of an experiment. If a single file continues to be overwritten, the change history will be lost. It is also not known exactly which configuration was used on what specimen or experiment run. For this reason, prior to each run a copy of these files will be made to the Path = *Drive\Test Cell\Project Identifier\Specimen Number\Experiment Run Number_ Experiment Run Name\Configuration* Directory. In this way, all the necessary configuration data is captured for reference purposes.

4.8 Trajectory Files

The trajectory files contain the trajectory information for a given device. For the different types and definitions of some of these, see sections 3.16 and 3.41 of the latest revision of the Foot Functional Specification document. In summary, the trajectories for a given experiment may include:

- Ground/Tibia Motion Trajectory
- Desired Ground Reaction Forces Trajectory
- Individual Tendon Actuator Force Trajectories
- Control Trajectories

For each trajectory listed, there are four types or ways to represent the trajectory:

- *ASCII Text Trajectory* – A trajectory where the data is in a tab delimited ASCII text format with n rows of header as defined by the comment character (#) and a fixed column format. This trajectory should be pre-processed to be normalized to the physiological parameters. In addition, the data will need to be pre-processed such that the sign and coordinate system corresponds to the appropriate coordinate system as shown in Appendix II of the latest revision of the Foot Functional Specification document.
- *Normalized Waveform Trajectory* - A trajectory where the time scale is from 0 to 1 in constant dt increments and the amplitude is a function of the physiological parameters. The trajectory is stored as a Labview waveform data type.
- *Engineering Units Waveform Trajectory* - A trajectory where the time scale is from 0 to the Move Duration for a given experiment in constant dt increments and the amplitude is from one end to the other end of the load or position range. The value and increments of dt is a function of the required dt for a given machine. For example, the dt for the Ground/Tibia Trajectory is a function of the robot motion program parameters and limits. The value of dt should be calculated to be the maximum value possible which still provides smooth performance of the machine. The trajectory is stored as a waveform data type.
- *Motion Program Trajectory* – A trajectory where the data is in an ASCII text format and formatted to be a motion program for the targeted actuator intended to follow the trajectory.
- *Control Trajectory* – A trajectory where all the data for the state of a 6 DOF simulation can be stored. The state includes kinetics, kinematics, gains, etc. for a series of setpoints in time. The setpoints do not need to be continuous in time and the data in between the setpoints is interpolated according to the interpolation enum. In addition, each control waveform has attributes that can be set to store channel names, units, etc.

Synchronizing multiple trajectories with ASCII text files:

In order to synchronize multiple trajectories and few simple assumptions need to be considered. First, the time stamp in the ASCII text file will not be used for any calculations other than to verify that the dt of the data points is constant. The duration of this file will be considered to be equal to 1 second. Secondly, the start and end of the file correspond to the start and end of the prescribed motion. Therefore, to synchronize two trajectories for different actuators the starting and ending data points in the respective files must have occurred at the same absolute time. It is possible and acceptable for the number of data points in the two files to be different. The program will stretch or compress the trajectories as needed with the assumption that the first data point of all the loaded trajectories correspond to the same absolute time, and that the last data point of all the loaded trajectories correspond to the same absolute time.

Conversion between trajectory types:

Due to specimen coordinate system or tendon actuator type changes, converting from motion programs to any other format is not advised. All trajectories with the exception of the motion programs will be stored in actuator and coordinate system neutral reference frames. For example, this means that the ground/tibia motion engineering unit trajectory will match the coordinate system of the normalized and ASCII text file which is from processed data collected in a gait lab. They are specimen and actuator independent.

There will be no default files for the trajectory storage. With the exception of the motion program, programmatically these will be individually stored in memory, and stored as a set in the Trajectory Set file. However, conventions should be followed for easy human recognition of these files. In addition, the program will be able to export individual trajectory files and it will follow these same conventions.

With the exception of the motion programs, these files will not be programmatically accessed so there is no requirement on the location of the storage of these files. However, it is advised that they are stored at Path = *Drive\Test Cell\Project Identifier\Data\Trajectories*. The motion programs will be a function of the parameters of a given experiment run and will be stored at Path = *Drive\Test Cell\Project Identifier\Specimen Number\Experiment Run Number_ Experiment Run Name\Data*.

There is currently untested software to convert between the Control trajectory and the other data types. Potentially trajectories can be converted from Control to the others, but the reverse is not possible due to the amount of information captured in the Control trajectory that is not stored in the other types.

Header Information:

ASCII Text Header – All rows in the header portion of the file must start with a “#” character. These rows will be ignored by the Musculoskeletal Simulator program but should provide important human readable information. The header rows should be used to reference as many of the following parameters as possible.

- Author
- Date created
- Original data set from which the data was created
 - Author
 - Date of collection
 - Publication
- Any post-processing software and revision
- Physiological parameters
 - Body weight
 - Foot Length
 - Foot Width
- Trajectory parameters or assumptions
 - Original move duration
 - Tendon Forces
- Notes
- Trajectory file revision history
- Data column definitions
 - Names
 - Descriptions
 - Units

Motion Program Header – All rows in the header portion of the file must start with the appropriate comment character. These rows will be ignored by the motion controller but should provide important human readable information. The header rows should be used to reference as many of the following parameters as possible.

