

Modelling Instrumented Treadmills

Important Note: Before going through this tutorial or collecting instrumented treadmill data, you should review the wiki page dedicated to [Treadmills](#). This page includes most of the important information you need to know to collect treadmill data that can be modelled and analysed in Visual3D.

With that noted, this tutorial covers the basic processing steps that apply when analysing data from instrumented treadmills. These steps are intended to be general and should be applicable for instrumented treadmills from different vendors including:

1. [AMTI](#)
2. [Bertec](#)
3. [Treadmetrix](#)

In any case, please consult any vendor instructions or guides for your instrumented treadmills to ensure that you understand the sensors you are using. Users may want to be familiar with [Force Platforms](#) in general and have completed the [tutorial](#) on the subject.

Data and Files

This tutorial uses a subset of files from our Visual3D Workshop dataset. These files include 1 [static trial](#), 4 [dynamic trials](#) using an instrumented treadmill, and a [model template file](#).

Load Files and Apply Model

1. Model > Hybrid Model from C3D File. Select Static01.c3d. 2. In the Models tab, Load Model Template file. Select ModellingInstrumentedTreadmillsTutorial.mdh. 3. File > Open/Add... Select the 4 dynamic trials. 4. Model > Assign Model to Motion Files. Assign Static01.c3d to all 4 dynamic trials. 5. Use the Signal and Events tab's Animation feature to verify that the model has been applied to all four dynamic motion trials and that the treadmill platforms appear in the animation.

Model the Instrumented Treadmill

6. Settings > Use Processed Analogs for Ground Reaction Force Calculations. 7. Settings > Use Processed Targets for Model/Segment/LinkModelBased items. 8. Don't check Use Processed Forces. 9. Ensure [ALL_FILES are active](#), then Force > Modify Minimum Force Platform Value.... Set the New Minimum for all files to 50 N. 10. Ensure ALL_FILES are active, then open the Pipeline Workshop. Run the [FP_Auto_Baseline](#) with the default parameters. This command automatically determines the appropriate baseline values to remove bias from each analog signal and accomplishes three things:

1. Zeros the force signal without having to specify a [frame range](#) (AUTO_ZERO)
2. Defines the [minimum force platform value](#) based on the noise in the signal (AUTO_FP_MINIMUM)
3. Looks at the before/after the signal loaded - and finds when the signal crosses the reduced FP minimum to ensure as much of the signal is used as possible (USE_REDUCED_FP_MINIMUM_AROUND_FORCE)

The [minimum force platform value](#) is determined by identifying the values of the plate during all the periods where the plate is unloaded. Each force component is used to determine the set of frames containing median values for each component, and a single representative minimum force frame is determined from these values.

Process ANALOG Signals

11. Filter the [ANALOG](#) signals in the dynamic motion trials using a Lowpass filter. 12. Verify that the ANALOG signals were filtered in the Signals and Events tab by: a. making one file active and graphing the Force.Fx1 signal from both the [ORIGINAL](#) and [PROCESSED](#) folders. b. Opening up the Data View for ANALOG::PROCESSED::Force.Fx1 and viewing the [Signal Processing History](#). 13. Hit RECALC in order to run the [RECALC pipeline](#) and cause these changes to take effect.

NOTE: These steps can also be done using the pipeline using the [Lowpass_Filter](#) and [Recalc](#) commands.

14. Validate the Force Assignments.

Compute Kinematic Variables

As an example of kinematic variables where the treadmill must be taken into account, we will compute [temporal distance measures](#).

15. Ensure ALL_FILES are active, then open the Pipeline Workshop. Run the [Automatic_Gait_Events](#) command with the default parameter values. 16. Run the [Metric_Compute_Temporal_Gait](#) command accounting for the speed and direction of the treadmill. 17. Verify the temporal distance measures for each file.

Compute Kinetic Variables

As an example of calculating kinetic variables, we will calculate [JOINT_POWER model-based items](#) for each lower extremity [joint](#).

18. Create the first JOINT_POWER item. 19. Create the remaining JOINT_POWER items. 20. Verify the values.

Conclusion

This tutorial guided you through the process of using Visual3D to model data collected with an instrumented treadmill. It covered the following topics:

1. Modelling the treadmill
2. Processing the ANALOG signals
3. Computing Kinematic variables
4. Computing Kinetic variables

From:
<https://has-motion.com/wiki/> - Wiki Documentation Site

Permanent link:
https://has-motion.com/wiki/doku.php?id=visual3d:tutorials:kinematics_and_kinetics:modelling_instrumented_treadmills&rev=1732907774

Last update: **2024/11/29 19:16**