- Musculoskeletal Simulator software name and revision
- Date created
- File path which contains
 - Project
 - Specimen
 - Experiment Run Number and Name
- Physiological parameters
 - Body weight
 - Foot Length
 - Foot Width
- Trajectory parameters

- Move duration
- Notes

4.9 Summary Files

A summary file editor window will appear in the End consumer action loop (this way it works for the regular test as well as the abort case). The header of the file will be indicated on the window and will contain all the test configurable parameters such as simulated body weight, cycle time, daq rate, recorded data types (strain, tendon, force plate) cycle number, Experiment Run number and name, project name, specimen name, list and paths of all the files generated during the run. It also provides a place to add any comments which will be appended to the file. The file should be named *Project Name_Specimen Number_Experiment Run Number_Experiment Run Name_Summary.txt* and stored in the root *Experiment Run Number* folder.

4.10 Log Files

Incomplete...

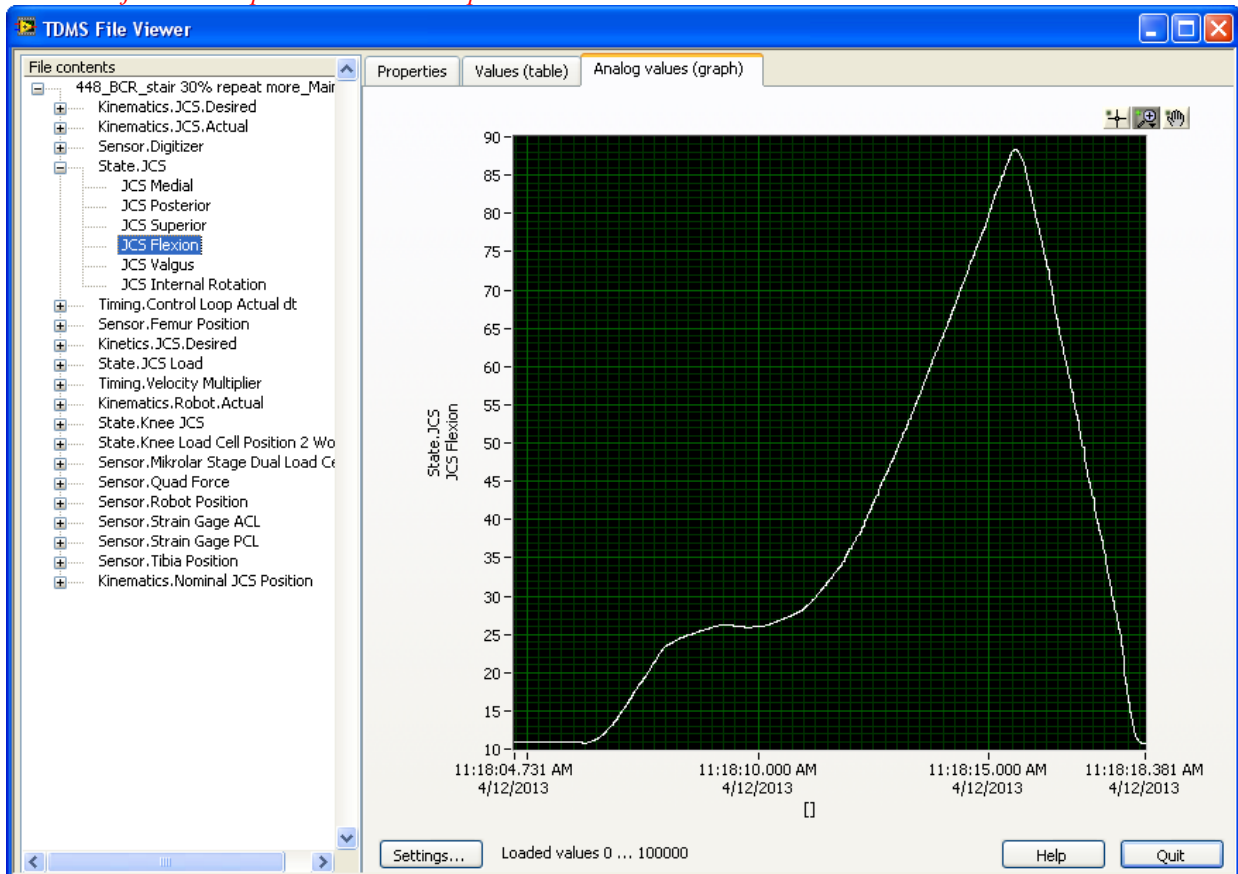
4.11 Experiment Run Files

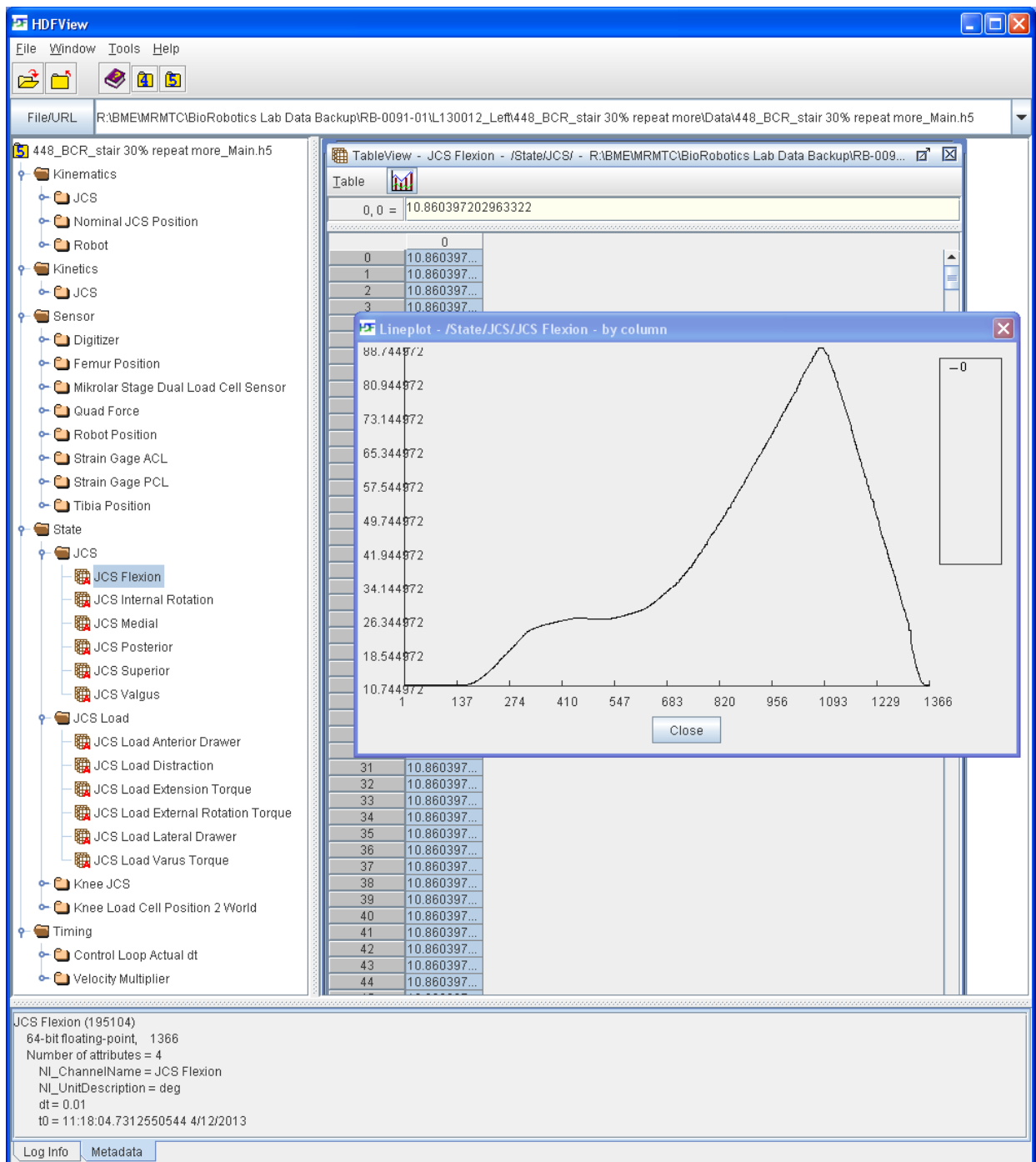
Files that are specific to an experiment run are:

- Experiment Run Data (Resulting Force Plate and Tendon Load Cell Data): *Experiment Run Number* + "_Main.tdm"
- Bone Strain Data: *Experiment Run Number* + "_Strain.tdm"
- Robot Exercise Trajectory: *Experiment Run Number* + ".robot"
- Tendon force profiles: *Experiment Run Number* + "." + *Tendon Name*

These files will reside in the folder:

C:\MS\Project Name\Specimen Number\Experiment Run Number





Optimized Files

Files created during an **Optimize** cycle will be:

- Robot Exercise Trajectory: “Optimized.robot”
- Tendon force profiles: “Optimized.” + *Tendon Name*
Example: Optimized.Peroneus_brevis

After the optimization cycle, these files will be placed in the folder generated for the prior experiment run:

C:\MS\Project Name\Specimen Number\Experiment Run Number

During the optimization cycle, the user will be prompted to enter the new Experiment Run Number and the system will automatically generate the Robot Exercise Trajectory and the Tendon Force Profiles by creating the folders and copying the optimized file contents into these files. Once this process is completed, the user would be able to execute a **Validate** followed by a **Run** to test the optimized force profiles and robot trajectories.

4.12 Zip Files

For a given Project, Specimen, or Experiment Run a WinZip file can be generated in the appropriate folder. This acts as a seal of the particular grouping of data so that programmatically will not be modified. It should contain all the data pertinent to the grouping.

During the normal course of an experiment run, a zip file will be automatically generated. If at any point prior to the completion of the run, the user invokes the **Abort** command, then the zip file will not be generated.

5 File Details

5.1 Tendon Actuator Configuration

Filename: *Project Identifier* + “_Tendon”

File extension: “.dat”

File Format: binary

File Type: Labview Datalog

Purpose:

This file contains all the configuration information for a group of tendon actuators that are intended to be used synchronously for a given experiment on the Musculoskeletal Simulator.

Contents:

See section 4.1 of the latest revision of the Foot Functional Specification document.

5.2 Force Plate Configuration

Filename: Project Identifier + “_ForcePlate”

File extension: “.dat”

File Format: binary

File Type: Labview Datalog

Purpose:

This file contains all the configuration information for the force plate used for a given experiment on the Musculoskeletal Simulator.

Contents:

See section 4.4 of the latest revision of the Foot Functional Specification document.

5.3 Bone Strain Configuration

Filename: Project Identifier + “_Strain”

File extension: “.dat”

File Format: binary

File Type: Labview Datalog

Purpose:

This file contains all the configuration information for a group of bone strain gage rosettes that are intended to be used synchronously for a given experiment on the Musculoskeletal Simulator.

Contents:

See section **X.X** of the latest revision of the Foot Functional Specification document.

5.4 Experiment Setup

Filename: Specimen Number + “_Setup”

File extension: “.dat”

File Format: binary

File Type: Labview Datalog

Purpose:

This file contains all the stored parameters for the experiment setup screens for a given specimen on the Musculoskeletal Simulator.

Contents:

See section 4.3 of the latest revision of the Foot Functional Specification document.

5.5 Trajectory Set

Filename: Specimen Number + “_Trajectory”

File extension: “.dat”

File Format: binary

File Type: Labview Datalog

Purpose:

This file contains all the trajectories for all trajectory format types (ASCII Text, Normalized waveform, Engineering units waveform, and Motion Program) for a given specimen and experiment on the Musculoskeletal Simulator.

The trajectory sets will be a function of the parameters of a given experiment run and will be stored at Path = *Drive\Test Cell\Project Identifier\Specimen Number\Experiment Run Number_ Experiment Run Name\Data*.

Contents:

See section 4.5 of the latest revision of the Foot Functional Specification document.

5.6 Ground/Tibia Motion Trajectory

Filename: Project Identifier + “_” + Experiment Run Name + “_” + Trajectory Type + “_Trajectory”

File extension: “. ” + Trajectory Name

ASCII Text

File Format: ASCII Text

File Type: Text

Delimiter: Tab

Header Rows: As needed and is defined by “#” character.

Normalized and Engineering Units Waveform

File Format: binary

File Type: Labview Datalog

Motion Program

File Format: ASCII Text

File Type: Text

Delimiter: None

Header Rows: As needed and should be commented out using appropriate comment characters.

Purpose:

This file contains the trajectory for the relative motion between the tibia and the ground during a specified motion. With the exception of the motion program, this data should be processed into the GND coordinate system using GTP formatted data. Refer to section 5.10 and Appendix II of the latest revision of the Foot Functional Specification document for the definition of the coordinate system.

Normalization:

The trajectory terms need to be normalized to physiological parameters of Foot Length (FL) and Foot Width (FW). Before any normalization calculations should occur, the raw data will need to be converted to the proper reference frame. Since the angles do not need scaled, only the a, m, and s values need to be normalized. Since the data is stored in the anatomical reference frame it does not matter if it is from a right or left specimen. *Note: this of course produces a left handed coordinate system for a left specimen, but the system software will resolve this.*

Trajectory Variables

Ground/Tibia Position (a, m, s, r, t, o)

Variable Abbreviations

E.x = x variable in Engineering Units*

N.x = x variable in Normalized Units

* Note that these must be the same units that the physiological parameters use.

Equations

$$N.a = \frac{E.a}{FL} \cdot 100\%$$

$$N.m = \frac{E.m}{FW} \cdot 100\%$$

$$N.s = \frac{E.s}{\frac{1}{2}(FL + FW)} \cdot 100\%$$

$$N.r = E.r$$

$$N.t = E.t$$

$$N.o = E.o$$

ASCII Contents:

Header – Standard. See section 4.4.

Data –

Column #	Name	Description	Units
1	Time	Relative Time	Seconds
2	a	anterior translation of ankle center	%FL
3	m	medial translation of ankle center	%FW
4	s	superior translation of ankle center	%m/m
5	r	internal rotation angle (positive for internal rotation of the tibia)	rad
6	t	lateral tilt angle (positive for bowleggedness)	rad
7	o	somersault angle (positive for forward rotation)	rad

Waveform and Motion Program Contents:

Refer to section 4.5 and 5.64 of the latest revision of the Foot Functional Specification document.

5.7 Ground Reaction Forces Trajectory

Filename: Project Identifier + “_” + Experiment Run Name + “_” + Trajectory Type + “_Trajectory”

File extension: “.” + Trajectory Name

ASCII Text

File Format: ASCII Text

File Type: Text

Delimiter: Tab

Header Rows: As needed and is defined by “#” character.

Normalized and Engineering Units Waveform

File Format: binary

File Type: Labview Datalog

Motion Program

N/A

Purpose:

This file contains the ground reaction forces between the foot and the ground during a specified motion. This data should be processed into the GND coordinate system. Refer to sections 5.7, 5.10, and Appendix II of the latest revision of the Foot Functional Specification document.

Normalization:

The trajectory terms need to be normalized to physiological parameters of Body Weight (BW), Foot Length (FL) and Foot Width (FW). Before any normalization calculations should occur, the raw data will need to be converted to the GND coordinate system.

For the center of pressure (COP) parameters (CP_A, CP_M) the method used for the averaging and normalization should be similar to the method developed by Nigg et al (1990). This will take into account how to handle line of progression and orientation of anatomical structures on the foot relative to the COP values. In addition to recasting the data in the anatomical reference frame described in this document the only exception that needs to be made with Nigg's method is the location of the origin. These changes are needed to orient the data in the GND coordinate system. Nigg employs an origin that is located by the intersection of the midlines of the foot length and width when the foot print left on the force plate as well as the subject's foot are examined. When looking at center of pressure data only, this origin method is certainly adequate. However, if combining with motion capture data, a ground coordinate system that can be found in both data sets may provide more universal results. For this reason, the definition of O_{GND} is found in section 5.10 of the latest revision of the Foot Functional Specification document.

For the remaining parameters, they too will need to be converted into the GND coordinate system. The normalization of the forces (Fa, Fm, Fs) will use the commonly accepted practice of percent body weight (%BW). The last parameter is the internal rotation couple moment at the center of pressure, Tr. This parameter will be scaled by body weight and the average of the foot length and width. Since the data is stored in the anatomical reference frame it does not matter if it is from a right or left specimen. *Note: this of course produces a left handed coordinate system for a left specimen, but the system software will resolve this.*

Trajectory Variables

Ground Reaction Forces (Fa, Fm, Fs, CPa, CPm, Tr)

Variable Abbreviations

E.x = x variable in Engineering Units*

N.x = x variable in Normalized Units

* Note that these must be the same units that the physiological parameters use.

Equations

$$N.F_A = \frac{E.F_A}{BW} \cdot 100\%$$

$$N.F_M = \frac{E.F_M}{BW} \cdot 100\%$$

$$N.F_S = \frac{E.F_S}{BW} \cdot 100\%$$

$$N.CP_A = \frac{E.CP_A}{FL} \cdot 100\%$$

$$N.CP_M = \frac{E.CP_M}{FW} \cdot 100\%$$

$$N.T_R = \frac{E.T_R}{BW \cdot \frac{1}{2}(FL + FW)} \cdot 100\%$$

ASCII Contents:

Header – Standard. See section 4.4.

Data –

Column #	Name	Description	Units
1	Time	Relative Time	Seconds
2	Fa	Force in the posteroanterior axis (N) – positive in anterior direction	%BW
3	Fm	Force in mediolateral axis (N) – positive in medial direction	%BW
4	Fs	Force in inferosuperior axis (N) – positive in superior direction	%BW
5	CPa	Center of pressure in the posteroanterior axis (m) – positive in anterior direction	%FL
6	CPm	Center of pressure in the mediolateral axis (m) – positive in medial direction	%FW
7	Tr	Resulting internal rotation couple moment (Nm)	%Nm/(N-m)

Waveform Contents:

Refer to section 4.5 of the latest revision of the Foot Functional Specification document.

5.8 Tendon Actuator Trajectory

Filename: Project Identifier + “_” + Experiment Run Name + “_” + Trajectory Type + “_Trajectory”

File extension: “.” + Trajectory Name

ASCII Text

File Format: ASCII Text

File Type: Text

Delimiter: Tab

Header Rows: As needed and is defined by “#” character.

Normalized and Engineering Units Waveform

File Format: binary

File Type: Labview Datalog

Motion Program

File Format: ASCII Text

File Type: Text

Delimiter: None

Header Rows: As needed and should be commented out using appropriate comment characters.

Purpose:

This file contains the tendon force trajectories for a given muscle during a specified motion. This data is a scalar and as such is coordinate system independent.

Normalization:

The trajectories need to be normalized to physiological parameter of Body Weight (BW). The normalization of the forces will use the commonly accepted practice of percent body weight (%BW)

Trajectory Variables

Tendon Actuator Force – (F)

Variable Abbreviations

E.x = x variable in Engineering Units*

N.x = x variable in Normalized Units

* Note that these must be the same units that the physiological parameters use.

Equation

$$N.F = \frac{E.F}{BW} \cdot 100\%$$

ASCII Contents:

Header – Standard. See section 4.4.

Data –

Column #	Name	Description	Units
1	Time	Relative Time	Seconds
2	F	Muscle Force	%BW

Waveform and Motion Program Contents:

Refer to section 4.5 and 5.64 of the latest revision of the Foot Functional Specification document.

5.9 Summary File

Incomplete...

5.10 Main Data File

Filename: Experiment Run Number_Experiment Run Name + “_Main”

File extension: “.tdms”

File Format: binary

File Type: Labview TDMS and eventually HDF5 format (.h5)

Purpose:

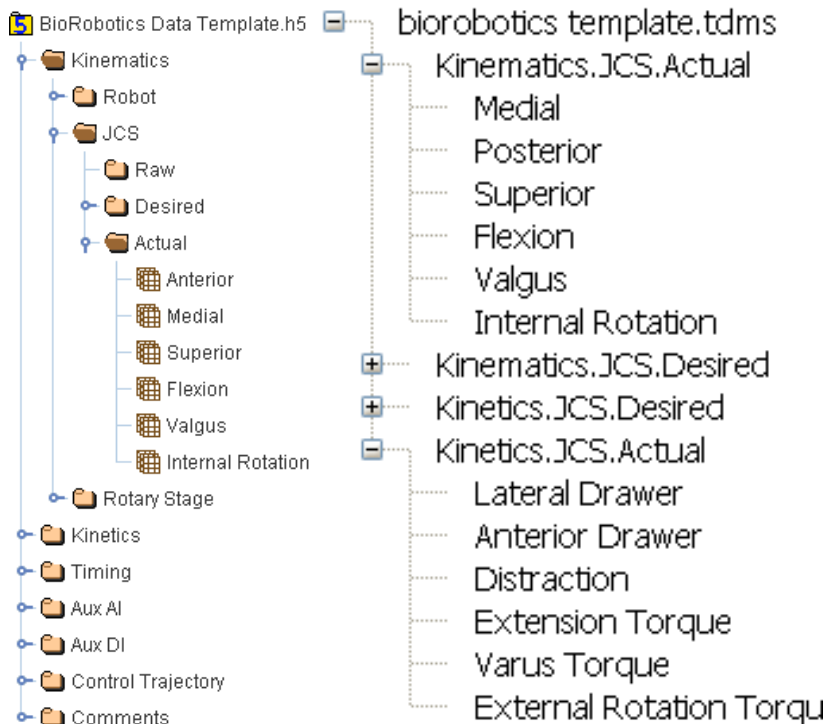
This file stores all the main data for a given experiment run. Data is organized into groups depending on the transducer, type, state, or source of the data. It contains all the data acquired by the DAQ boards, all the configuration information, as well as any data displayed to the user during the execution of the motion. This same file includes the raw as well as the processed data.

Contents:

The file is organized by groups and channels. Not all files will have contents for all groups, and not all group names will have the same channel names underneath. However, the following will outline for each group the following items:

1. Data type expected
2. Number of channels expected
3. Order of channels expected
4. Units expected

These group names will be universal across platforms and each UMS joint test system will define more specifically things like which groups are used, the corresponding channel names and units, etc. For the amount and type of data being created the TDMS format is limiting in its ability to create and organize data in a hierarchical format. The HDF5 format provides more flexibility toward this end, but current HDF5 LabVIEW drivers have yet to be proven to be a reliable format for quickly streaming large amounts of data to disk. This section will be organized such that data can be stored in both file types and still be interpreted in the same way. It will be organized via the following data structure. **Category.DataSource.DataState.Channels**. In HDF5 format this will have with it a corresponding tree structure. In .tdms format we will just have a large group name and use “.” in between all the various subgroups. The figures below show the comparison of the two techniques.



Commonly Used Categories:

Kinematics
Kinetics
Timing
Aux AI Channels
Aux DI Channels
Configuration
Comments

Commonly Used DataSources:

Robot
Rotary Stage
Indexing Ring
Load Cell
Force Plate
Tendon Actuator
Optotrak
Microscribe
JCS
Nominal JCS Position
Aux
Sync Trigger
Control Loop Actual dt
Velocity Multiplier
Control Trajectory
Setup
Transformation

Specimen details (donor info, specimen details, healthy/diseased, etc.)

- Experimenter details (who collected)
- Setup details (robot model, etc., when collected)
- Other data (muscle representations, pressures, etc.)
- Licensing (who owns the data and liabilities, etc.)

Commonly Used DataStates:

Raw
Desired
Actual

Not all combinations of Category, DataSource, and DataState are meaningful parameters. For example, *Kinematics.Load Cell* has no practical meaning and this group should not be created. In addition, each file need not contain all the meaningful parameters if they are not meaningful to that particular experiment. For example, anything having to do with the rotary stage has no meaning when collecting data from a foot/ankle experiment since that hardware is not utilized. Additional parameters can be added to these lists as it makes sense, however, this should only occur after it is determined that none of the current parameters can be built into a meaningful combination. The following is a non-exclusive list of commonly used combinations.

Commonly Used Combinations:

Kinematics.Robot.Raw
Kinematics.Rotary Stage.Raw
Kinematics.Indexing Ring
Kinematics.JCS.Raw
Kinematics.JCS.Desired

Kinematics.JCS.Actual
 Kinematics.Nominal JCS Position
 Kinematics.Aux.Raw
 Kinematics.Aux.Desired
 Kinematics.Aux.Actual
 Kinetics.Load Cell.Raw
 Kinetics.JCS.Raw
 Kinetics.JCS.Desired
 Kinetics.JCS.Actual
 Kinetics.Tendon Actuator.Raw
 Kinetics.Tendon Actuator.Desired
 Kinetics.Tendon Actuator.Actual
 Kinetics.Aux.Raw
 Kinetics.Aux.Desired
 Kinetics.Aux.Actual
 Timing.Sync Trigger
 Timing.Control Loop Actual dt
 Timing.Velocity Multiplier
 Aux AI Channels
 Aux DI Channels
 Configuration.Control Trajectory
 Configuration.Setup
 Configuration.Load Cell
 Configuration.Transformation
 Comments

For proper interpretation of the each group, subgroups, and channel, sometimes additional information is required. This type of information are considered to be properties or attributes of the data. For example, the channel data for anterior translation could a long list of amplitude values, but it is unknown where each of these amplitude values line up in time. Two scalar value attributes such as initial time and dt between each sample are all the additional information needed to build a waveform. In addition, channel names and units can also be stored as attributes. In addition to using attributes at the channel level, these can also be utilized at the group level. For example, knowing that all information in the *Kinematics.JCS.Actual* group is in the C2-C3 reference frame carries a different meaning than if it was in the C4-C5 reference frame. It may also be important to know the source of the data. For example, if the source for the *Kinematics.JCS.Actual* data is calculated based on the robot position rather than derived from optical motion tracking variation in the results can be more correctly interpreted.

Common group/channel attribute names:

NI_ChannelLength
 NI_ChannelName
 NI_DataType
 NI_OriginalChannelName
 NI_UnitDescription
 wf_increment
 wf_samples
 wf_start_offset
 wf_start_time
 TargetDevice
 Reference Frame (TIB_FEM, TIB_WORLD, C2_C3, etc..)
 Source (robot, optotrak, rotary stage, tendon actuator, microscribe, load cell, tekscan sensor etc...)
 Run Name (could be attribute for entire file)
 Run Number
 Cycle Number

***** Below is TBD *****

Group: Kinematics

Description: These are the kinematics of the specimen in some anatomical reference frame. This group will contain both desired and actual kinematics and be organized into banks of channels. Each bank will be Channels:

Group – GRF_Raw

Channel	Description	Units	Transducer - Notes
Dev1/ai0	Fx	m	Force Plate - Direct from DAQ board - Unfiltered - Calibration matrix not applied
Dev1/ai1	Fy	m	
Dev1/ai2	Fz	m	
Dev1/ai3	Mx	rad	
Dev1/ai4	My	rad	
Dev1/ai5	Mz	rad	

Group – Force Plate

Channel	Description	Units	Transducer - Notes
Fa	anterior shear force	N	Force Plate - Unfiltered - In GND coordinate system
Fm	medial shear force	N	
Fs	superior force	N	
CPa	anterior COP	m	
CPm	medial COP	m	
Tr	internal rotation moment	Nm	

Group – GRF_Raw

Channel	Description	Units	Transducer - Notes
Dev1/ai0	Fx	Volts	Force Plate - Direct from DAQ board - Unfiltered - Calibration matrix not applied
Dev1/ai1	Fy	Volts	
Dev1/ai2	Fz	Volts	
Dev1/ai3	Mx	Volts	
Dev1/ai4	My	Volts	
Dev1/ai5	Mz	Volts	

Group – Sync Pulse

Channel	Description	Units	Transducer - Notes
Dev1/ai6	Sync Pulse	Volts	Robot Controller - 5 to 0V falling edge means trigger start

Group – Tendon Actuator_Raw

Channel	Description	Units	Transducer - Notes
Dev1/ai7	F1	Volts	Tendon Actuator - Unfiltered - Direct from DAQ board - Includes data for all channels (enabled and disabled)
Dev1/ai8	F2	Volts	
Dev1/ai9	F3	Volts	
Dev1/ai10	F4	Volts	

Dev1/ain	Fn	Volts	
----------	----	-------	--

Group – Force Plate

Channel	Description	Units	Transducer - Notes
Fa	anterior shear force	N	Force Plate - Unfiltered - In GND coordinate system
Fm	medial shear force	N	
Fs	superior force	N	
CPa	anterior COP	m	
CPm	medial COP	m	
Tr	internal rotation moment	Nm	

Group – Tendon Actuator

Channel	Description	Units	Transducer - Notes
F1	Tendon 1 force	N	Tendon Actuator - Unfiltered - Only includes data for the enabled channels
F2	Tendon 2 force	N	
F3	Tendon 3 force	N	
F4	Tendon 4 force	N	
Fn	Tendon n force	N	

5.11 Strain Data File

Filename: Experiment Run Number_Experiment Run Name + “_Strain”

File extension: “.tdms”

File Format: binary

File Type: Labview TDMS

Purpose:

This file stores all the strain data for a given experiment run. Data is organized into groups depending on the rosette or source of the data. This file is considered the raw data file. It contains all the data acquired by the DAQ board as well as any data displayed to the user during the execution of the motion.

Contents:

Group – Force Plate_Raw

Channel	Description	Units	Transducer - Notes
Dev1/ai0	Fx	Volts	Force Plate - Direct from DAQ board - Unfiltered - Calibration matrix not applied
Dev1/ai1	Fy	Volts	
Dev1/ai2	Fz	Volts	
Dev1/ai3	Mx	Volts	
Dev1/ai4	My	Volts	
Dev1/ai5	Mz	Volts	

Group – Sync Pulse

Channel	Description	Units	Transducer - Notes
Dev1/ai6	Sync Pulse	Volts	Robot Controller - 5 to 0V falling edge means trigger start

Group – Tendon Actuator_Raw

Channel	Description	Units	Transducer - Notes
Dev1/ai7	F1	Volts	Tendon Actuator - Unfiltered - Direct from DAQ board - Includes data for all channels (enabled and disabled)
Dev1/ai8	F2	Volts	
Dev1/ai9	F3	Volts	
Dev1/ai10	F4	Volts	
Dev1/ain	Fn	Volts	

Group – Robot Position

Channel	Description	Units	Transducer - Notes
x	X – position	mm	Robot Position - In ROB coordinate system - Data may not be continuous since MEI controller can't produce the positions fast enough if the DAQ rate is too high.
y	Y – position	mm	
z	Z – position	mm	
Roll	Roll angle	deg	
Pitch	Pitch angle	deg	
Yaw	Yaw angle	deg	

Group – Force Plate

Channel	Description	Units	Transducer - Notes
Fx	X shear force	N	Force Plate - Unfiltered - In GND coordinate system
Fy	Y shear force	N	
Fz	Normal force	N	
Xp	X COP	m	
Yp	Y COP	m	
Tz	COP moment	Nm	

Group – Tendon Actuator

Channel	Description	Units	Transducer - Notes
F1	Tendon 1 force	N	Tendon Actuator - Unfiltered - Only includes data for the enabled channels
F2	Tendon 2 force	N	
F3	Tendon 3 force	N	
F4	Tendon 4 force	N	
Fn	Tendon n force	N	

5.12 Processed Main Data File

Filename: Experiment Run Number_Experiment Run Name + “_Main” + “_Processed”

File extension: “.tdms”

File Format: binary

File Type: Labview TDMS

Purpose:

This file stores all the main data for a given experiment run. Data is organized into groups depending on the transducer or source of the data. This file is considered the processed data file. It contains all the filtered and converted versions of the pertinent data from the run. At a minimum this file needs to contain all the data necessary to execute the optimization process. In the time domain, this file starts from where the sync pulse was sent and is the length of the move duration

Contents:

Group – GTP

Channel	Description	Units	Transducer - Notes
a	anterior translation of ankle center	m	Robot Position - In GTP coordinate system - This is the desired trajectory
m	medial translation of ankle center	m	
s	superior translation of ankle center	m	
r	internal rotation angle (positive for internal rotation of the tibia)	rad	
t	lateral tilt angle (positive for bowleggedness)	rad	
o	somersault angle (positive for forward rotation)	rad	

Group – GRF

Channel	Description	Units	Transducer - Notes
Fa	anterior shear force	N	Force Plate - Filtered then recalculated - In GND coordinate system
Fm	medial shear force	N	
Fs	superior force	N	
CPa	anterior COP	m	
CPm	medial COP	m	
Tr	internal rotation moment	Nm	

Group – Tendon Actuator

Channel	Description	Units	Transducer - Notes
F1	Tendon 1 force	N	Tendon Actuator - Filtered - Only includes data for the enabled channels
F2	Tendon 2 force	N	
F3	Tendon 3 force	N	
F4	Tendon 4 force	N	
Fn	Tendon n force	N	

5.13 Optimized Ground/Tibia Motion Trajectory File

Incomplete...

5.14 Optimized Ground Reaction Forces Trajectory File

Incomplete...

5.15 Optimized Tendon Actuator Trajectory File

Incomplete...

5.16 Experiment Zip File

Filename: Project Identifier_Specimen Number_Experiment Run Number_Experiment Run Name

File extension: “.zip”

File Format: WinZip

File Type: binary

Delimiter: N/A

Header Rows: N/A

Incomplete...

This file will contain the following files:

- *Project Name + “_Tendon.dat”*
- *Project Name + “_Strain.dat*
- *Project Name + “.robot”*
- *Project Name + “.target”*
- *Project Name + “. ” + ”Tendon 1 Name”*
- *Project Name + “. ” + ”Tendon 2 Name”*
- *Project Name + “. ” + ”Tendon 3 Name”*
- *Project Name + “. ” + ”Tendon 4 Name”*
- *Project Name + “. ” + ”Tendon 5 Name”*
- *Specimen Number” + “.setup”*
- *Experiment Run Number + “. ”_Main.TDM”*
- *Experiment Run Number + “. ”_Strain.TDM”*
- *Experiment Run Number + “.robot”*
- *Experiment Run Number + “. ” + Tendon Name 1*
- *Experiment Run Number + “. ” + Tendon Name 2*
- *Experiment Run Number + “. ” + Tendon Name 3*
- *Experiment Run Number + “. ” + Tendon Name 4*
- *Experiment Run Number + “. ” + Tendon Name 5*

5.17 Message Log File

Incomplete...

5.18 Processed Strain Data File

Filename: Experiment Run Number_Experiment Run Name + “_Strain” + “_Processed”

File extension: “.tdms”

File Format: binary

File Type: Labview TDMS

Purpose:

This file stores all the strain data for a given experiment run. Data is organized into groups depending on the rosette or source of the data. This file is considered the processed data file. It contains all the filtered and converted versions of the pertinent data from the run. At a minimum this file needs to contain all the data necessary to execute the optimization process. In the time domain, this file starts from where the sync pulse was sent and is the length of the move duration

Contents:

Group – PJCS Position

Channel	Description	Units	Transducer - Notes
<i>a</i>	Medial translation	m	Robot Position - In PJCS coordinate system - Data must be continuous (interpolate as needed)
<i>b</i>	Posterior translation	m	
<i>c</i>	Superior translation	m	
α	Dorsi/Plantarflexion	rad	
β	In/Eversion	rad	
γ	Internal rotation	rad	

Group – Force Plate

Channel	Description	Units	Transducer - Notes
Fx	X shear force	N	Force Plate - Filtered then recalculated - In GND coordinate system
Fy	Y shear force	N	
Fz	Normal force	N	
Xp	X COP	m	
Yp	Y COP	m	
Tz	COP moment	Nm	

Group – Tendon Actuator

Channel	Description	Units	Transducer - Notes
F1	Tendon 1 force	N	Tendon Actuator - Filtered - Only includes data for the enabled channels
F2	Tendon 2 force	N	
F3	Tendon 3 force	N	
F4	Tendon 4 force	N	
Fn	Tendon n force	N	

5.19 Control Trajectory File

Filename: Project Identifier + “_” + Experiment Run Name + “_” + Trajectory Type + “_Trajectory”

File extension: “.” + Trajectory Name

ASCII Text

File Format: ASCII Text

File Type: Text

Delimiter: Tab

Header Rows: As needed and is defined by “#” character.

Normalized and Engineering Units Waveform

File Format: binary

File Type: Labview Datalog

Motion Program

N/A

Purpose:

This file contains a trajectory where all the data for the state of a 6 DOF simulation can be stored. The state includes kinetics, kinematics, gains, etc. for a series of setpoints in time. The setpoints do not need to be continuous in time and the data in between the setpoints is interpolated according to the interpolation enum. In addition, each control waveform has attributes that can be set to store channel names, units, etc.

ASCII Contents:

Header – Standard. See section 4.4.

Data –

Column #	Name	Description	Units
1	Time	Relative Time	Seconds
2	Fa	Force in the posteroanterior axis (N) – positive in anterior direction	%BW
3	Fm	Force in mediolateral axis (N) – positive in medial direction	%BW
4	Fs	Force in inferosuperior axis (N) – positive in superior direction	%BW
5	CPa	Center of pressure in the posteroanterior axis (m) – positive in anterior direction	%FL
6	CPm	Center of pressure in the mediolateral axis (m) – positive in medial direction	%FW
7	Tr	Resulting internal rotation couple moment (Nm)	%Nm/(N-m)

Waveform Contents:

Refer to section 4.5 of the latest revision of the Foot Functional Specification document.

Appendix I. REFERENCES

- Wu, G., Siegler, S., Allard, P., Kirtley, C., Leardini, A., Rosenbaum, D., Whittle, M., D'Lima, D., Cristofolini, L., Witte, H., Schmidt, O., Stokes, I., 2002. ISB recommendations on definitions of joint coordinate system of various joints for the reporting of human movement – Part I: ankle, hip, spine. *Journal of Biomechanics* 35, 543-548.
- Motriuk, H.U., Nigg, B.M., 1990. A Technique for Normalizing Centre of Pressure Paths. *Journal of Biomechanics* 23, 927-932.
- Yeadon, MR, 1990. The simulation of aerial movement--I. The determination of orientation angles from film data. *J Biomech* 23:59-66.